



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**WEBOVÝ SIMULÁTOR FOTBALOVÝCH
LIG A TURNAJŮ**

WEB SIMULATOR OF FOOTBALL LEAGUES AND CHAMPIONSHIPS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN URBANCZYK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ HYNEK

BRNO 2019

Zadání diplomové práce



21701

Student: **Urbanczyk Martin, Bc.**
Program: Informační technologie Obor: Informační systémy
Název: **Webový simulátor fotbalových lig a turnajů**
Web Simulator of Football Leagues and Championships
Kategorie: Uživatelská rozhraní

Zadání:

1. Seznamte se s principy strojového učení založených na technice učení s učitelem.
2. Proveďte průzkum existujících fotbalových lig a turnajů. Prostudujte jejich pravidla a analyzujte atributy, které mají vliv na výsledky daných turnajů.
3. Navrhněte model reprezentující vybrané atributy fotbalových lig a turnajů. Získejte reálná historická data.
4. Navrhněte webový systém pro simulaci fotbalových lig a turnajů využívající predikci výsledků na základě historických dat.
5. Navržený webový systém implementujte.
6. Otestujte funkcionality a použitelnost výsledné systému.

Literatura:

- Machine Learning. *Coursera: Online Courses From Top Universities* [online]. 2018 [cit. 2018-10-01]. Dostupné z: <https://www.coursera.org/learn/machine-learning>
- SciKit-Learn Developers. *SciKit-Learn User Guide* [online]. 2018 [cit. 2018-10-01]. Dostupné z: http://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf
- Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Burlington: Morgan Kaufmann Publishers/Elsevier, 2010, ISBN 978-0-12-375030-3.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Hynek Jiří, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 23. října 2018

Abstrakt

Tato diplomová práce pojednává o tvorbě webového simulátoru fotbalových soutěží. V rámci této práce byla nastudována problematika systémů fotbalových principů a soutěží. Dále pak problematika strojového učení a jejich využití v této práci. Dále proběhla analýza existujících a podobných řešení a části těchto řešení také posloužily jako inspirace. Poté byl vytvořen návrh systému a všech jeho klíčových podčástí. Následně byl tento simulátor implementován a jednotlivé jeho klíčové podčásti otestovány. Aplikace umožňuje simulovat pět nejlepších ligových soutěží dle žebříčku koeficientů organizace UEFA.

Abstract

This thesis is about the creation of a simulator of football leagues and championships. I studied the problematics of football competitions and their systems and also about the base of machine learning. There was also an analysis of similar and existing solutions and I took inspiration for my proposal from them. After that, I made the design of the whole simulator structure and of all of its key parts. Then the simulator was implemented and tested. The application allows simulating top five competitions in UEFA club coefficients rating.

Klíčová slova

fotbal, fotbalové soutěže, simulátor, simulace, strojové učení, regresivní modely, predikce, webová aplikace, Python, SciKit-learn, REST architektura, Node.js

Keywords

football, football competitions, simulator, simulation, machine learning, regression models, prediction, web application, Python, SciKit-learn, REST architecture, Node.js

Citace

URBANCZYK, Martin. *Webový simulátor fotbalových lig a turnajů*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek

Webový simulátor fotbalových lig a turnajů

Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Martin Urbanczyk
21. května 2019

Poděkování

Děkuji panu Ing. Jiřímu Hynkovi za pomoc a vstřícnost při tvoření této diplomové práce, za jeho energii a množství nápadů, které mi předal. Dále pak děkuji Martinu Pavelkovi za pomoc při nastavování serveru, na němž aplikace běží a Sylvii Enenkelové za tvorbu loga a pomoc při designu aplikace. Dále pak děkuji Maroši Vasilišínovi a Miroslavu Pavelkovi za pomoc při testování aplikace. V neposlední řadě pak mé rodině a přítelkyni za podporu po celou dobu studia.

Obsah

1	Úvod	4
2	Úvodní analýza	5
2.1	Motivace pro tvorbu simulátoru	5
2.2	Analýza tématu	5
2.3	Existující řešení využívající simulaci výsledků	6
2.3.1	Football Manager	6
2.3.2	Czech Soccer Manager	7
2.3.3	FIFA	8
2.3.4	Existující diplomové práce	9
2.4	Přínosy a cíle řešení	9
3	Fotbal, fotbalové ligy a turnaje	10
3.1	Základní pravidla fotbalu	10
3.2	Světové organizace	11
3.2.1	FIFA	11
3.2.2	CONCACAF	11
3.2.3	CONMEBOL	11
3.2.4	CAF	11
3.2.5	AFC	12
3.2.6	OFC	12
3.2.7	UEFA	12
3.3	Typy soutěží	13
3.4	Národní soutěže	13
3.4.1	Anglické soutěže	14
3.4.2	Španělské soutěže	14
3.4.3	Italské soutěže	15
3.4.4	Německé soutěže	15
3.4.5	Francouzské soutěže	15
3.5	Klubové mezinárodní soutěže	16
3.5.1	Koeficienty	16
3.5.2	Liga mistrů UEFA	16
3.5.3	Evropská liga	16
3.6	Reprezentační soutěže	18
3.6.1	Mistrovství světa	18
3.6.2	Kontinentální mistrovství	18
3.7	Složení fotbalového týmu	19
3.7.1	Sestavy a pozice hráčů	19

3.7.2	Taktické možnosti	19
3.8	Hráči	20
3.8.1	Přestupová období	20
3.9	Sázkové kanceláře	20
3.9.1	Tvorba kurzu	20
3.9.2	Vliv dalších události na vývoj kurzu	21
4	Principy strojového učení	23
4.1	Úvod do strojového učení	23
4.2	Dělení algoritmů dle způsobu učení	24
4.2.1	Učení s učitelem	24
4.2.2	Techniky učení s učitelem	24
4.2.3	Učení bez učitele	25
4.2.4	Problém přeučení	26
4.3	Zisk a předzpracování dat	26
4.4	Srovnání modelů	27
4.4.1	Regresní model	27
4.4.2	Neuronová síť	30
4.4.3	Vícevrstvá neuronová síť	31
4.4.4	Samotné srovnání modelů	32
4.5	Využití strojového učení ve světě	33
4.6	Využití strojového učení	33
5	Návrh	34
5.1	Název a logo	34
5.2	Cílový uživatel	34
5.3	Souhrn funkcionality	35
5.4	Návrh systému	35
5.5	Návrh uživatelského rozhraní	36
5.5.1	Hlavní obrazovka	38
5.5.2	Obrazovka soutěže	38
5.5.3	Obrazovka týmu	39
5.5.4	Obrazovka hráče	39
5.5.5	Obrazovka zápasu	39
5.5.6	Obrazovka uživatele	39
5.5.7	Barevná paleta	40
5.6	Databáze	40
5.7	Komunikace s databází	40
5.7.1	Návrh REST API architektury	40
5.8	Simulace celého systému a simulační krok	42
5.9	SciKit-learn knihovna	42
5.9.1	Regrese v rámci knihovny	43
5.10	Systém predikce výsledků	43
5.10.1	Vstupy predikce	44
5.10.2	Výstupy predikce	44
5.10.3	Proces predikce	44
5.10.4	Návrh měření úspěšnosti predikce	45
5.11	Systém generování náhody	46

5.11.1	Faktory náhody	46
5.11.2	Konečný výsledek	47
6	Implementace	48
6.1	Získávání fotbalových dat	49
6.1.1	Zisk dat z webu	49
6.1.2	Zpracování dat	49
6.1.3	Integrace dat do databáze aplikace	49
6.2	Struktura statických a uživatelských dat	50
6.3	Architektura a závislosti systému	51
6.3.1	Řešení závislostí	51
6.3.2	Konfigurační soubory	52
6.4	Serverová část	52
6.4.1	Nastavení serveru	52
6.4.2	Node.js REST API	52
6.5	Implementace predikce zápasu	55
6.5.1	Zisk relevantních dat	56
6.5.2	Zpracování dat	57
6.5.3	Struktura celého modelu	57
6.5.4	Trénování modelu	58
6.6	Testování validity predikce a integrace	59
6.6.1	Rámcové ověření rozsahu výsledků	59
6.6.2	První iterace testování	59
6.6.3	Druhá iterace testování	60
6.6.4	Finální testování	62
6.6.5	Integrace do systému	62
6.7	Implementace simulátoru	62
6.7.1	Tok celé simulace	63
6.7.2	Ligový systém	63
6.7.3	Tvorba výsledku zápasu	64
6.7.4	Přepoččet mezi sezónami	65
6.7.5	Nulování uživatelské simulace	66
6.7.6	Uživatelské a asynchronní akce na popředí	66
6.7.7	Tvorba šablon a vizuální stránka aplikace	67
6.8	Použité nástroje pro vývoj	67
7	Testování	68
7.1	Uživatelské testování GUI	68
7.2	Testování průběhu simulací	69
7.2.1	Validace reálnosti	69
7.2.2	Výstup testování	70
8	Budoucí vývoj	72
9	Závěr	73
	Literatura	74

Kapitola 1

Úvod

Tato práce se zabývá vytvořením webového simulátoru fotbalového světa, založeného na reálných a současných datech. Simulátor jako takový je unikátní, protože žádné z existujících řešení se nezaměřuje hlavně na simulaci bez vlivu uživatele. Cílem je zpracovat uživatelsky přívětivé, jednoduché, ale zároveň komplexní řešení, které bude vytvářeno modulárně a do budoucna bude rozšiřitelné. Odhadování zápasů je řešeno s využitím polynomiální regrese. Výsledek predikce je pak ovlivněn faktory náhody tak, aby se simulace dynamicky vyvíjela a předpovídala nečekané výsledky, které se i v reálném světě dějí na základě různých vlivů. Zároveň má simulace přinášet i zábavu uživateli, pokaždé se tedy může vyvinout úplně jinak. Řešení je bez omezení, tudíž uživatel může simulovat svůj alternativní svět bez omezení roku. Může si vybírat oblasti, které ho zajímají, každopádně i nevybrané země mají stejný vliv na hru a vyvíjejí se stejným tempem. Každá nová hra by měla odrážet i aktuální pravidla, čili v provozu bude nutné simulátor frekventovaně aktualizovat a upravovat. V současné době jsou soupisky aktuální k sezóně 2017/2018, v níž začal vývoj systému. Vývoj simulátoru odevzdáním práce nekončí, je naplánován postup dalších řešení na příští dva roky. Přístup v současné verzi bude na vyžádání, členství je bezplatné.

Důležité pro tvorbu systému bylo pochopit fungování reálného fotbalového světa a zvolit vhodné a klíčové faktory ovlivňující simulaci a zápasy. Fungování celého světa je ale nevyzpytatelné a velkou roli hraje také psychika, morálka i náhoda. Tyto faktory náhody pak byly také zahrnuty do generování výsledku.

V kapitole 2 popisují motivaci pro tvorbu této práce, zároveň pak hodnotím existující řešení, srovnávám s mým řešením a nakonec popisují přínosy a cíle výsledku práce. V kapitole 3 pak stručně popisují fakta fotbalového světa, především v oblastech ovlivňujících vývoj simulátoru a sběr dat, ale také pár zajímavostí, které jsou také pro práci podstatné. V kapitole 4 pak popisují teoreticky strojové učení a jeho modely. V kapitole 5 pak je uveden návrh řešení a všech jeho klíčových podčástí. V kapitole 6 je pak popsána samotná implementace těchto podčástí. Kapitola 7 obsahuje oblast testování projektu. A v kapitole 8 je popsán plán a rozsah dalších rozšíření vzhledem k budoucímu vývoji.

Kapitola 2

Úvodní analýza

V této kapitole popíši moji motivaci k výběru tématu, analýzu tématu, zhodnotím a ukáži existující řešení a popíši přínosy celé této práce do tohoto odvětví. Tato kapitola je důležitá – je potřebné navrhnout a implementovat simulátor tak, aby byl jedinečný a tím pádem i potenciálně úspěšný. Z této analýzy pak bude vycházeno i v ostatních kapitolách a slouží jako základní stavební kámen práce.

2.1 Motivace pro tvorbu simulátoru

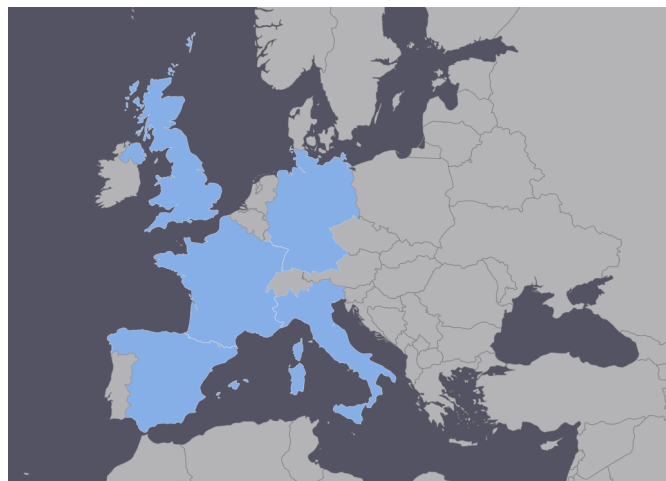
Cílem je vytvořit fotbalový simulátor, kde ale uživatel nebude přímým účastníkem trénování mužstva a nebude tak uměle zasahovat do celé simulace. Například ve velmi reálné hře Football Manageru¹ jsem si experimentálně bez výběru týmu pustil simulaci 100 let evropského fotbalu. Bylo zajímavé sledovat vývoj lig, kupříkladu boje o český titul v 90. letech 21. století mezi týmy Loko Vltavín a Lokomotiva Praha. Ve všech hrách podobného charakteru je stejná charakteristika – přestože si hráč nevybere žádný tým, pořád nějakým způsobem simulaci ovlivňuje (třeba jen tím, že hra uměle více vyvíjí jen ligy, které jsou nastaveny jako zapnuté). Před pár lety jsem tedy přišel na nápad, že bych chtěl vytvořit nezávislý simulátor, který by co nejvíce realisticky, ale zároveň i částečně náhodně z hlediska faktoru zábavy generoval alternativní budoucnost fotbalu. Součástí diplomové práce pak je implementace pěti největších národů dle koeficientů UEFA, graficky znázorněny jsou na obrázku 2.1. Soupisky jsou aktuální vzhledem k sezoně 2017/2018, při níž jsem s vývojem začal.

Jistá menší řešení jsem si pro sebe již v minulosti vytvořil (drobný simulátor fotbalu nebo závodů Formule 1). Chtěl jsem ale vytvořit komplexní a reálný simulátor, který by šlo časem více a více zdokonalovat a případně s ním i uspět na trhu. Konkrétní cíle a přínosy pak popisují v kapitole níže.

2.2 Analýza tématu

Simulace fotbalu je téma, které je obsaženo ve více projektech, především počítačových hrách. Tyto hry se ale spíše zaměřují na krátkodobá období a především záleží na krocích a aktivitě hráčů. Toto je v pořádku vzhledem k tomu, že se jedná o zábavný prvek a tyto hry cílí na zábavu a možnost dopřát hráči zasáhnout do fotbalového světa. Co se týče jiné oblasti, vzniklo pár diplomových, či bakalářských závěrečných prací, které se soustředily

¹<https://www.footballmanager.com>



Obrázek 2.1: Mapa národů, jejichž nejvyšší soutěže jsou implementovány v rámci práce

na odhadování a generování výsledků zápasů (viz kapitola 2.3.4). Tyto výsledky byly ale generovány pro maximální realističnost a končily vždy stejně, protože sloužily ke vygenerování potenciálního výsledku pro účely budoucího sázení. Mé řešení by mělo být průsečíkem těchto dvou oblastí. Z jedné strany by mělo generovat zábavu pro uživatele/hráče, zároveň by ale hráč neměl zasáhnout do vývoje týmu a simulace by měla být podle původních parametrů. Takové řešení dosud na trhu neexistuje.

Především hry fungují jako desktopová řešení, jejich hraní pak vyžaduje určité hardwarové parametry a určitou verzi systému. Mé řešení naopak bude jako webová aplikace, uživatel se tedy bude moci do svého účtu přihlásit z jakéhokoliv zařízení s prohlížečem, nebude tedy limitován konkrétním zdrojem a zároveň všechna odsimulovaná data budou uložena v databázi. Především starší fotbalové manažery měly hry ukládány lokálně, při jejich ztrátě pak hráč přišel o veškerý uložený postup. Football Manager teprve před pár lety umožnil ukládání postupu skrze cloud na platformě Steam.² Průsečík existujících řešení a mého návrhu je pak přehledně vidět na obrázku 2.2.

2.3 Existující řešení využívající simulaci výsledků

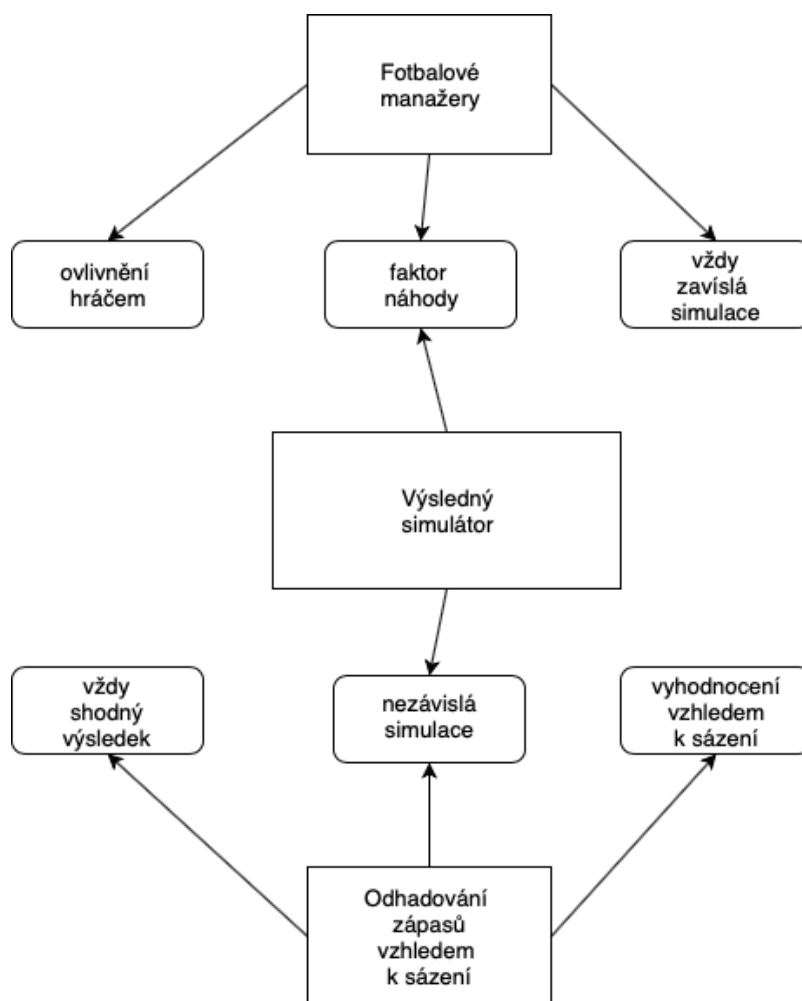
V této kapitole popíši jednotlivá existující řešení, uvedu základní informace o řešeních a především se zaměřím na část simulování zápasů a průběhu turnajů a lig. Zároveň pak uvedu v čem se produkt liší od vyvíjeného simulátoru.

2.3.1 Football Manager

Football Manager je série videoher, kde hráč plní roli fotbalového manažera. První ročník vyšel v roce 2004 a hru vyvíjí japonská společnost SEGA.³ Hra (současný ročník 2019) je dostupná na platformách Microsoft Windows, Mac OS a Nintendo Switch. Všeobecně je hodnocena jako nejvěrnější simulátor reálného fotbalového světa. Hra se prodává v krabicové verzi.

²<https://store.steampowered.com>

³<https://www.sega.com/>



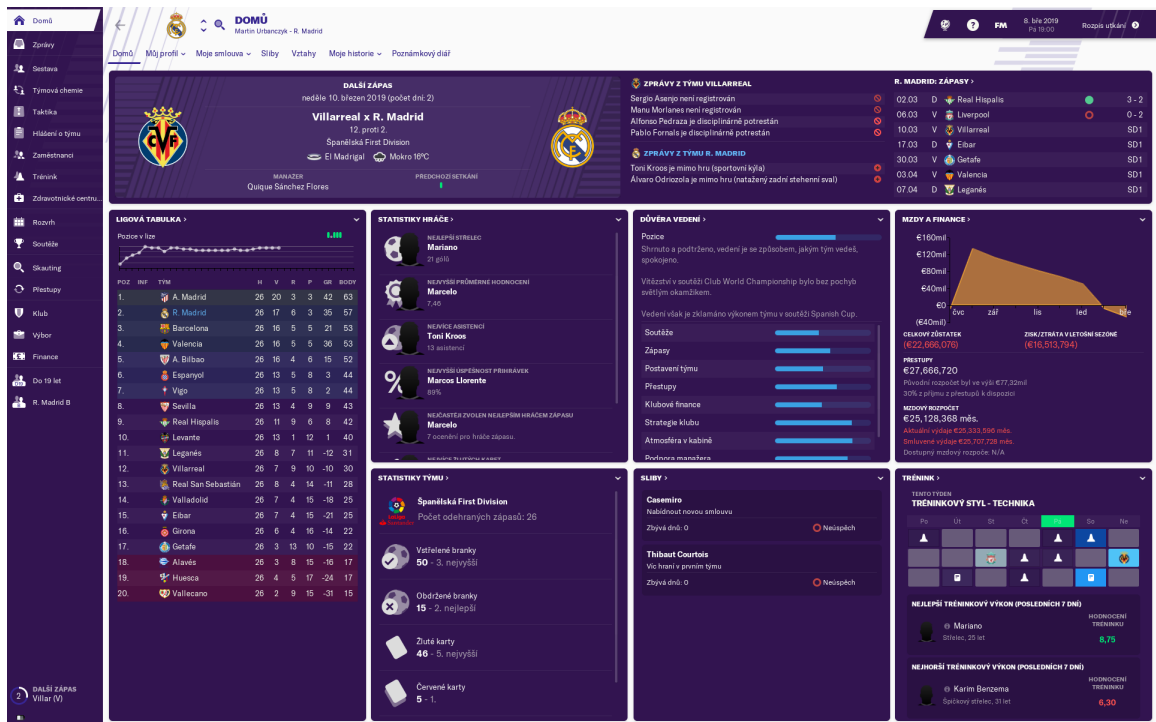
Obrázek 2.2: Srovnání existujících typů, průsečík u simulátoru

Jak jsem zmiňoval výše, tak vývoj simulace velmi ovlivňuje to, které ligy si hráč zapne, případně kde působí. Několikrát jsem zkusil pustit simulaci světa bez aktivního manažera. V zemích, které byly zapnuté se výrazně měnily výsledky a vítězové. Naopak v zemích, které byly jen k nahlédnutí, vítězily stále stejné týmy (ve Španělsku Real Madrid a FC Barcelona, v Německu Bayern Mnichov). V mém simulátoru bude vývoj lig všude stejným tempem nehledě na to, které ligy si uživatel vybere k bližšímu sledování. Zároveň pak hráč nebude zasahovat do vedení týmů, zasahovat do výběru týmů a dalších věcí spojených s vedením celku. Účelem je totiž simulovat celý svět bez umělých zásahů. Částečně se u hry inspiroji při vytváření designu webové aplikace.

2.3.2 Czech Soccer Manager

Czech Soccer Manager⁴, zkráceně dále CSM, je manažerská hra do českého tvůrce Petra Vašíčka. Hra je oproti Football Manageru jednodušší, obsahuje menší množství soutěží a také méně reálnou simulaci. Poslední finální verze vyšla v roce 2004, od té doby ale každý

⁴<http://csmweb.net/>



Obrázek 2.3: Úvodní dashboard v aplikaci Football Manager

rok vychází aktualizace soupisek. Hra je zdarma a je podporována pro operační systém Microsoft Windows.

CSM nemá až tak reálnou simulaci výsledků jako například Football Manager. Navíc je omezena na 16 sezon a hru nelze zapnout bez vedení týmu. Chtěl bych se ale inspirovat její jednoduchostí. Hra totiž nesimuluje spoustu věcí kromě výsledků (náladu v kabině, rozhovory, tiskové konference, ...), tudíž se hráč může soustředit především na svůj tým, taktiku a jednotlivé zápasy. Jednoduchost této hry je taky částečnou inspirací pro jednoduchost simulátoru.

2.3.3 FIFA

FIFA⁵ je fotbalový simulátor, vyvíjený společností Electronic Arts.⁶ První ročník vyšel v roce 1994 a hra se prodává v krabicové verzi. Hra je oproti výše zmíněným jiná. Soustředí se totiž na zápas jako takový a co nejvěrnější herní zpracování fotbalového zápasu. Obsahuje ale i módy turnajů a kariéry, kde simuluje výsledky a proto ji zde také zmiňuji.

Z podstaty hry vyplývá, že v mém simulátoru nebudou řešeny jednotlivé prvky zápasu a ovládání hráčů. Simulace sezón a výsledků je zde poměrně věrná, nicméně se zjevně nepočítá s velkým množstvím sezón. Obecně lze ale vidět, že hra se vydává spíše arkádovým stylem a velmi záleží na vlivu hráče. Tato hra je od simulátoru poměrně vzdálena, proto zde není příliš oblastí, kde se inspirovat pro vývoj.

⁵<https://www.easports.com/cz/fifa>

⁶<https://www.ea.com/cs-cz>

2.3.4 Existující diplomové práce

Existuje řada závěrečných prací, které se zabývají odhadem zápasu, případně oblastmi, které ve své práci využiji, proto je zde alespoň ve stručnosti zmíním. Práce se nezaměřují na souvislou simulaci, spíše odhad výsledku konkrétních zápasů, často pro účely sportovního sázení a odhadování kurzů. Přesto lze v těchto pracích nalézt řadu užitečných informací.

Odhadováním výsledků pomocí techniky neuronových sítí se zabývá práce od Vojtěcha Sokola [54]. Autor se zaměřil také na odhadování hokejových zápasů, což ale není problém této práce. Zároveň práce vznikla se účelem lepšího odhadnutí kurzů s cílem zvýšit zisk v sázení. Zajímavá je také diplomová práce od Jaroslava Bezděka [28], ten se zaměřil konkrétně na odhad vstřelených branek, kde využil data z Premier League z řady historických sezón. Byť je zde jen vzorek dat z fotbalového světa, poznatky z práce lze také částečně využít v mé práci a je důležité ji zde zmínit.

Další závěrečné práce se nezabývají konkrétně odhadem výsledků, každopádně zahrnují informace, které bych rád využil. Adam Blažej ve své diplomové práci [30] například popisuje vliv investic na výsledky fotbalových týmů v Premier League. V současné době jsou peníze ve fotbale více a více důležité, proto je třeba i v simulátoru s rozpočty celků kalkulovat i v dlouhodobějším měřítku. Tomáš Selinger zase ve své diplomové práci [52] popisuje a diagnostikuje psychické vlastnosti ve fotbale. Tyto zpravidla motivační faktory mohou generovat výraznější nepředvídatelnost zápasů.

2.4 Přínosy a cíle řešení

Simulátor bude unikátní produkt. Bude se zaměřovat čistě na odhadování a simulování výsledků a průběhů jednotlivých ročníků. Hlavní fotbalové manažery touto oblastí také zabývají, je ale zřejmé, že to není jejich hlavní zaměření. V mé práci půjde hlavně o simulaci. Zároveň bych chtěl odstínit vliv hráče, přestože zde bude faktor náhody, tak by simulace měla být řízena pouze umělou inteligencí a systémem. Cílem je vytvořit komplexní fotbalový simulátor, který bude brát všechny výsledky stejnou váhou, bude přístupný jednoduše skrze prohlížeč všem uživatelům a bude nenáročný na hardware. Zároveň se bude ale snažit co nejlépe tvořit alternativní budoucnost fotbalového světa s prvkem náhody.

Kapitola 3

Fotbal, fotbalové ligy a turnaje

Pro celou práci a simulátor jako takový je velmi důležité prozkoumat, pochopit a zachytit fungování fotbalového světa. Především jde o oblast fotbalových lig a turnajů, jejich struktury a pravidel. V simulátoru by měly tyto ligy co nejvíce odpovídat realitě. Je tudíž nutné podchytit celou množinu detailů. V této kapitole se nejdříve krátce zaměřím na světový fotbal, vysvětlím základní údaje, pravidla a jednotlivé organizace zastřešující kompletní profesionální fotbal. Dále se pak zaměřím na jednotlivé ligy, především tedy ty které jsou obsaženy v první verzi simulátoru – tedy pro účely diplomové práce. Uvedu zde i obecné informace, které jsou využitelné u všech lig a turnajů. Tato kapitola je nutná pro modelaci celého systému.

3.1 Základní pravidla fotbalu

Fotbal je celosvětově nejznámější sport, čítající miliony profesionálních hráčů a amatérských hráčů.¹ Jeden fotbalový zápas trvá 90 minut (plus nastavení rozhodčího) a rozdělen je do dvou poločasů po 45 minutách. V zápase participují dva týmy po jedenácti hráčích a skupina rozhodčích (hlavní, čároví, brankoví, videorozhodčí). Cílem zápasu je vstřelit více branek než soupeř.

V ligových soutěžích a pohárových soutěžích založených na skupinách se za výsledky zápasů rozdělují body. Zpravidla se jedná o 3 body za výhru, 1 bod za remízu a 0 bodů za prohru. V případě vyřazovacích zápasů s odvetami se počítá skóre. V případě rovnosti skóre rozhodují vstřelené branky na hřišti soupeře. Pokud jsou skóre stejná (zrcadlově) v obou zápasech, pak následuje prodloužení a případně soutěž v pokutových kopech.

Rozhodčí má pravomoc udělit tresty za porušení pravidel hráči. Může udělit žlutou kartu za středně závažný přestupek. Tato karta nemá okamžitý dopad, slouží spíše jako varování pro napomínaného hráče. Při zisku dvou žlutých karet v jednom zápase pak rozhodčí uděluje hráči kartu červenou, která značí jeho okamžitý odchod ze hřiště a zákaz startu v dalším soutěžním zápase této soutěže (nebo napříč více soutěžemi). Rozhodčí může červenou kartu udělit i bez předchozí žluté za závažné přestupky. V případě faulu uvnitř pokutového území pak může rozhodčí udělit i pokutový kop, případě přímý nebo nepřímý volný kop hlouběji v poli. V případě, že končí míč mimo území hřiště, se uděluje vhazování či rohový kop opačnému týmu.

Videorozhodčí je pak nová role, sloužící především k přezkoumání klíčových záležitostí jako jsou penalty, červené karty a branky. Tento videorozhodčí nemá žádnou operativní

¹https://www.fifa.com/mm/document/fifafacts/bcoffsurv/emaga_9384_10704.pdf

moc, vydává pouze doporučení hlavnímu rozhodčímu. Tato role vznikla inspirací z jiných sportů jako jsou tenis a lední hokej a nutno také říct díky tlaku veřejnosti a po hrubých chybách rozhodčích z minulosti.

3.2 Světové organizace

3.2.1 FIFA

Mezinárodní federace fotbalových asociací (zkráceně typicky nazývána FIFA [1]) je hlavní řídicí organizace světového fotbalu. Federace má své oficiální sídlo ve švýcarském městě Curych.² Součástí jsou všechny národní federace (celkově 209 zemí). Mezi jednu z hlavních povinností této organizací patří pořádání Mistrovství světa ve fotbale, pravděpodobně největší fotbalové akce. V soutěžích FIFA nemohou působit jiné než členské země, každá země musí projít řádnou registrací a splnit odpovídající podmínky. Pod organizací FIFA pak stojí organizace pro jednotlivé kontinenty:

- **CONCACAF:** organizace pro Severní Ameriku
- **CONMEBOL:** organizace pro Jižní Ameriku
- **CAF:** organizace pro Afriku
- **UEFA:** organizace pro Evropu
- **AFC:** organizace pro Asii a Austrálii
- **OFC:** organizace pro Oceánii

3.2.2 CONCACAF

CONCACAF³ je řídicí organizace pro severoamerický fotbal. V současné době čítá 41 národních fotbalových asociací. Státy v této organizaci pak dělíme na tři zóny (severoamerickou, stredoamerickou a karibskou). Zároveň pak jsou členy i tři státy z Jižní Ameriky.

3.2.3 CONMEBOL

CONMEBOL⁴ je řídicí organizace pro jihoamerický fotbal. Přestože čítá jen 10 členských zemí, tak se jedná o nejstarší kontinentální organizaci. Sídlí v paraguayském městě Luque. Hlavní reprezentační soutěží je pak Mistrovství Jižní Ameriky, známé především pod názvem Copa América. Dvě hlavní klubové soutěže se nazývají Pohar Osvoboditelů (Copa Libertadores) - obdoba evropské Ligy Mistrů a Copa Sudamericana.

3.2.4 CAF

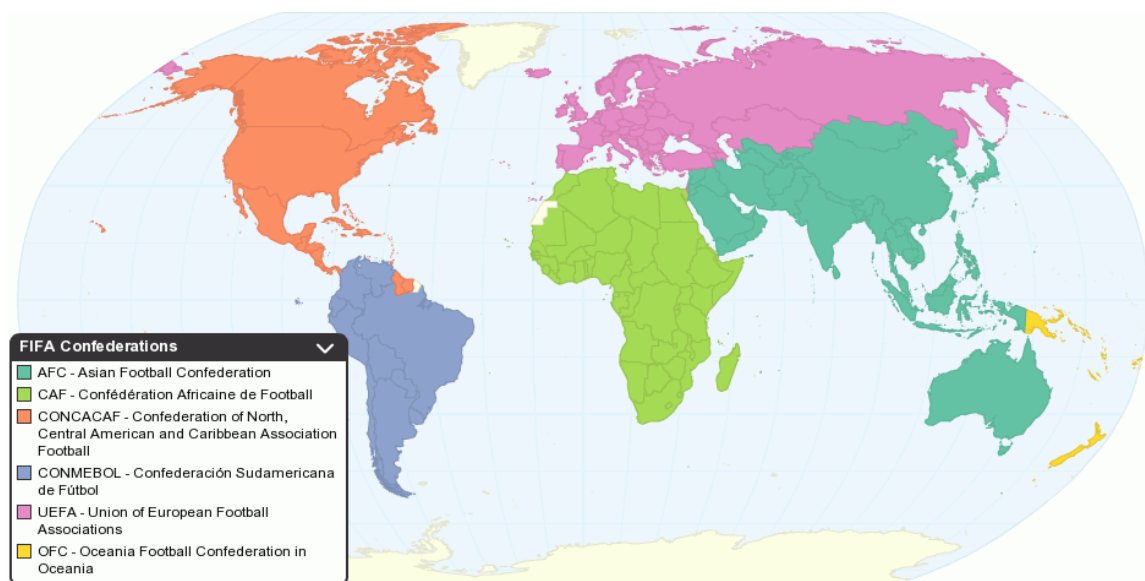
CAF⁵ neboli konfederace afrického fotbalu je hlavní řídicí organizace pro africký fotbal. Čítá 54 členských federací a patří k největším z těchto organizací. Klíčovou soutěží na Africkém kontinentu je Africký pohár národů, který se oproti ostatním kontinentálním soutěžím koná

²Místo, kde mají kanceláře nejvyšší představitelé.

³<https://www.concacaf.com>

⁴<http://conmebol.com>

⁵<http://www.cafonline.com>



Obrázek 3.1: Mapa působení konfederací FIFA

netradičně v zimě – z důvodu vysokých teplot na kontinentu v létě. Členské země CAF byly poměrně upozaděny, první Mistrovství světa totiž pořádala až v roce 2010 Jihoafrická republika.

3.2.5 AFC

AFC⁶ je hlavní řídicí organizací pro země Asiského kontinentu a Austrálii. Některé asijské země jsou ovšem členy UEFA. Jedná se buď o historické důvody (Izrael, Kypr) nebo o země na hranici obou kontinentů (Rusko, Turecko). Hlavní soutěží této organizace je pak Mistrovství Asie. Největšími velmocemi jsou pak země jako Japonsko nebo Jižní Korea.

3.2.6 OFC

OFC⁷ je hlavní řídicí organizace pro země Oceánie. Ze všech těchto konfederací má nejnižší důležitost. Na Mistrovství světa totiž nemusí jet ani jeden zástupce této organizace (vítěz kvalifikace ještě bojuje s jedním postupujícím z Asijské kvalifikace). Klíčovou soutěží je pak Oceánský pohár národů.

3.2.7 UEFA

UEFA⁸ je hlavní organizací pro evropský fotbal. Jedná se o konfederaci s nejvíce členy (55 členských zemí) a také nejvíce místy pro kvalifikanty na Mistrovství světa. Některé členské země nejsou suverénními státy (Skotsko, Faerské Ostrovy, ...).

UEFA pořádá Mistrovství Evropy (zkráceně EURO), turnaj se 24 členy. Tento turnaj se pořádá každé 4 roky. Od roku 2018 také Ligu Národů. Na klubové úrovni pak nejslavnější klubové soutěže – Ligu Místrů a Evropskou ligu. Detaily o těchto soutěžích pak níže.

⁶<http://www.the-afc.com>

⁷<https://www.oceaniafootball.com>

⁸<https://www.uefa.com/>

3.3 Typy soutěží

Ligy a poháry nelze slučovat do jedné kategorie. Nejzákladnější dělení je pak na tyto tři typy:

- **Národní soutěže:** účinkují zde kluby z jedné země.
- **Klubové soutěže:** účinkují zde kluby z více zemí.
- **Reprezentační soutěže:** účinkují zde výběry jednotlivých zemí a soutěží mezi sebou.

Je tedy nutné odlišit v jakých soutěžích participují kluby a platí zde veškerá klubová pravidla (jako například registrace hráčů, nemožnost přestupu mimo vymezená období atd.). Dalším typem soutěží jsou totiž reprezentační turnaje. Tyto soutěže se pak neprolínají co se týče například trestů za karty. Ovšem některé mezinárodní tresty platí skrze oba typy těchto soutěží, zde se jedná ale o vážné prohřešky jako jsou například penalizace za doping. Budu zde popisovat různá pravidla, ta se ale s časem mění, protože jednotlivé svazy i konfederace mají pravomoc více či méně i základní pravidla upravovat.

3.4 Národní soutěže

Každá národní federace si spravuje své ligové soutěže. Tyto soutěže jsou strukturovány do úrovní (1. liga, 2. liga, regionální soutěže, ...). Týmy navzájem postupují a sestupují mezi úrovněmi. Pro nižší soutěže pak většinou platí rozdělení do více skupin na stejné úrovni – typicky dle menších regionů. Většinou se jedná o amatérské soutěže a důvodem rozdělení je zamezení nadměrnému cestování mezi zápasy. Veškerou kontrolu nad soutěžemi má pak národní federace, která je v tomto ohledu autonomní. Musí se samozřejmě dodržovat stanovy FIFA a UEFA. V této kapitole budu především popisovat evropské soutěže, v simulátoru budou totiž v části pro diplomovou práci pouze Evropské země.

V nejvyšší soutěži pak (v Evropské zóně) část týmů postoupí do Ligy Mistrů, část týmů do Evropské ligy. Počty těchto postupujících se odvíjejí od výše národního koeficientu za posledních 5 let. Do Evropské ligy pak zpravidla postupuje i vítěz domácího, či ligového poháru.

Domácí pohár je pohárová soutěž (hraje se na jeden, či dva vítězné zápasy). V této soutěži se utkávají týmy z jedné země mezi sebou napříč různými ligovými úrovněmi. Je to tedy příležitost pro týmy z nižších soutěží nebo ty, které se z ligy nedostanou do evropských pohárů. V některých zemích pak existuje ještě ligový pohár, což je typ soutěže podobný domácímu poháru, ale je omezen na určitou úroveň dané země. Například v Anglii i z této soutěže (Carabao cup⁹) vítěz má jistou místenku v Evropské lize - toto ale nemusí být pravidlem.

Každá země si pak může pořádat ještě řadu dalších menších pohárů - většina evropských zemí pořádá alespoň domácí superpohár (před počátkem ligové sezóny soubor vítěze domácí ligy a vítěze domácího poháru). Ve větších federacích pak existují ještě například regionální poháry pro určité rozmezí úrovní ligových soutěží.

Níže pak popisuji jednotlivé národní soutěže, především pak ty, které budou obsaženy v diplomové práci, potažmo výsledném simulátoru. Zvolil jsem pět nejlepších lig dle současného žebříčku koeficientů.

⁹Řada soutěží nese název hlavního sponzora ve svém názvu.

3.4.1 Anglické soutěže

Nejvyšším řídicím orgánem anglických soutěží je FA (The Football Association¹⁰). Sídlo má na stadioně Wembley v Londýně a pod jeho správu spadají všechny fotbalové soutěže v této zemi a Anglická fotbalová reprezentace. Ostatní území v rámci Velké Británie (Skotsko, Severní Irsko, Wales) mají své vlastní asociace, soutěže a reprezentace. Některé týmy ale působí v Anglických soutěžích (Cardiff nebo Swansea z Walesu). Zároveň jsou zastoupeny každý sám za sebe v rámci soutěží UEFA. Asociace vznikla jako první asociace na světě, Anglie jako taková je díky tomu označována jako domov a kolébka fotbalu. Čestným předsedou asociace je člen britské královské rodiny (v současnosti princ William).

Nejnámější soutěží je Premier League. V této nejvyšší soutěži participuje 20 týmů, v současné době (sezóna 2018/2019) postupují 4 týmy do Ligy Mistrů a 3 týmy do Evropské ligy. Zároveň 3 nejhorší týmy sestupují do nižší soutěže. V průběhu období srpen až květen se odehrává celkově 38 kol. Každý tým se utká se všemi ostatními týmy v lize. Jednou utkání probíhá na domácím stadionu a jednou na stadionu hostí. Nejvíce titulů získal tým Manchester United FC (dosud 20). Mnozí nazývají Premier League nejprestižnější a nejnáročnější liguovou soutěží na světě (byť dle koeficientů je na vrcholu soutěží španělská La Liga).

Domácí pohár se nazývá FA Cup a je nejstarší fotbalovou soutěží na světě - první ročník proběhl již v sezóně 1871/1872. Soutěž se skládá z řady kol, týmy z vyšších soutěží nastupují až v pozdních fázích turnaje. Vítěz postupuje do Evropské ligy a finálový zápas se odehrává na stadionu anglické reprezentace - ve Wembley. Historicky nejúspěšnějším týmem je londýnský Arsenal FC (13 vítězství).

V Anglii je také Ligový pohár, v současné době mající název Carabao cup. Jeho vítěz se také kvalifikuje do Evropské ligy. Systém je podobný výše zmíněnému poháru se sníženým počtem účastníků z nižších lig. Pohár nemá takovou prestiž jako FA Cup.

Vítězové soutěží FA Cup a Premier League se v srpnu utkávají v soutěži Community Shield na jeden vítězný zápas [34]. Asociace pak spravuje dále řadu nižších soutěží, ženských soutěží a regionálních pohárů. Mezi historicky nejúspěšnější týmy patří Manchester United, Arsenal FC, Liverpool FC a Chelsea FC. Je nutné zmínit, že anglické celky jsou díky zvýšení sponzorských a televizních práv ekonomicky nejsilnějšími a v přestupových obobích celkově pravidelně nejvíce utrácí .

3.4.2 Španělské soutěže

Hlavním řídicím orgánem spravujícím španělské liguové a pohárové soutěže je španělská fotbalová federace, která sídlí nedaleko hlavního španělského města – Madridu. Mimo správu soutěží v zemi je tato organizace také zodpovědná za jmenování manažerů pro mužskou, ženskou a veškeré mládežnické reprezentace.

Nejvyšší liguovou soutěží v zemi je Primera División (nazývaná také jako La Liga). Podobně jako v Anglii 4 týmy postupují do Ligy Mistrů, 3 pak do Evropské ligy a nejhorší 3 týmy padají do nižší liguové soutěže. Nejvíce titulů drží katalánský tým FC Barcelona (25). Systém zápasů je shodný s Premier League a počet týmu je také stejný - 20 [13].

Domácí královský pohár se nazývá Copa del Rey¹¹. Nejsilnější celky do něj vstupují až v průběhu ledna, když v domácí lize probíhá krátká přestávka. Od 4.kola po semifinále se hrají zápasy systémem doma-venku, postupuje ten s lepším skóre. Ve Španělsku není

¹⁰<http://www.thefa.com>

¹¹<https://www.laliga.es/en/other-competitions>

žádný ligový pohár, jen klasický Superpohár před začátkem ligové sezóny. Zajímavostí je, že v nižších soutěžích narozdíl od jiných zemí (opět například Anglie) plnohodnotně figurují i rezervní týmy. Celky tak mají možnost nechat své mladé a rezervní hráče rozehrát i v profesionálních soutěžích. Nejslavnějšími celky ze Španělska z historického hlediska jsou především FC Barcelona, Real Madrid CF, a Atlético de Madrid.

3.4.3 Italské soutěže

Italská fotbalová federace je nejvyšším řídicím orgánem v Itálii. Nejvyšší italská ligová soutěž se jmenuje *Seria A*, účastní se jí 20 celků a systém postupů a sestupů je stejný jako v Anglii a Španělsku. Označení nižších soutěží je pak pomocí dalších písmen (*Seria B, C, ...*). Typický výraz pro získání mistrovského titulu je *scudetto*¹². Obecně se pak italská soutěž označuje jako velmi takticky svázaná a orientovaná na defenzivu. Historicky neúspěšnějším týmem je Juventus Turín se 34 tituly. Jako další slavné celky lze zmínit AC Milán, Internazionale FC a SSC Neapol [14].

Coppa Italia je domácím pohárem v Itálii. Hraje se podobným systémem jako poháry ve výše zmíněných zemích. Vítěz ligy a poháru se utká v Supercoppa Italiana tradičně před začátkem ligy. Italské soutěže jsou nechvalně proslulé i korupčními skandály. Pravděpodobný vrchol těchto kauz byl v roce 2006 a skončil mnoha odečty bodů a vyřazením Juventusů do druhé nejvyšší soutěže.¹³ Od té doby se italský fotbal z krize sbírá a zejména na Evropské scéně zaostává za konkurenty ze Španělska.

3.4.4 Německé soutěže

Řídicí organizací Německého fotbalu je Německý fotbalový svaz (Deutscher Fußball-Bund). Nejvyšší soutěží je 1. Bundesliga. V této soutěži účinkuje 18 týmů. Do evropských pohárů postupuje stejný počet týmů jako ve výše zmíněných zemích. Rozdílný je pak sestup do nižší soutěže - 17. a 18. celek z tabulky sestupuje přímo a 16. celek se utká ve dvojzápasové baráži s 3. nejlepším celkem z druhé ligy. Nejúspěšnější celek je FC Bayern se 28 mistrovskými tituly. Dalšími slavnými celky jsou Borussia Dortmund, Schalke 04 a Bayer Leverkusen [3].

Domácí pohár se nazývá DFB Pokal. Zajímavé je, že celky z nejvyšší soutěže zde účinkují již od prvních kol, často tedy v těchto kolech vznikají vysoké a divoké výsledky. DFL Supercup je německý superpohár pro soubor mistra ligy a vítěze domácího poháru.

3.4.5 Francouzské soutěže

Francouzská nejvyšší soutěž se nazývá Ligue 1 a tento název má od roku 2002. Systém postupu do Evropských pohárů je jiný, protože Francie nemá tak vysoký koeficient. Do Ligy Mistrů postupují pouze tři celky, do Evropské ligy další tři. Soutěž čítá 20 celků, poslední tři pak sestupují do Ligue 2. V současné době je po herní i ekonomické stránce hegemonek tým Paris Saint Germain – od příchodu katarských šejků jako majitelů klubu v roce 2010. Nejvíce titulů z historického hlediska ovšem má AS Saint Étienne – celkem 10 titulů [10]. Francouzská reprezentace je pak současným držitelem titulů mistrů světa (z roku 2018). Mezi francouzské celky patří i slavný tým z Monaka – AS Monaco.

Francouzský fotbalový domácí pohár funguje stejným principem jako Španělský Copa del Rey. Mistr ligy a vítěz poháru se pak utkávají o Trophée des champions. Ve Francii se

¹²ocenění pro mistry italské ligy, viz. <https://en.oxforddictionaries.com/definition/scudetto>

¹³<https://sport.aktualne.cz/fotbal/kauza-juventus-2006-za-korupci-padly-do-zivotni-tresty/r~i:article:704215/?redirected=1545473725>

pak hraje i ligový pohár. Kromě týmů PSG a AS Monaco ale v posledních letech země čeká na výrazný průlom na klubové evropské scéně.

3.5 Klubové mezinárodní soutěže

V následujících podkapitolách se zaměřím na evropské klubové soutěže pod hlavičkou UEFA. Byť ostatní konfederace také mají své mezinárodní poháry bezpochyby žádný z nich nedostahuje kvalit a presitže soutěží evropských. Navíc se v diplomové práci a simulátoru zaměřuji na výše zmíněných šest zemí.

3.5.1 Koeficienty

Nasazení týmů z jednotlivých lig do evropských pohárů není dáno staticky, ale dynamicky se mění s vývojem národního koeficientu. Národní koeficient měří úspěšnost klubu z jednotlivých lig v evropských pohárech za posledních 5 sezón. Za výhru tým přičte 2 body a za remízu 1 bod. V předkolech se body půlí. Speciální bonusy jsou pak za postupy do základních skupin Ligy mistrů a Evropské ligy a za postupy ve vyřazovacích částech. Celkový zisk se pak dělí počtem zástupců. A tyto roční koeficienty se pak sčítají za posledních pět sezón¹⁴. V praxi tak velmi často pro nevyrované ligy je výhodnější mít v pohárech méně zástupců tak, aby nekonkurenceschopní zástupci nesnižovali průměr. V současné době (květen 2019) vede tabulku již dlouhodobě Španělsko s více než 19-ti bodovým náskokem. V letech 2014-2018 totiž z 10 možných titulů v Evropských pohárech získaly 9 titulů právě španělské celky [39].

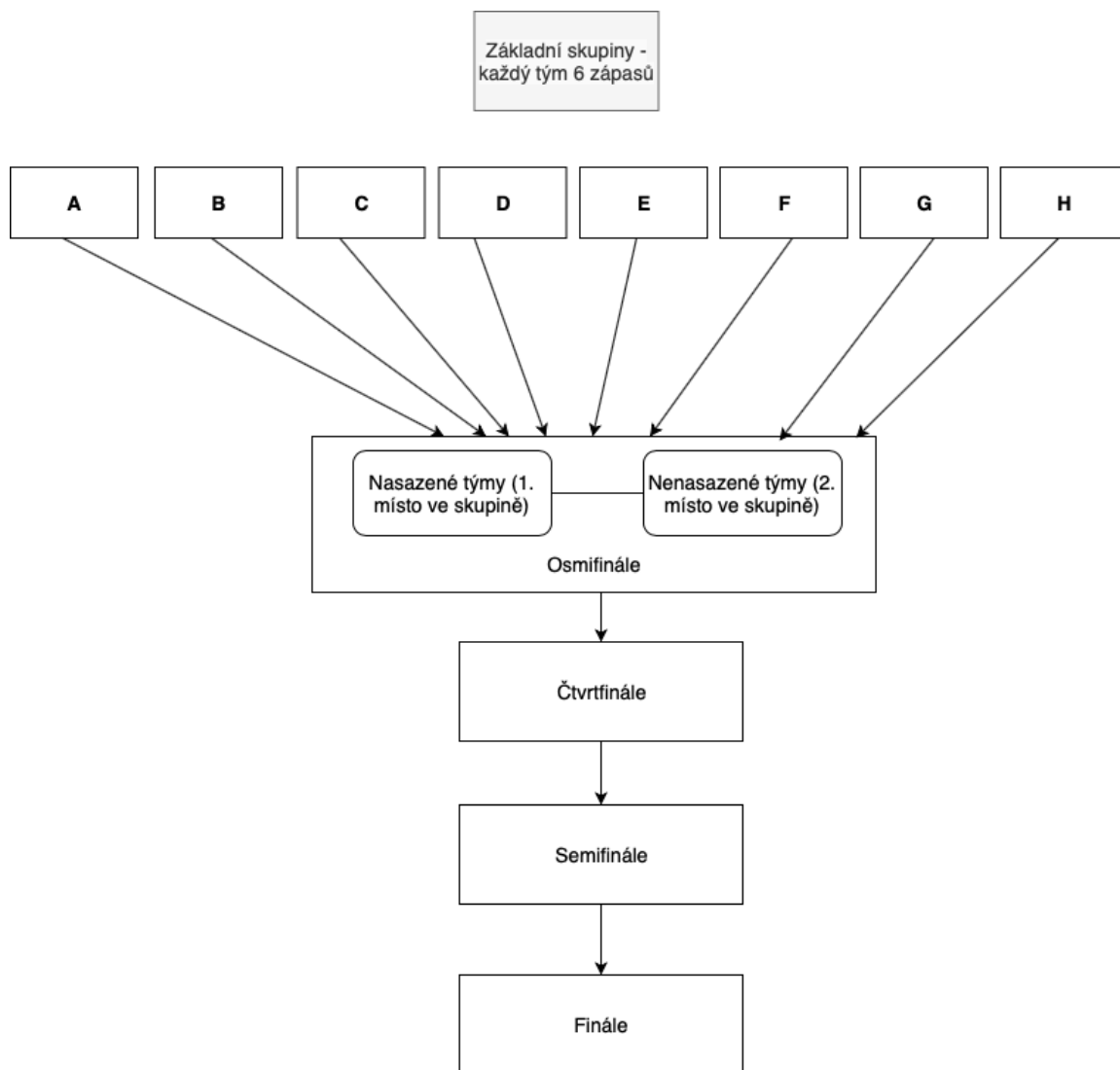
3.5.2 Liga mistrů UEFA

Liga mistrů je nejprestižnější soutěž, ve které se utkávají nejlepší celky z celé Evropy. Soutěž začíná již na konci června prvními předkoly (celkově jsou 4 předkola pro níže postavené země v tabulce koeficientů) a končí na konci května nebo začátku června. Po předkolech následuje základní fáze, kde se 32 celků utká v 8 skupinách. Do vyřazovací fáze postoupí z každé skupiny dva nejlepší, třetí tým se kvalifikuje do jarní fáze Evropské ligy. Od osmifinále do semifinále se hraje vyřazovací část, kde v každé fázi se celky utkají ve dvojzápase. Finálový zápas se pak hraje na předem daném neutrálním stadionu. Finále Ligy mistrů je jednou z nejprestižnějších událostí v celém sportovním světě. Nejúspěšnějším celkem je Real Madrid se 13 tituly. Do sezóny 1992/1993 se konal předchůdce Ligy mistrů - PMEZ (Pohár mistrů evropských zemí) [51]. Vítěz Ligy mistrů se v srpnu utkává s vítězem Evropské ligy o Evropský superpohár. Tato soutěž uvozuje celou fotbalovou sezónu.

3.5.3 Evropská liga

Evropská liga je druhou nejprestižnější soutěží v evropském fotbale. Stejně jako Liga mistrů obsahuje 4 předkola a také skupinovou fázi (zde ale 12 skupin po 4 týmech). Z každé skupiny také postupují dva nejlepší týmy. Ve vyřazovací fázi je doplní týmy, které vypadly z Ligy mistrů. Vyřazovací fáze obsahuje o jedno kolo více než Liga mistrů, jinak je systém podobný tomu v Lize mistrů. Vítěz soutěže kromě trofeje získá i místo v dalším ročníku Ligy mistrů, pokud se daný tým nekvalifikoval z domácí soutěže [21]. Soutěž v současné podobě zažila

¹⁴<https://www.uefa.com/memberassociations/uefarankings/country/#/yr/2019>



Obrázek 3.2: Systém Ligy mistrů

svou premiéru v roce 2009, do té doby se soutěž jmenovala Pohár UEFA (a předtím Pohár vítězů pohárů). Nejúspěšnějším týmem je španělská FC Sevilla s 5 tituly [6].

3.6 Reprezentační soutěže

V menších ligových přestávkách během sezóny se konají reprezentační srazy. Pro názornost uvažujme opět Evropský fotbal. Ten se dělí do dvou cyklů po dvou letech. Reprezentace napříč všemi úrovněmi se utkávají v kvalifikaci na Evropský šampionát a na Světový šampionát. Tyto kvalifikace na sebe navazují. Vždy je rozlosována řada skupin (liší se cyklus od cyklu) a část účastníků postoupí přímo, další část se utká v dodatečné baráži v listopadových termínech. Co se týče kvalifikací na světový šampionát, tak evropská část kvalifikace vypadá velmi podobně jako kvalifikace na kontinentální šampionát [7]. Každá konfederace má ale právo si svou kvalifikaci uspořádat podle vlastního pravidel. Finálový turnaj se pak koná vždy v létě pokud není dáno jinak (plánované zimní Mistrovství světa v Kataru v roce 2022). Koná se také řada přátelských utkání, v Evropě ale část z nich byla od roku 2018 nahrazena novou soutěží - Ligou národů [23].

Složení reprezentačních týmů se v průběhu cyklů mění, hráče na každou reprezentační přestávku a šampionát nominuje trenér nebo manažer reprezentace. Není tedy dán přesný postup výběru. Často se tedy stává, že velmi kvalitní hráč si za reprezentaci nezhraje, protože nesedí do současné taktiky, či konceptu hry. Hráči také obvykle v pozdějších letech kariéry ukončují reprezentační kariéru, velmi často z důvodu soustředění se na klubové záležitosti.

3.6.1 Mistrovství světa

Mistrovství světa je považováno za největší fotbalovou událost. Po kvalifikacích kde se rozhodne o naplnění kvót pro každý kontinent probíhá zhruba měsíční turnaj o 32 týmech. Týmy jsou rozlosovány předem dle výkonnostních košů do osmi skupin. Z každé skupiny postoupí dva celky a ty jsou nasazeny dle pozic do vyřazovacího pavouka. Ve vyřazovací fázi se pak týmy utkají na jeden zápas. Pokud skončí v základní hrací době nerozhodně, tak následuje prodloužení a poté soutěž v pokutových kopech. Vítěz získává slavnou trofej a na čtyři roky titul mistra světa ve fotbale. Historicky nejúspěšnější zemí je Brazílie se sedmi tituly [25].

Pořadatele mistrovství určuje hlasování na summitu organizace FIFA. Stává se tak již několik let před turnajem tak, aby pořadatel mohl připravit potřebnou infrastrukturu, případně dostavět sportoviště. Všichni kandidáti na pořadatelství musí před výběrovým řízením dostatečně prokázat svoji připravenost a schopnost připravit adekvátní podmínky [12]. Pořadatelé se mohou spojit a vytvořit vícenárodní pořadatelství (jako příklad Japonsko a Jižní Korea v roce 2002).

3.6.2 Kontinentální mistrovství

Každá konfederace si také pořádá svá kontinentální mistrovství. Konkrétně se jedná o tyto šampionáty:

- **Asie a Austrálie:** Mistrovství Asie
- **Afrika:** Africký pohár národů
- **Evropa:** Mistrovství Evropy

- **Jižní Amerika:** Copa América
- **Severní Amerika:** Gold cup
- **Oceánie:** Oceánský pohár národů

Systém každého ze šampionátů je jiný. Zde se opět zaměřuji na Evropu. Závěrečného turnaje se účastní 24 celků, které jsou rozděleny do šesti skupin. Z každé skupiny postupují do vyřazovací fáze dva celky. A ze všech skupin se pak po skončení základní fáze určí čtyři nejlepší celky na třetích místech, které postupující doplní. Vyřazovací fáze se pak hraje standartním turnajovým pavoukem dle pozice ve skupinách [22]. Vítěz každého z kontinentálních šampionátů postupuje na Kofederační pohár FIFA konající se před dalším světovým šampionátem na místě budoucího pořadatele.

3.7 Složení fotbalového týmu

Fotbalový tým se skládá z maximálně 30 hráčů, limity pro registraci do soutěží jsou ale většinou nižší (typicky 23 hráčů). Navíc musí být splněny různé podmínky. Například v klubu musí být určitý počet odchovanců nebo počet cizinců mimo Evropskou unii nesmí přesáhnout daný počet¹⁵. Tyto podmínky si určuje každá federace zvlášť. Společné podmínky jsou pak pro evropské poháry. Kromě hráčů pak ještě v týmu je řada dalších pozic. Jedná se o manažera, trenéry, fyzioterapeuty, lékaře, skauty a mnoho dalších pozic. Zde nejsou (kromě trenérů) žádné speciální kvóty a kluby si mohou zaměstnat kolik členů týmu chtějí.

3.7.1 Sestavy a pozice hráčů

Každý hráč má svoji pozici. Nejhrubější rozdělení je na brankáře, obránce, záložníky a útočníky. Ovšem uvnitř každé této skupiny mohou být absolutně rozdílní hráči s jinými rolemi. Jako příklad vezmeme defenzivního záložníka a ofenzivního záložníka. Defenzivní záložník působí jako štít a velmi často by se měl stahovat až mezi střední obránce a napadat soupeřovu rozehrávku. Ofenzivní záložník ale je na druhé polovině hřiště a poskytuje podporu útočnickům a křídům. Rozdělení pozic je následovné [8]:

- **Brankář:** jen jedna pozice
- **Obránce:** krajní obránce, křídelní obránce, stoper, střední obránce
- **Záložník:** defenzivní obránce, střední záložník, krajní záložník, ofenzivní záložník
- **Útočník:** střední útočník, křídlo, falešná devítka

3.7.2 Taktické možnosti

Role hráčů jsou velmi důležité pro trenérovu tvorbu taktiky. Ovlivňují rozestavení hráčů na hřišti. Pro hru celého týmu je zásadní určit takové rozestavení a pokyny, aby hráčům tento herní styl seděl a ideálně všichni hráli na jejich přirozených pozicích. Během zápasu se samozřejmě taktika mění podle stavu. Trenér může zasáhnout do taktiky střídáním, kdy může zdánlivě vyměnit hráče post za post, ale přesto může změnit obraz celé hry. Nejčastější rozestavení ve fotbalovém světě je 4-4-2 (4 obránce, 4 záložníci, 2 útočníci).

¹⁵<https://latitudelaw.com/news/the-fas-policy-on-non-eu-football-players-and-possible-implications-following-brexit/>

Taktika má zásadní vliv na předváděnou hru a výsledky celého týmu. Taktika má také vliv na nákupy a prodeje hráčů, velmi často pokud přijde do týmu nový trenér, který sází na určitý stav, tak si okamžitě přivede hráče, kteří právě do tohoto stylu vyhovují.

3.8 Hráči

Každý profesionální zaměstnaný hráč je v klubu vázán smlouvou. Nemůže tedy svévolně tým opustit. Buď tedy může být odkoupen po domluvě obou klubů nebo může počkat na vypršení smlouvy a odějít na základě *Bosmanova pravidla* [9]. Třetí možností je pak (pokud bylo předem domluveno) složení výstupní klauzule. Výstupní klauzule je předem daná částka, po jejímž zaplacení se kluby nemusejí spolu domlouvat a zájemce se rovnou může pustit do jednání s hráčem.

3.8.1 Přestupová období

Přestupovat se může jen v určitém období v roce. Toto přestupové období si určuje národní organizace sama, ovšem do Evropských pohárů jsou data registrace, po nichž již není možné přidat na soupisku další nové hráče. Tradiční je přestupové období od 1. do 31. ledna v zimě a od 1. července do 31. srpna v létě [9]. Většinou ale kluby více nakupují v létě před novou sezónou a v zimě jen doplňují kádr. Tradičně nejaktivnější jsou v přestupových obdobích anglické celky, zejména kvůli ekonomické síle i papírově slabších celků.

3.9 Sázkové kanceláře

V rámci práce je také potřeba analyzovat o jak vyrovnaný zápas jde a na základě toho určit úspěšnost predikce. Proto bylo nutné analyzovat a pochopit problematiku samotné tvorby kurzu bookmakerem a postupný vývoj tohoto kurzu. V České republice mezi nejznámější sázkové kanceláře patří vzhledem k sázení na fotbalové zápasy Tipsport¹⁶, Fortuna¹⁷ a Chance¹⁸. Každá společnost zaměstnává své bookmakery a kurzy se tudíž mohou (byť nevýrazně) lišit. Pro sázkaře je tak výhodné sledovat více než jednu kancelář pro nalezení pro něj nejvýhodnějšího kurzu.

3.9.1 Tvorba kurzu

Bookmaker je osoba zodpovědná za nastavení kurzů v sázkové kanceláři. Cílem této osoby je vhodně nastavit kurzy u dané události vzhledem k dlouhodobé ziskovosti kanceláře. K výpočtu kurzu pak bookmaker přidává marži (o ní snižuje dané kurzy). Tato marže se může lišit na základě různých faktorů. Příkladem těchto faktorů může být třeba soutěž, pro jejíž zápas je kurz tvořen. Typicky jsou tyto osoby alokovány na určité sportovní (či nespportovní) oblasti, případně na konkrétní soutěže či zápasy.

Samotné kurzy se pak určují nejen na výhru, remízu, či prohru daných týmů. Dalšími faktory totiž může být například počet vstřelených branek, kurzy na jednotlivé střelce, počet faulů, žlutých karet a mnoha dalších. Rozsáhlost této množiny je pak typicky dána konkrétním zápasem, jeho potenciální oblíbeností a množstvím dat, kterými k němu bookma-

¹⁶<https://www.tipsport.cz>

¹⁷<https://gm.ifortuna.cz>

¹⁸<https://www.chance.cz>

ker a společnost disponuje. Je tedy pravděpodobné že na zápas play off Ligy mistrů bude vypsáno více druhů sázek než například na zápas nejvyšší albánské soutěže.

Zde popisují postup výpočet kurzů na odhad výhry, remízy, či prohry týmů v daném zápase. Samotný výpočet se pak skládá z těchto činností [2] :

1. **Pravděpodobnostní kalkulace:** bookmaker si utvoří pomocí svých dat, programů a vlastní intuice pravděpodobnosti daných výsledků.
2. **Výpočet kurzů:** z vypočítaných pravděpodobností se pak snaží vytvořit férové kurzy, typicky pak používá vzorec v rovnici 3.1.
3. **Snížení kurzů o marži:** poté bookmaker snižuje kurz o danou marži (pravděpodobný rozsah je 2% – 10%, zpravidla čím nižší marže je určena, tím více se ji bookmaker jist výpočtem daných pravděpodobností.
4. **Ochrana proti rizikovým sázkám:** aby bookmaker zabránil ztrátě, reguluje tuto marži dále. Kurzy na významné evropské soutěže mají snížení o marži méně výrazné než události v méně sledovaných sportech, či odvětvích, o nichž nemusí mít bookmaker tolik informací.

$$\text{odd} = \frac{100}{\text{probability}} \quad (3.1)$$

3.9.2 Vliv dalších události na vývoj kurzu

Ve výše uvedené sekci popisují počáteční tvorbu kurzu. Nicméně další důležitou oblastí je samotný vývoj kurzu mezi prvotním vytvořením a začátkem konkrétní události. Vlivy na změnu kurzu by se daly rozdělit do tří kategorií: **změny v samotných týmech, vliv sázkařů** a **vliv konkurence** [5]. Obecně lze pak říci, že změna nemá povětšinou vliv na rozlišení, jestli má zápas jasného favorita, větší změnu ale může způsobit u poměrně vyrovnaných zápasů.

Vliv změny v týmech

Typickým příkladem ovlivnění kurzů je zranění klíčového hráče. Zde je to na samotném bookmakerovi o kolik se kurz změní. V potaz se v takovém případě bere i forma tohoto hráče v posledních zápasech. Bookmaker navíc často dopředu neví, kteří hráči nastoupí do oficiálního oznámení sestav týmů. Toto oznámení je pak vydáno desítky minut před výkopem zápasu (je dáno protokolem každé soutěže).

Vliv sázkařů

Kanceláře také mají k dispozici počty sázejících na jednotlivé události v reálném čase. Kurz se velmi drobně snižuje při zvyšujícím se počtu sázek. Cílem tohoto snižování je nejdříve nalákat sázející a se zvyšujícím se počtem pak zvyšovat marži. Samotní sázkaři většinou čekají co nejdéle vzhledem k výkopu, čili i kurz se drobně snižuje po celou dobu vypsání [5].

Vliv konkurence

Vzhledem k tomu, že kurzy sázkových kanceláří jsou veřejné, tak tyto kanceláře navzájem monitorují rozdílnot mezi svými kurzy. Jsou pak schopny v reálném čase uzpůsobovat své kurzy konkurenci tak, aby nepřicházeli o sázkaře. Tím pádem dochází k regulaci kurzu. Je proto téměř nemožné pro konkrétní sázkovou kancelář nabízet dlouhodobě výhodnější kurzy než konkurence, aniž by byla ve ztrátě [5].

Kapitola 4

Principy strojového učení

V této kapitole stručně popíši metody a principy strojového učení založené především na technice učení s učitelem. Zároveň zde srovnám některé modely. Konkrétní implementace pak bude popsána v kapitole věnované implementaci. V simulátoru pak bude implementován model pro predikci, která se natrénuje na historických výsledcích zápasů, některé vstupy budou z databáze a vliv na predikci mají také starší výsledky dostupné ze služby API FOOTBALL.

4.1 Úvod do strojového učení

Strojové učení je proces, který zkoumá principy, které umožňují strojům učení. Učením se v tomto případě myslí ekvivalent lidského učení. Člověk prováděním jakékoliv činnosti získává zkušenosti a na základě nich je schopen danou činnost provádět lépe. Stejně tak stroj by měl provádět další iteraci svého algoritmu nebo činnosti v další iteraci již lépe a být tím pádem efektivnější na základě svých zkušeností. Tím pádem je schopen pracovat optimálněji. Strojové učení je tedy velmi úzce spojeno s oblastmi umělé inteligence (je jeho podmnožinou) a kybernetiky. V praxi se tedy strojové učení dá využít i pro částečné nahrazení lidské práce například u pásů v továrnách a obecně jsou možnosti využití těchto technik prakticky neomezené.

Oblasti, které jsou se strojovým učením úzce spjaty jsou například také oblasti dolování dat, optimalizace a statistika. Základní dělení je pak na učení s učitelem a učení bez učitele (dalšími oblastmi, které se uvádějí je pak kombinace těchto metod a zpětnovazební učení). Toto dělení je podle způsobu učení, existuje pak ještě například dělení dle způsobu zpracování (dávkové, či inkrementální) [37].

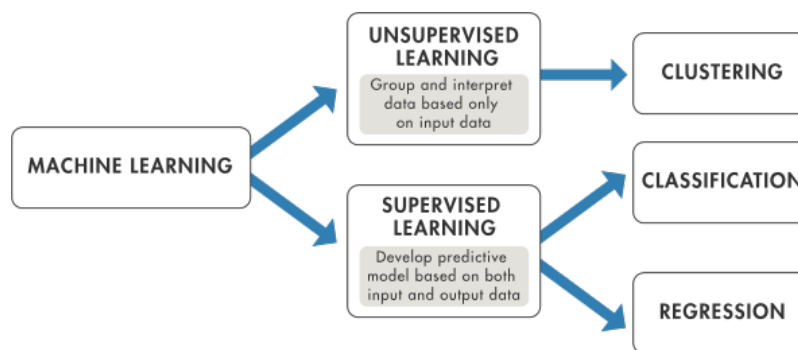
Při postupu učení rozlišujeme především dva typy dat. Data trénovací, na nichž probíhá trénování samotného modelu (známé vstupy i výstupy). Data testovací pak slouží k ověření úspěšnosti modelu a nejsou zahrnuta v množině trénovacích dat. Obecný postup strojového učení pak je takový:

- **Příprava vstupních dat:** data transformujeme do takové podoby s níž můžeme pracovat.
- **Trénování:** učicímu se systému předáme množinu dat na kterých se pak naučí požadovanou činnost nějakým ze způsobů zmíněných výše (učení s učitelem, bez učitele, případně jinou metodou).

- **Validace:** testujeme systém na datech, u nichž známe výsledek a sledujeme úspěšnost této predikce.
- **Aplikace, či přeučování:** pokud je validace úspěšná, můžeme systém začít využívat, pokud nikoliv, pak upravujeme trénování a vracíme se na bod druhé odražky tohoto testování nebo často také do prvního bodu po úpravě množiny vstupních dat.

4.2 Dělení algoritmů dle způsobu učení

Jak jsem již výše zmínil, hlavní způsob dělení je na učení s učitelem a učení bez učitele. Klíčové je zde pochopit, že při učení s učitelem je zde přítomna v trénovacích datech výstupní hodnota x , kterou požadujeme. Pokud ano, pak se jedná o učení s učitelem, jelikož učitelem je právě tato hodnota x . Pokud nikoliv, pak se jedná o učení bez učitele a typicky zde hledáme nějaká spojená data (souvislosti) a hledáme shluky takto podobných dat. Tato spojená data mohou pak představovat například hledané objekty. Členění algoritmů pak lze vidět na obrázku 4.1 níže.



Obrázek 4.1: Dělení algoritmů strojového učení dle způsobu učení [24]

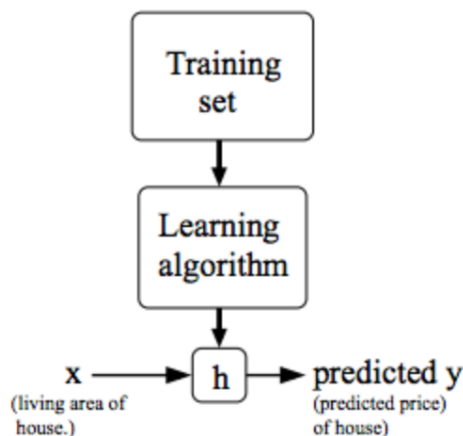
4.2.1 Učení s učitelem

Ještě před samotným učením je důležité analyzovat danou oblast a na základě těchto znalostí vybrat množinu atributů, které by měly být relevantní vzhledem k dané oblasti. Cílem celého tohoto procesu je pak nastavení modelu takového, aby co možná nepřesněji odpovídal skutečnému modelovanému procesu. Tímto modelem je zde myšlen natrénovaný algoritmus. Učení s učitelem má pak své specifické problémy, stanovení skutečné chyby modelu a přeučení modelu [37]. Algoritmus tedy má známý vstup a známé odpovědi na tento vstup. A snaží se generovat odpovídající model, který by provedl odpovídající predikce poté i na základě nových, dosud neznámých dat. Tento princip pak bude využíván zde v simulátoru, při predikci na starších výsledcích, kde máme k dispozici výsledek takového zápasu. Na obrázku 4.2 můžeme vidět graficky znázorněn obecný postup učení s učitelem.

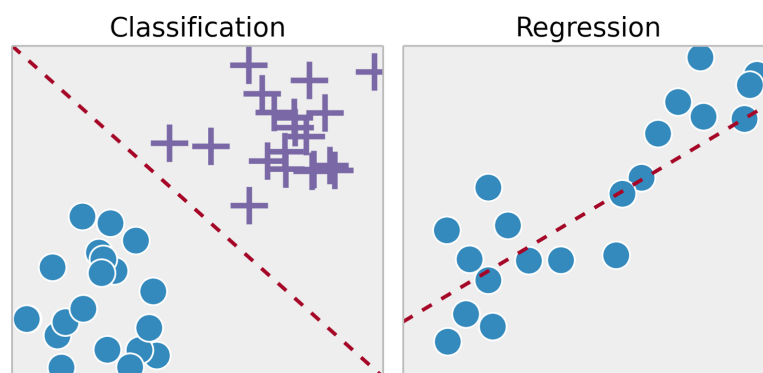
4.2.2 Techniky učení s učitelem

Existují dva způsoby technik jak přistupovat k učení s učitelem. Jedná se o **klasifikaci** a **regresi**. Na obrázku 4.3 a v textu níže můžeme vidět rozdílnost výsledku obou metod.

Klasifikační techniky generují diskrétní výsledek, typicky náležitost výsledku do podmnožiny. Jednoduchým příkladem pak může být rozpoznávání e-mailů v naší e-mailové



Obrázek 4.2: Model učení s učitelem [32]



Obrázek 4.3: Rozdílnosti povahy výsledků klasifikace a regrese

schránce a jeho rozdělení do složky přijatých (normální e-mail), hromadných (newsletter) a nevyžádaných (spam). Metodu klasifikace tedy typicky používáme když chceme objekty kategorizovat, tedy rozdělit do skupin, případně tříd objektů. Příkladem technik klasifikace pak mohou být rozhodovací stromy nebo neuronové sítě.

Naproti tomu regresní techniky generují spojitý výsledek, může se například jednat o vývoj určité veličiny v čase (například vývoj teploty). Regresi pak typicky používáme pokud by náš požadovaný výsledek mohl být reprezentován reálným číslem nebo pokud pracujeme s nějakým rozmezím dat. Jako příklad regresní techniky můžeme uvést například zobecněný lineární model nebo neuronové sítě.

4.2.3 Učení bez učitele

V této diplomové práci se zaměřuji na učení s učitelem, oba způsoby ale je podstatné zmínit pro pochopení problematiky. Učení bez učitele tedy nevyužívá žádnou předem známou veličinu, čili naším úkolem je popsat data. K tomu využíváme například techniku shlukování, případně určení závislostí [57]. V praxi to tedy znamená že dáváme učicímu se systému pouze vstup. Velmi triviálním příkladem z lidského světa může být například učení dítěte, kterému

dáváme různě velké kostky a čekáme zdali se naučí je poskládat ve správném pořadí dle jejich velikostí.

4.2.4 Problém přeučení

Častým problémem v oblasti strojového učení je tzv. *overfitting* neboli *problém přeučení*. Přetrénovaný model je takový model, který je příliš závislý na konkrétních datech. Typicky pak pro konkrétní trénovací data obsahuje minimální chybu. Pokud je ale použit na predikci výsledků vycházející z jiné množiny dat, pak dosahuje špatných výsledků. Při přetrénování daného algoritmu totiž model chápe odchylky nebo náhodné chyby ve výsledcích jako standardní prvky. Tím pádem s těmito odchylkami počítá i při nových predikcích. Nicméně, dané odchylky může způsobovat šum nebo jak jsem zmínil náhodný faktor, v jiné množině dat tyto odchylky již nebudou. Neznamená to, že by odchylky v nových datech nevyskytovaly, jen se budou projevovat jinak. Model má tedy problém s generalizací. Přeučení pak způsobuje i snížení výkonnosti daného modelu [31]. Příkladem možného přeučení v této diplomové práci by mohlo například být, pokud bych do trénovacích dat vložil pouze zápasy určité množiny ligových soutěží. Proto je důležité vložit do těchto dat ideálně zápasy z většiny evropských soutěží (jak papírově nejsilnější, tak papírově nejslabších). A také zápasy, kde se týmy z různých soutěží utkávají (typicky Evropské poháry).

Předcházení přeučení

Vytvoření modelu predikujícího správné výsledky je poměrně komplexní a složitá věc. Je tedy vhodné zmíněnému problému přeučení předcházet. Jako příklady základních technik prevence pak mohou sloužit tyto principy [50, 16]:

- **rozšíření množiny dat manuálně:** čím rozsáhlejší množinu dat používáme v modelu, tím menší je pravděpodobnost vzniku přeučení, prvním přístupem by tedy mělo být manuální rozšíření této množiny, nejlépe sémanticky odlišnými daty
- **použití rozšiřování dat:** přístup z minulého bodu se dá i automatizovat, v mém projektu by se tak například po každém odehraném kole v reálném světě mohly právě odehrané zápasy přidat do množiny dat a provést tak zpřesnění a rozšíření sady dat
- **snížení komplexity architektury:** snížení přesnosti a odstranění některých parametrů funkcí, příkladem může být například snížení stupně polynomu u polynomiální regrese
- **odstranění některých vstupů:** tento krok by již měl být pokryt ve zpracování dat, nicméně pokud v modelu evidentně existuje problém s přeučením, je vhodné znovu analyzovat vstupy

4.3 Zisk a předzpracování dat

Před samotným vykonáváním funkce je nutné připravit odpovídající a správně nastavenou množinu dat. Tento proces se nazývá předzpracování dat a tato aktivita se skládá z těchto podčástí [58]:

- **zisk samotných dat:** dolování z databází, webových stránek apod.

- **výběr relevantních atributů**
- **čištění dat:** například vyplňování nulových hodnot
- **doplnění výstupů:** v případě potřeby dotvoření výstupů na základě vstupů, které lze vypočítat při zpracování a doplnit tak množinu dat
- **získávání odvozených atributů**
- **převod typů dat**
- **transformace dat**
- **formátování pro jednotlivé modelovací techniky**

Tato analýza je důležitá pro získání potřebných výsledků. Prerekvizitou je důkladná analýza oblasti. Pokud bychom analyzované oblasti dobře nerozuměli, není příliš pravděpodobné, že by se podařilo vybrat relevantní atributy pro získání očekávaných výsledků. Zároveň je klíčový dobrý výběr zdroje dat. Hodnocení zdrojů je závislé na několika faktorech, ale z programátorského hlediska je pro nás důležitý výběr správného formátu dat. Zpravidla pak také tato celá část zabere nejvíce času a zdrojů. O výběru zdrojových dat pro praktickou část této práce pak píší v sekci [6.1](#).

4.4 Srovnání modelů

Vzhledem k povaze projektu zde rozeberu dva přístupy ke tvorbě predikčního modelu, mezi kterými jsem se před samotným návrhem dané predikce rozmýšlel. Těmito modely jsou: **regresní funkce** a **neuronová síť**.

4.4.1 Regresní model

Klíčovou oblastí této práce, na níž stojí predikce výsledků je regresní model. Na základě vstupních dat se stroj učí určit výstupní hodnotu, kterou je reálné číslo. Vstupními daty může být třeba množina dvojic. Cílem je tedy nalézt funkci, která pro danou vstupní hodnotu co nejlépe aproximuje výstupní hodnotu. Regresi pak dále dělíme na *linární* a *polynomiální*. Lineárních regresí pak můžeme používat více zároveň. Rozdíly ve vzorcích těchto typů regrese lze vidět na následující obrázku. Důvod výběru je pak zmíněn v sekci věnované návrhu provádění predikce.

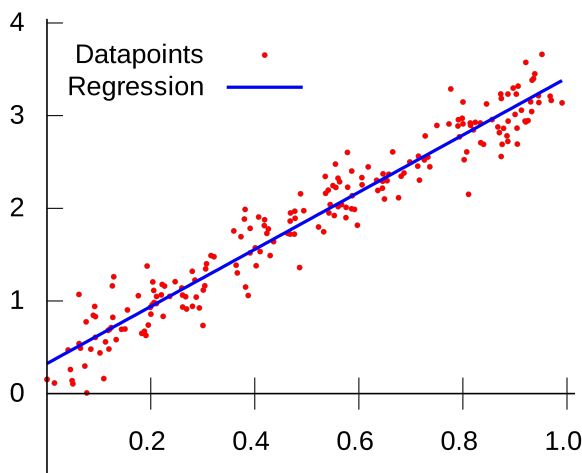
Simple Linear Regression	$y = b_0 + b_1x_1$
Multiple Linear Regression	$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$
Polynomial Linear Regression	$y = b_0 + b_1x_1 + b_2x_1^2 + \dots + b_nx_1^n$

Obrázek 4.4: Vzorce odpovídající jednotlivým typům regrese [\[18\]](#)

Klíčovým prvkem všech regresních modelů je *korelační koeficient*. Tento korelační koeficient pak udává rozsah vnesených bodů, seskupených okolo rovnice daného modelu. Rozsah hodnot tohoto korelačního koeficientu je v intervalu $<-1,1>$. Tato korelační analýza ale může být značně matoucí, proto by měla být vždy doplněna vizualizací bodů a funkce [35].

Lineární regrese

Lineární regrese je typ regrese, který využívá proložení množiny bodů přímkou (4.5) [40]. Závislost těchto veličin je tedy lineární. Předpokladem je, že obě veličiny (vstupní i výstupní) jsou spojitě. U regrese obecně pak zpravidla nastává problém odlehlých bodů. I malá skupina odlehlých bodů způsobená nějakou odchylkou je schopná změnit tvar přímky a tím i výpočtu. Pokud máme dostatečný objem dat, pak můžeme odlehlý bod vyloučit. Před vyloučením bychom se ale vždy měli přesvědčit o příčině vzniku tohoto bodu. Případně ověřit, jestli vznik tohoto bodu způsobila například hrubá chyba. Díky odlehlým bodům, které výrazně mění tvar přímky pak není vhodné používat tuto metodu vždy. Někdy samotná přímka nestačí k zjištění vzorů v datech (v případě existence nelineárních závislostí) a je nutné využít polynomiální regresi.



Obrázek 4.5: Graf funkce lineární regrese

Polynomiální regrese

Polynomiální, neboli polynomická regrese je model, který je lineární vzhledem k parametřům, ale popisovaná závislost mezi proměnnými je nelineární. Zakřivení přímky se pak zvyšuje se zvyšujícím se stupněm polynomu. Polynomiální regrese je ale z matematického hlediska stále lineární problém [44]. Polynomiální regrese je chápána jako speciální případ více lineárních regresí [43].

Pro dosažení proložení bodů nelineární křivkou je ovšem nutné transformovat vstupní atributy na vyšší stupeň. Této transformaci lze využít více způsoby, jako příklad uvedme například funkci `PolynomialFeatures` z používané knihovny `scikit-learn`. K vytvoření rovnice vyššího řádu pak můžeme přidat vlastnosti původních atributů jako nové atributy. Příklad této transformace lze vidět na obrázcích 4.6 a 4.7 (lineární funkce a funkce s polynomem 2.řádu). Zajímavé je, že se stále jedná o model lineární, křivka ale již ze své podstaty

linární není [27]. To tedy umožňuje využívat výkonnosti a rychlosti lineárních modelů, ale použít optimálnější a přesnější funkci pro přesnější proložení bodů.

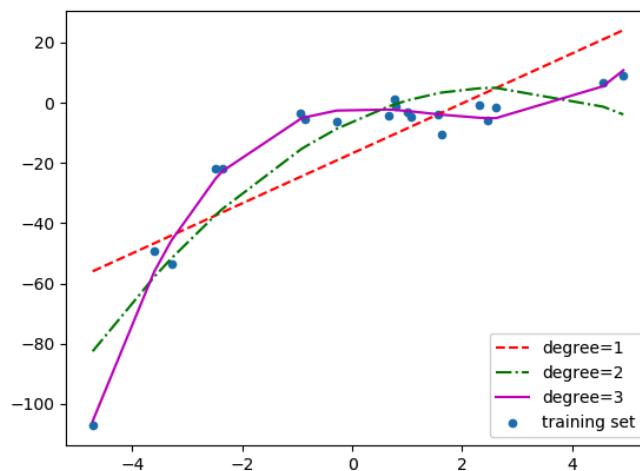
$$Y = \theta_0 + \theta_1 x$$

Obrázek 4.6: Lineární model

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2$$

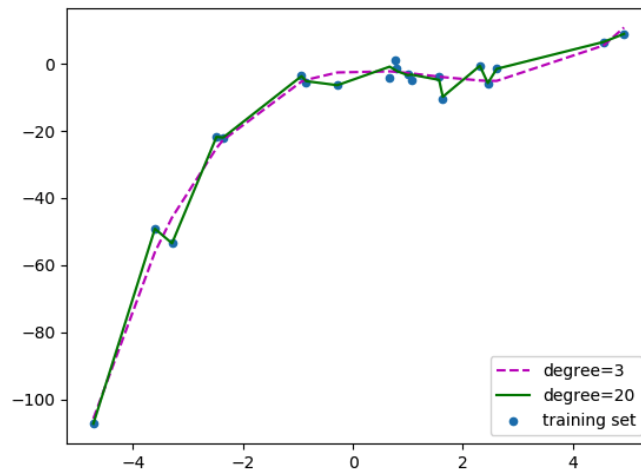
Obrázek 4.7: Model s transformovanými atributy

Na obrázku 4.8 vidíme, jak zakřivení pomohlo lépe proložit stejné vstupní body. Polynomiální regrese se pak využívá v případech, kdy trénovací data sice jsou korelovaná, ale zjevně se v nich vyskytují nelineární závislosti [48]. Při trénování pak je důležité citlivě určit stupeň této křivky a připravit rozsáhlou množinu testovacích dat na níž bude testována schopnost regresního modelu predikovat odpovídající výsledky.



Obrázek 4.8: Výsledky proložení bodů křivkou s různým stupněm polynomu

Problémem pak může být až příliš vysoký stupeň polynomu, který pak může způsobit přetrénování neboli již zmíněný *overfitting* a přílišné zakřivení přímky pak nemusí sedět na nová data, jejichž výsledky chceme získat. Tento problém můžeme vidět na obrázku 4.9, kde vidíme, že křivka se stupněm 20 prochází všemi body, tedy typicky generalizuje i odchylky. Použití této křivky je tedy dáno množinou trénovacích dat a na datech nových, s jinými odchylkami by pravděpodobně v mnoha případech docházelo ke generování nesouvisejících hodnot [17].



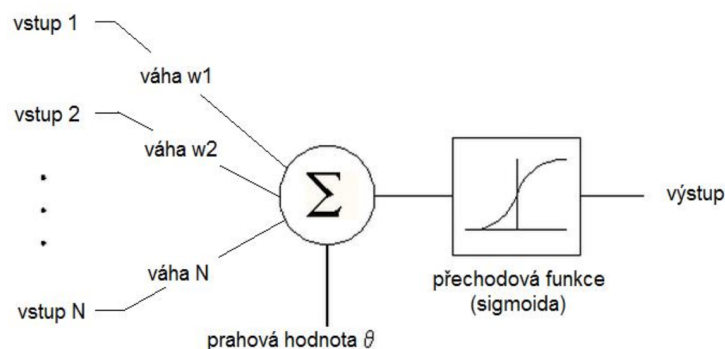
Obrázek 4.9: Znáznornění přetrénované funkce

4.4.2 Neuronová síť

Umělá neuronová síť je algoritmus, který se inspiruje biologií, konkrétně činností lidského mozku. Mozek je tvořen množstvím propletených buněk, kterým říkáme neurony. Tyto buňky spolu komunikují pomocí elektrických impulzů. Jak již vyplývá z textu výše, lidé technického zaměření se snaží napodobovat činnost lidského mozku pro využití v oblasti umělé inteligence. Neuronové sítě tedy úzce souvisí se strojovým učení [46].

Neuron

Základní stavební jednotkou neuronové sítě je *neuron*. Z biologického hlediska se jedná o jednu buňku. Pro neuron se v oblasti umělé inteligence používá i označení *perceptron*. Do neuronu přichází n vstupů, které mohou být buď signály z vnějších okolí nebo výstupy jiných neuronů ze stejné sítě. Každý vstup je pak ještě rozšířen o váhu w . Pokud je překročena prahová hodnota, pak je neuron nabuzen a vysílá na svůj výstup signál, neboli přechodovou funkci. Na obrázku 4.10 pak můžeme vidět samotný model neuronu.



Obrázek 4.10: Model neuronu (perceptronu)

Váhy vstupů však mohou být i záporné, což značí jejich inhibiční charakter. Suma, která je vážená v kombinaci s váhou w reprezentuje vnitřní potenciál neuronu. Matematická funkce neuronu (na obrázku 4.11) je založena na ostré nelinearitě a daná funkce má nulový práh. Tím pádem práh neuronů (se záporným znaménkem) můžeme chápat jako váhu dalšího formálního vstupu s konstantní jednotkovou hodnotou [56].

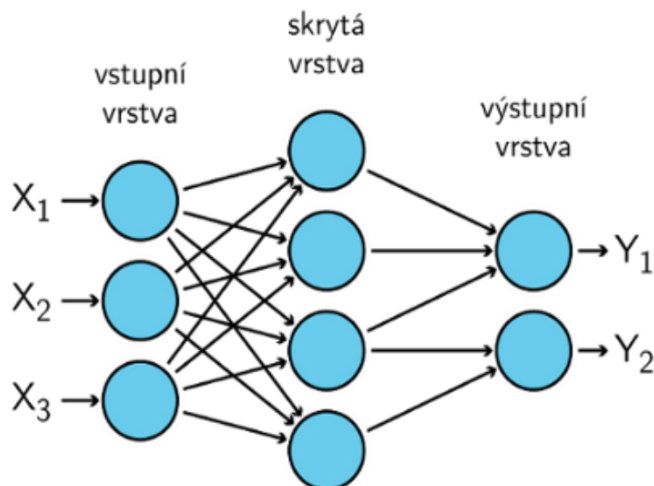
$$y = \sigma(\xi) = \begin{cases} 1 & \text{jestliže } \xi \geq 0 \\ 0 & \text{jestliže } \xi < 0 \end{cases}, \text{ kde } \xi = \sum_{i=0}^n w_i x_i.$$

Obrázek 4.11: Matematická funkce neuronu

Pomocí neuronu pak lze dělat klasifikaci. Na základě vstupních dat může neuron zkoumaný objekt klasifikovat a přidělit ho do určité třídy. Zajímavé je, že klasifikovat lze i pouze jedním neuronem, prakticky se ale využívají komplexnější neuronové sítě, kupříkladu vícevrstvá neuronová síť.

4.4.3 Vícevrstvá neuronová síť

Vícevrstvá neuronová síť přidává další (skrytou vrstvu) k vrstvám vstupním a výstupním. Těchto vnitřních vrstev může být i více. Tento typ sítě je obecně nejrozšířenější a nejpoužívanější. Jeho znázornění lze vidět na obrázku 4.12. Síť se skládá ze skupiny neuronů, výstupní neurony jsou dány pak tím jak chceme kódovat výstup. Tato síť je standardně učena učním s učitelem, velmi často pomocí algoritmu jménem *Back Propagation Error*, učení pak trvá poměrně dlouho. Skrytá vrstva se pak využívá tak, že pokud při ladění a trénování nemá síť dostatečnou schopnost klasifikovat, může přistoupit ke vložení dalšího neuronu do skryté vrstvy. Případně může učící soubor rozšiřovat o další stavy. Chybovost by se pak učním měla snižovat a asymptoticky se blížit k 0 [47].



Obrázek 4.12: Obrázek vícevrstvé neuronové sítě

4.4.4 Samotné srovnání modelů

V této podsekcí se budu věnovat především srovnání regrese polynomiální a neuronových sítí. Základem analýzy budou požadované prekvizity, výhody a nevýhody daných modelů. Obecně ovšem nelze říct, že je jeden model lepší než druhý. Oba dva modely mají svá specifika a každá je efektivnější pro jiný účel nebo rozsáhlost množiny dat. Tuto analýzu pak doplňuje sekce 5.10 v části Návrhu.

Komplexnost

Neuronové sítě jsou komplexnějším algoritmem, mohou obsahovat vysoké množství vrstev. Při konkrétní implementaci (pokud uživatel netvoří úplně nový postup) se zpravidla s nejhlubšími vrstvami pracuje jako s černou skříňkou. Složitost těchto algoritmů pak ale může být i značnou nevýhodou, protože pokud neuronová síť opravdu funguje jako černá skříňka, často tedy nezískáme důvod k tomu jak konkrétně k rozhodnutí došlo. Jediný kdo teoreticky má přístup k vnitřní reprezentaci (u knihoven problematické) by pak byl samotný návrhář sítě nebo případně programátor, který danou síť implementoval. Z tohoto důvodu pak například banky příliš neuronové sítě nepoužívají, protože nejsou poté schopny sdělit bližší informace jako například důvod nepřidělení půjčky [29].

Jak jsem již výše zmínil, algoritmus polynomiální regrese není tak komplexní jako neuronové sítě. Při implementaci tohoto algoritmu je především důležité určitě správný tvar a stupeň daného polynomu. Tyto parametry lze ovšem exaktně a rychle upravovat, úprava jádra modelu je tedy pak snazší než u druhé varianty. S rostoucí množinou dat a vstupů ale určení vhodnosti může být problémem. V případě složitých problémů s velkým množstvím vstupů je tedy výhodnější použít neuronovou síť.

Množina dat a vstupů

Neuronové sítě obecně pracují lépe s velmi rozsáhlou množinou dat. Čím robustnější množina dat je, tím se výpočet neuronové sítě zlepšuje. Zároveň s tím ovšem roste i časová náročnost trénování. Neuronové sítě pak nejsou vhodné pro tvorbu predikce, pokud nemáme data v rozsahu statisíců, či miliónů záznamů. Zároveň díky výrazným postupům a zlepšením v rámci samotných algoritmů neuronových sítí a neustálé optimalizaci těchto algoritmů je možné používat stále větší a větší množiny dat.

Naproti tomu, polynomiální regrese je uzpůsobena i při trénování na méně rozsáhlé množině dat s méně vstupy. Neznamena to, že by nutně použití této metody na datech rozsáhlejších s objemnou množinou vstupů bylo špatné, vhodnější je ale v tom případě již využít neuronových sítí. Přidávání dat totiž nemusí vždy způsobit zpřesnění výsledku, vzhledem k nevyužití vah. Použití regresní model je tedy výhodné pokud nemáme nebo nepotřebujeme velmi rozsáhlou množinu dat [33].

Náročnost na čas a zdroje

Obecně lze říct, že výpočet trénování neuronové sítě trvá napříč různými knihovnamí, zabývajícími se strojovým učením, mnohonásobně déle než výpočet v rámci klasických algoritmů [29]. Naproti tomu pak na již natrénovaném modelu samotné predikce jsou neuronové sítě zpravidla jedním z nejrychlejších algoritmů. Neuronové sítě jsou tedy klíčové v oblastech, kdy potřebujeme provést rozhodnutí (predikci) v co nejkratším možném čase.

Co se týče polynomiální regrese, tak proces trénování trvá kratší čas a využívá méně zdrojů než neuronová síť. To ale částečně vychází z užší množiny dat a vstupů a také nižší komplexity celého algoritmu, který je postaven na lineárních modelech.

4.5 Využití strojového učení ve světě

Lze říci, že se strojovým učením se setkáváme v každodenním životě. V některých oblastech by to zřejmě dokázal odhalit i laik, například při využití v chytrých domácnostech, kdy například Google Assistant dokáže rozpoznat náš hlas a reagovat na něj [11]. Díky tomu dokáže i software například v chytrých telefonech zjišťovat naše pravidelné návyky a na ně pak reagovat. Právě v těchto chytrých domácnostech pak dokáže umělá inteligence například na základě toho kdy majitel domácnosti chodí domů z práce zapnout před příchodem klimatizaci a nastavit ji na oblíbenou hodnotu, nebo zapnout ráno kávovar v požadovaný čas. Jako typický příklad strojového učení pak lze dále uvést vyhledávání na Googlu [45].

Zajímavou oblastí je například také marketing a reklama. Naše počítače i telefony nebo programy tvořící reklamní maily (newslettery) jsou schopni nám vytvořit nabídku, která je uzpůsobena často nakupovanými nebo prohlíženými produkty. Prodejce pak dokáže lépe zacílit na potřeby jednotlivých zákazníků, připravit jim nabídku na míru a zvýšit tak pravděpodobnost zvýšení zisků [53]. Z toho vyplývá že všechny největší společnosti vynakládají nemalé prostředky pro tvorbu algoritmů strojového učení.

Významnou oblastí využití těchto technik je robotika. Strojové učení se používá typicky na učení robota rozpoznávat objekty, případně zvyšovat efektivitu práce. Významnou možností vývoje je pak postupné nahrazování lidské síly v továrnách.

4.6 Využití strojového učení

Jak bylo již zmíněno, strojové učení bude v praktické části práce využito pro odhadování výsledků zápasů. K tomu bude potřeba zajistit relevantní množinu historických dat. Tím pádem zde hovoříme o učení s učitelem – známe vstup a také výstup (tedy konkrétní historický výsledek).

Vzhledem k tomu, že ve fotbalovém zápase je početná množina faktorů, které výsledek ovlivňují, bude nutné vybrat takové faktory, které mají největší vliv na výsledek. Poté, až bude model dostatečně natrénován, pak bude propojena se simulátorem, který generuje náhodné faktory, které ovlivňují výsledek také. Nebude to tedy implementováno tak, že by model pokaždé vygeneroval shodný výsledek, tento výsledek totiž bude ovlivňován zvenčí a bude ovlivňovat výsledek vygenerovaný predikčním modelem. Pro tvorbu modelu bude využita knihovna v jazyce Python. Výstupem celého tohoto systému pak bude skóre obou týmů, z něhož pak již simulátor provede všechny operace jako je přičtení bodů, ovlivnění forem týmů atd.

Klíčovou oblastí práce tedy bude využití strojového učení na predikci výsledků fotbalových zápasů. Bude potřeba analýza faktorů ovlivňujících zápasy a tyto faktory se pak stanou vstupními atributy.

Kapitola 5

Návrh

V této kapitole bude popsán návrh webové aplikace, klíčovou oblastí je pak návrh modelu pro predikci výsledků a generování náhody. Popíši zde strukturu celého systému, poté jeho jednotlivé komponenty. Zároveň zde bude také popsán návrh uživatelského rozhraní, databáze a celé struktury aplikace. Dále zde naznačím průběh simulace a simulační krok.

5.1 Název a logo

Ještě před samotným souhrnem návrhu je zde důležité uvést, že samotný systém má již svoje logo a název. Vzhledem k tomu, že výstupem práce je první ucelená verze systému, bylo i vzhledem k návrhu vzhledu a zacílení aplikace vytvořit jméno i logo. Aplikace se tedy jmenuje *Footie*, což je v britské angličtině hovorová zkratka pro fotbal. Logo bylo navrženo grafickým designérem a cílem bylo především vytvořit jednoduchý a jasný obrázek, obsahující hlavní barvy aplikace. Logo lze vidět na obrázku 5.1.



Obrázek 5.1: Logo Footie

5.2 Cílový uživatel

Ve své práci se zaměřuji na uživatele, kteří jsou znalí fungování fotbalového světa. Mohou to být buď fanoušci nebo hráči fotbalových manažerů. Řada těchto lidí hledá odlehčenou verzi hry, kde by mohli pouze sledovat dění ve fotbalovém světě. Jejich jediná možnost je spustit si fotbalový manažer a aktivně se jí neúčastnit, což vždy není možné. Zároveň by ovšem chtěli,

aby simulace nebyla omezená rokem ukončení a obsahovala co možná nejvíce soutěží pro sledování. Z dalších požadavků vyplynulo, že by simulátor požadoval výkonný hardware a bylo by možné jej mít spuštěný pohodlně i na pozadí. Simulátor tedy má poskytovat hlavně tuto požadovanou část a bude se postupem času rozšiřovat.

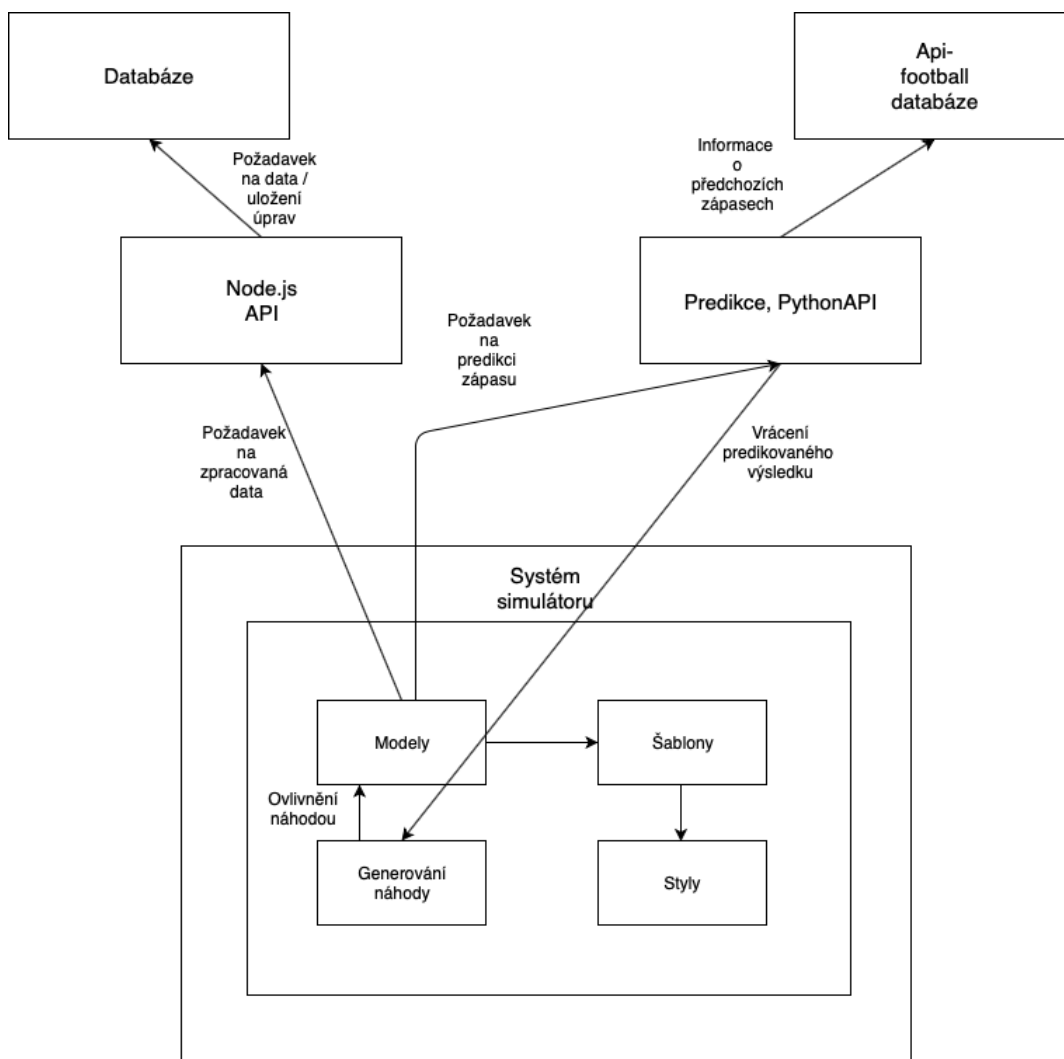
5.3 Souhrn funkcionality

Cílem je naprogramovat systém pro simulaci fotbalového světa. V tomto systému si pak uživatel bude moci vybrat ligy, které chce sledovat a bude moci neomezeně procházet jednotlivé ročníky a generovat výsledky po jednotlivých kolech soutěží, případně po jednotlivých zápasech. Systém v této práci umožňuje simulovat pět evropských lig (Anglie, Španělsko, Francie, Itálie, Německo) a generovat postupy a sestupy mezi nejvyšší a druhou nejvyšší soutěží. Samotný systém se bude vyvíjet a po každé sezoně se budou přepočítávat kvality týmů a hráčů. Systém bude schopen predikovat výsledky dle historických výsledků, dále ovšem bude přidávat určitou náhodu dle různých faktorů. Bude tedy simulovat alternativní budoucnost pro fotbalový svět. Každý zaregistrovaný uživatel pak bude mít svou simulaci, která bude startovat od počátečního data současné sezóny. Uživatel bude moci si simulaci smazat a začít znovu od zmíněného data.

5.4 Návrh systému

Samotný systém je rozdělen na tři hlavní části: logiku systému, databázi a model predikce. Tyto části spolu pak komunikují. Diagram ilustrující strukturu lze vidět na obrázku 5.2. Jádro systému bude postaveno na MVC architektuře, konkrétně se bude jednat o webovou aplikaci. V samotném jádře pak je i generování náhody.

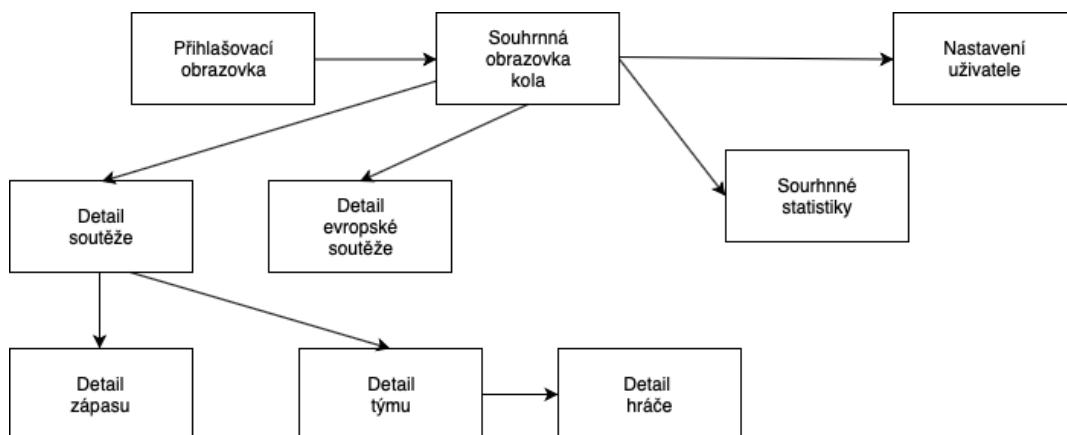
Jádro pak komunikuje s MySQL databází na vzdáleném serveru a podle své potřeby stahuje, či modifikuje data. Další částí je pak regresní model pro predikci výsledků, ta je uložena odděleně a systém pak předává své vstupy, které získal z databáze. Výstup z modelu pak neslouží jako konečný výsledek, ale je ještě zpracováván jádrem společně s výstupem náhody. Výstupem z modelu je počet vstřelených branek obou týmů, výstupem generování náhody je pak ovlivnění tohoto výsledku.



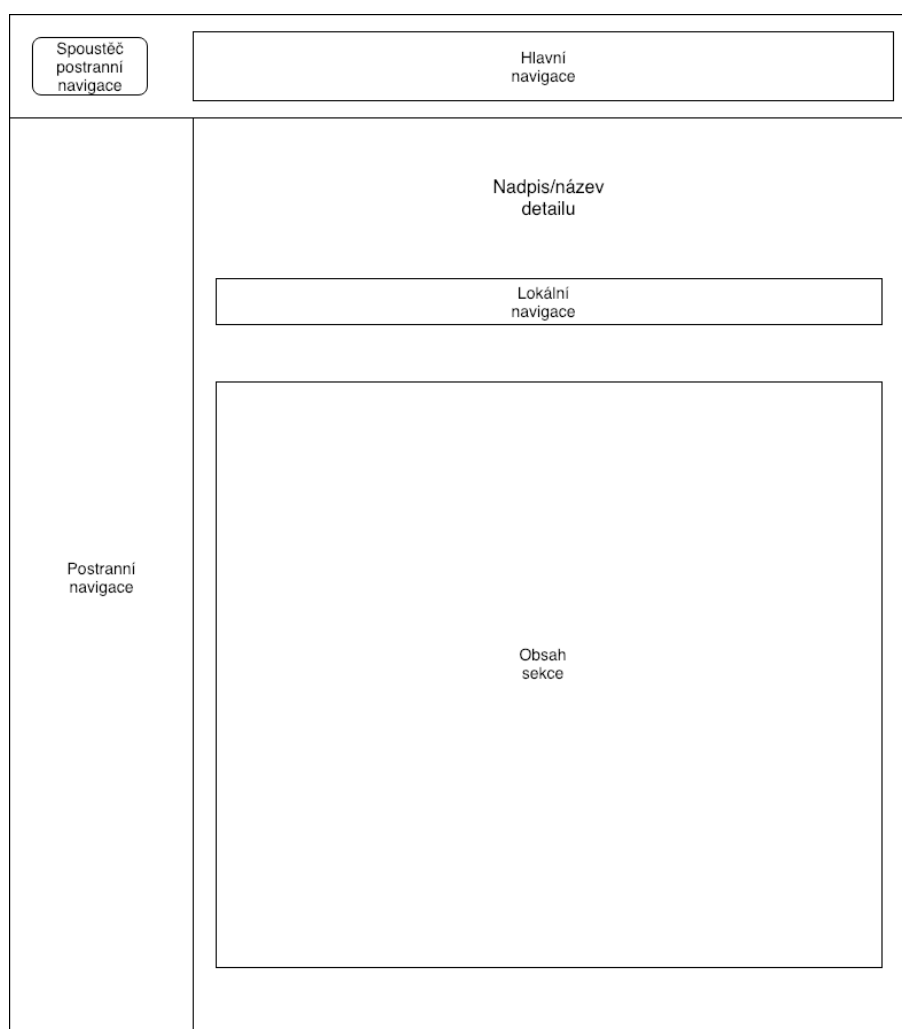
Obrázek 5.2: Struktura všech komponent

5.5 Návrh uživatelského rozhraní

Uživatelské rozhraní je navrženo s důrazem na jednoduchost a přiměřené předkládání informací uživateli. V určitých částech se inspiroji designem aplikace Football Manager. Hlavní navigace je přístupná z horní lišty, další navigace je pak při inicializaci stránky zabalena, obsahuje ale především navigování mezi jednotlivými soutěžemi. Hlavní část všech obrazovek pak spravuje obsah. Na každé obsáhlejší obrazovce je pak obsah přístupný skrze lokální navigaci, která zobrazuje pouze určitou sekci. Například u obrazovky detailu soutěže lze přepínat mezi rozpisem zápasů, tabulkou, výsledky a statistikami. Na diagramech níže jde pak vidět posloupnost obrazovek (obrázek 5.3) a návrh globální obrazovky (obrázek 5.4).



Obrázek 5.3: Posloupnost obrazovek



Obrázek 5.4: Obecná obrazovka

U všech obrazovek bylo klíčové navrhnout lokální navigaci. Každá obrazovka totiž obsahuje vysoké množství dat a je podstatné tato data předkládat uživateli přehledně a struktu-

rovaně. Přepínání mezi sekcemi bude probíhat asynchronně, bez znovunačítání celé stránky. Většina obrazovek vychází ze struktury obecné obrazovky a u těch, které se liší přidávám i znázornění pomocí barevného obrázku. Chtěl jsem především vycházet z obecného návrhu aby rozhraní bylo konzistentní a pro uživatele bylo snadné si zapamatovat umístění jednotlivých akcí dle specifikovaných postupů návrhu rozhraní. Dále zde byla využita konzistentní barevná paleta, asociovaná s jednotlivými akcemi. Pokud se daný element vyskytuje na více místech pak vypadá vždy stejně [38]. Jako příklad uveďme tabulku soutěže. Ta se vyskytuje na dvou obrazovkách a její struktura i barevné odlišení je stejné. Tabulka na detailu soutěže je komplexnější a obsahuje více dat.

Jak jsem již výše zmínil, důležitá je přístupnost z širokého rozmezí zařízení. Díky využití principů webových aplikací a systémů pak není pro využívání potřeba výkonných strojů a systém bude fungovat na všech zařízeních s podporovanými webovými prohlížeči [42].

5.5.1 Hlavní obrazovka

Po přihlášení se uživatel dostane na hlavní obrazovku kde vidí tři klíčové části simulace: **rozpis zápasů aktuálního týdne**, **náhled tabulek jednotlivých soutěží** a **výsledky z posledního týdne**. Měl by mít možnost simulovat si zápasy buď po jednom zápase nebo po soutěžích. Mezi těmito třemi sekcemi bude moci plynule přepínat. V rámci testování zde budou pravděpodobně i tlačítka pro delší intervaly simulace, což by ale ve finální sekci nebylo vhodné ponechat, vzhledem k pravděpodobným dlouhým načítacím časům. V principu tedy uživatel bude moci pro rychlou simulaci používat hlavně tuto obrazovku, na všechna detailnější data pak obrazovky další.

5.5.2 Obrazovka soutěže

V sekci o soutěži bude moci uživatel získat detailnější informace o konkrétní soutěži. Těmito informacemi jsou: **rozpis zápasů**, **výsledky zápasů**, **komplexní tabulka** a **statistiky o hráčích**. Tato obrazovka bude také sloužit jako jakýsi rozcestník mezi dalšími sekcemi zápasů, týmů a hráčů. Uživatel zde tedy bude moci dohledat rozpis a výsledky zápasů za celou aktuální sezónu. Na obrázku 5.5 můžeme vidět podobrazovku s tabulkou a rozpisem zápasů.



Obrázek 5.5: Barevný návrh obrazovky soutěže (tabulka a zápasy)

5.5.3 Obrazovka týmu

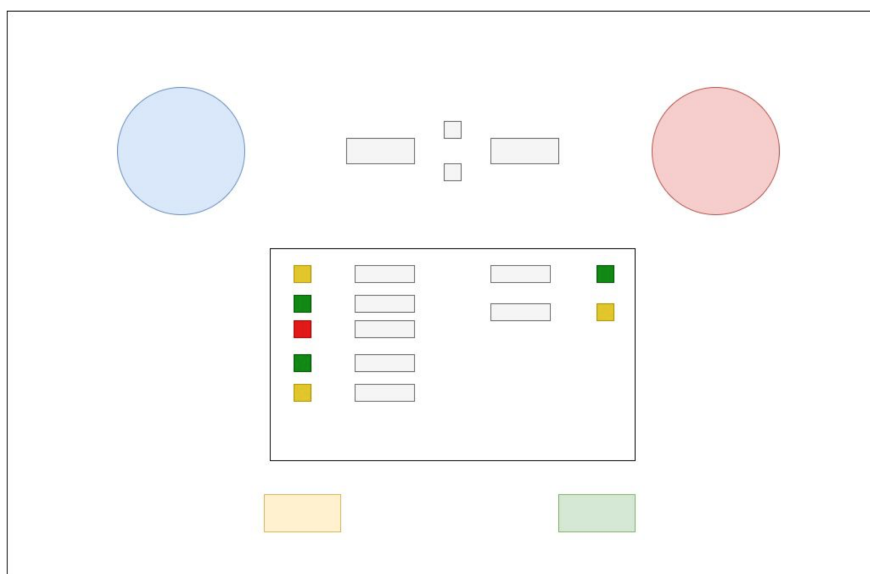
Na obrazovku konkrétního týmu se uživatel bude pravděpodobně chtít dostat pro zobrazení stavu konkrétního týmu, možnost dostat se do této sekce bude mít z jakékoliv sekce s výsledky, případně z tabulek nebo z detailu zápasu a hráče. Součástí této obrazovky jsou: **základní informace o týmu, hráči týmu, výsledky týmu, naplánované další zápasy týmu, nejlepší střelci týmu a historie týmu.**

5.5.4 Obrazovka hráče

Vzhledem k tomu, že systém je orientován především na týmy a ligy, tak o hráči jsou zobrazeny informace pouze v jedné sekci. Jde především o základní informace a statistiky daného hráče. Pozice hráče pak bude přehledně zobrazena v obrázku hřiště, takže uživatel bude schopen jednoduše analyzovat o jaký typ hráče jde (útočník, záložník, obránce nebo brankář).

5.5.5 Obrazovka zápasu

Návrh obrazovky je značně inspirován detailem zápasu ze hry Czech Soccer Manager. Především obsahuje skóre zápasu, ale i detailní informace o událostech v rámci zápasu, především tedy o střelcích a minutách vstřelení branek. Mimo rámec návrhu pak obrazovka pravděpodobně bude stejně jako ve zmíněné hře osahovat náhodnou fotografii z dané soutěže pro doplnění atmosféry. Návrh této obrazovky lze vidět na obrázku 5.6.



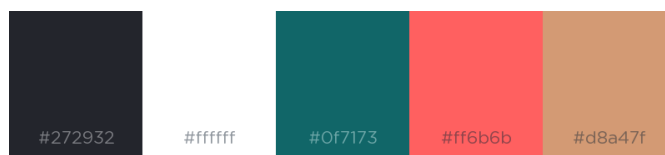
Obrázek 5.6: Barevný návrh obrazovky soutěže

5.5.6 Obrazovka uživatele

Uživatelská sekce je opět jednoduchá, uživatel si zde může vybrat ligy, které chce sledovat. Dále na této obrazovce uvidí své informace a bude si moct vynulovat svoji simulaci do původního stavu. Do budoucna tato obrazovka bude výrazně rozšířena, jak přibude více možností úprav skrze uživatele samotného.

5.5.7 Barevná paleta

Pro návrh rozhraní je nutno ještě přidat i samotné barvy vyskytující se v systému. Navržené hlavní barvy částečně vycházejí z loga. Snažil jsem se navrhnout takové barvy, aby byly dostatečně kontrastní, ale zase nebyly přespříliš výrazné, pokud to není potřeba. Jedinou výraznější barvou je pak navržená červená. Návrh barev především doplňuje funkcionalitu tlačítek v celém systému (potvrzení, zrušení a editace). Barevnou paletu navrženou pro systém pak lze vidět na obrázku 5.7.



Obrázek 5.7: Barevná paleta systému

5.6 Databáze

Pro databázi bude využita MySQL databáze na vzdáleném serveru.

Pro databázi bude využita MySQL databáze na vzdáleném serveru. Databáze je rozdělena na dvě části. Jedna část je statická a neměnná, ta obsahuje informace, které se v čase simulace nebudou měnit. Za tento typ lze považovat například názvy týmů, hráčů, stadionů. Co se bude měnit v čase (a navíc ještě každému uživateli jinak) budou především hodnocení a síla hráčů a týmů. Případně pak ještě počet postupujících dle aktuálních koeficientů a mnoho dalších atributů. Jak jsem již zmínil, databáze se budou vyvíjet vlivem simulátoru každému uživateli jinak. Tím pádem každý uživatel bude mít svou vrstvu databáze pro svou uloženou simulaci. Důležité tedy je správně navrhnout databázi pro vyšší množství požadavků a rychlé vyhledávání v obsáhlých tabulkách.

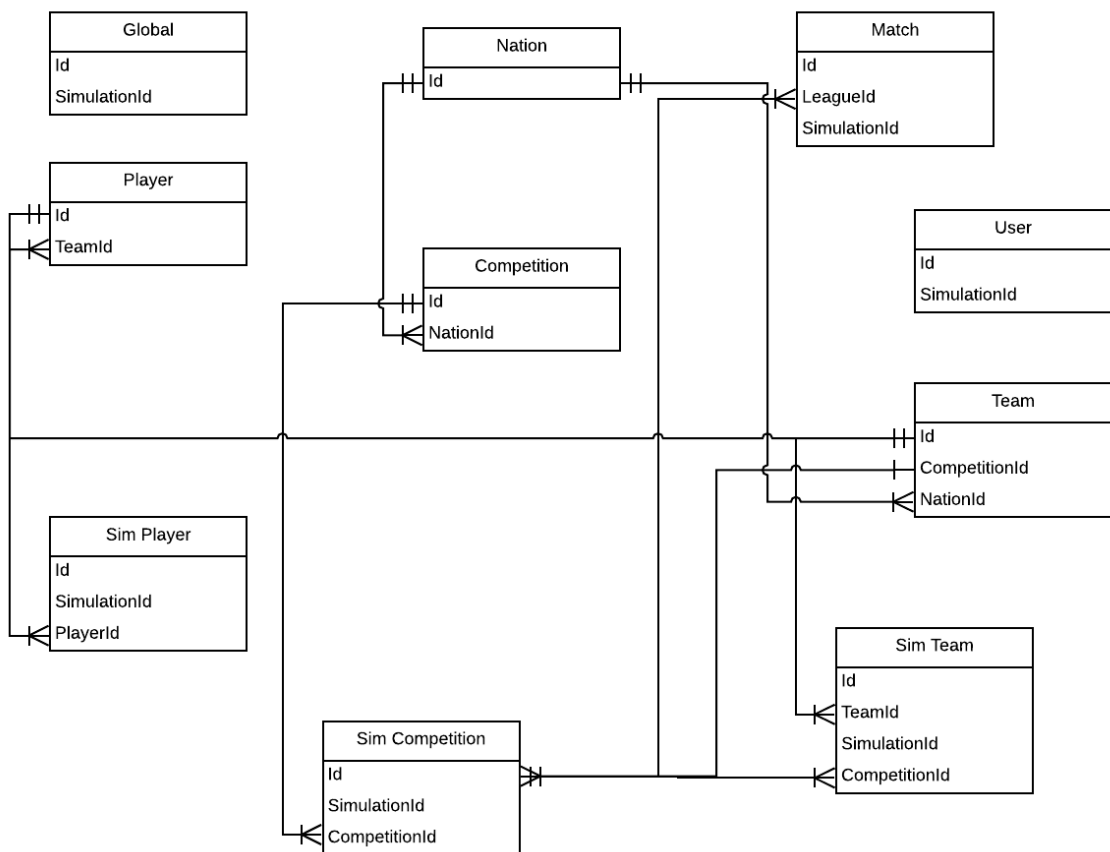
Návrh struktury databázových tabulek lze vidět na obrázku 5.8. Předpona sim před názvem tabulky značí provázanou tabulku s daty pro konkrétní simulaci. Důležitý jde zde také identifikátor simulace. Ten slouží k propojení dat s konkrétní simulací.

5.7 Komunikace s databází

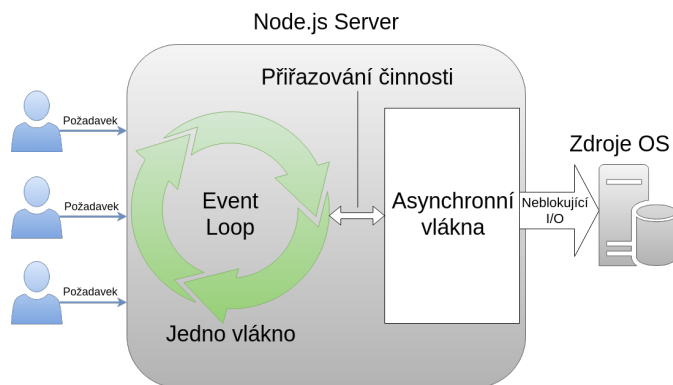
Na komunikaci s databází byla vybrána knihovna *Node.js*. Toto prostředí je postavené na jazyce JavaScript a primárním účelem této knihovny je tvorba serverové části webových aplikací. Klíčová část této technologie je tzv. *event loop*. Do tohoto cyklu pak vstupují všechny požadavky jako události, které jsou uvnitř serveru přiděleny jednotlivým, navzájem nezávislým vláknům [26]. Toto fungování lze vidět na obrázku 5.9.

5.7.1 Návrh REST API architektury

REST (Representational State Transfer) je architektura rozhraní, která je navržena pro distribuované rozhraní a je orientována datově. Velkou výhodou této architektury je jednotný přístup ke zdrojům. Všechny tyto zdroje mají vlastní unikátní identifikátor a existují čtyři typy přístupů k těmto zdrojům: **GET**, **PUT**, **POST** a **DELETE** [41]. Díky jednotnému přístupu ke zdrojům lze pak tuto architekturu využít ke tvorbě širokého spektra různých



Obrázek 5.8: Schéma databáze



Obrázek 5.9: Node.js server

typů serverových částí aplikací. Pomocí jedné architektury lze například implementovat stejné jednotné rozhraní jak pro webovou aplikaci, tak pro aplikaci mobilní.

Tato serverová část je navržena tak, aby nativně komunikovala s MySQL datábazí. Při návrhu systému bylo nutné nejdřív specifikovat modely, které vystupují v této architektuře.

Tyto modely souvisejí s jednotlivými částmi aplikace. Samotná skladba architektury se skládá ze tří klíčových součástí:

- **Cesty:** tato část definuje URL a další parametry pro použití jednotlivých zdrojů (implementace je různá dle metod GET, POST, UPDATE, DELETE).
- **Kontrolery:** zde probíhá zpracování vstupních dat, pokud je potřeba.
- **Modely:** na tuto část je přímo napojena databáze a modely obsahují jednotlivé SQL dotazy, pro přehlednost je vhodné aby modely komunikovaly každý se svou hlavní tabulkou a do dalších tabulek se dotazovali jen pro propojení.

V rámci návrhu jsem také specifikoval druhy požadavků na serverovou část. Tyto požadavky vycházely jak z návrhu uživatelského rozhraní, tak z požadavků na jednotlivé funkcionality. Pokud byly dva podobné požadavky, ale s menšími odchylkami, tak i tak byl navržen jiný zdroj, tak aby využití těchto zdrojů bylo v co největší míře znovupoužitelné i do budoucna vzhledem k dalším platformám. Klíčovým prvkem ve všech zdrojích je pak identifikátor konkrétní simulace. Inicializace dat ovšem neprobíhá skrze toto REST API, plnění databáze mají za úlohu inicializační skripty. Naimplementovaná konkrétní architektura je pak podrobněji popsána v příslušné implementační sekci.

5.8 Simulace celého systému a simulační krok

Systém bude začínat v počátečním datu aktuální sezóny. Uživatel bude moci simulovat jednotlivé zápasy či soutěže v rámci herního týdne. Simulační krok je tedy jeden týden sezóny. Tato kola je nutné simulovat ve správném pořadí (nebude tudíž možné odsimulovat jedno kolo španělské ligy a zároveň tři kola ligy anglické). Důvodem je zachování aktuálních odehraných zápasů pro potřeby evropských pohárů.

Po odehrání všech soutěží bude následovat přepočítání ročníku. V této operaci proběhne vynulování hodnot pro aktuální sezónu, přepočítání kvality týmu a hráčů podle výsledků v tomto ročníku a výpočet měnících se hodnot (například přidání 1 roku ke věku všech hráčů). Tuto operaci je možné provést až při ukončení všech soutěží v daném ročníku. Historická data jsou v redukované podobě ukládána pro uložení historie soutěží, týmů a zápasů.

5.9 SciKit-learn knihovna

Pro implementaci modelu učení s učitelem budu v simulátoru využívat knihovnu *SciKit-learn*¹, postavenou na programovacím jazyce Python (některé části této knihovny používají Cython pro zlepšení výkonnosti a rychlosti). Knihovna je zdarma k využití a vznikla v roce 2010 jako následek původního projektu vytvořeného na *Google Summer of Code*² [49].

Tato knihovna poskytuje algoritmy pro klasifikaci, regresi a shlukování. Díky funkcím knihovny není potřeba stavět tyto algoritmy od počátku, ale lze je částečně brát jako černou skříňku. V projektu pak využívání této knihovny bude stavěno mimo celý projekt a přes přístupové body budu předávat potřebné vstupy, výstupy a další potřebná data. Zároveň

¹<https://scikit-learn.org/stable/>

²<https://summerofcode.withgoogle.com>

však poskytuje funkce pro analýzu, čili pro každý model je možné si jednoduše vykreslit graf, rovnici nebo třeba spočítat odchylku.

Knihovna je dále navržena tak, že vhodně komunikuje s dalšími známými knihovnami pro vědeckou práci. Tyto knihovny se nazývají *NumPy*³ a *SciPy*.⁴

Knihovna poskytuje modely a nástroje jak pro učení s učitelem, tak bez učitele. Natrénovaný model je pak možné exportovat v samostatném modelu jako nový Python soubor a využít jej například přes API a tím navázat například na navrhovaný systém.

5.9.1 Regrese v rámci knihovny

Knihovna tedy poskytuje funkce a model pro práci s regresí. Pro každý druh regrese je nutné dodat vstupní data ve správném formátu (knihovna poskytuje i nástroje pro předzpracování) a poté je případně transformovat na hodnoty vhodnější pro samotný model. Zároveň je nutné rozdělit množinu dat na testovací a trénovací data. Je možné určit zdali se při každém trénování mají tato data rozdělit staticky nebo pokaždé náhodně jinak. Pro využití lineární regrese je hlavně využívána funkce *LinearRegression*.

Pro využití samotné polynomiální regrese je nutné použít funkci *PolynomialFeatures*, která jako parametr přijímá stupeň polynomu. Tato funkce pak transformuje vstupní parametry, tak jak je popsáno v sekci 4.4.1. Ještě před využitím této funkce je nutné transformovat vstupní data pro použití vně modelu. Příklad použití této transformace v programu s využitím knihovny lze vidět v kódu 5.1. Pro samotné trénování lze pak již poměrně jednoduše využít lineární model.

```
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
X = np.arange(6).reshape(3, 2)
poly = PolynomialFeatures(2)
poly.fit_transform(X)
poly = PolynomialFeatures(interaction_only=True)
poly.fit_transform(X)
```

Listing 5.1: Příklad použití polynomiální regrese v knihovně scikit-learn

5.10 Systém predikce výsledků

Klíčovou součástí návrhu modelu predikce byla analýza srovnání některých modelů strojového učení. Po zvážení všech kladných i záporných stránek obou modelů jsem se rozhodl pro využití polynomiální regrese. S tím, že užití komplexnějších neuronových sítí je do budoucna stále možné, proto jsou i v práci několikrát zmíněny.

Klíčovým faktorem zde byla možnost rychlých a jednoduchých změn při trénování tohoto modelu a ladění vzhledem ke kurzům sázkových kanceláří (které u dané množiny dat predikovaných zápasů nebudou dostupné). Dalším faktorem byl cíl predikce výsledků. Fotbalové výsledky v simulátoru nemusejí být úplně přesné (vysokou roli hraje také náhoda nebo nečekané faktory). Výsledky by ale měly odpovídat realitě. Zároveň je pak model vhodnější vzhledem k omezené množině dat k historickým zápasům a dostupné databázi týmu a hráčů. Databáze týmů a hráčů je totiž dostupná pouze za posledních pět ročníků.

³<https://www.numpy.org>

⁴<https://www.scipy.org>

Vzhledem k tomu, že bereme v potaz i vzájemné zápasy týmů, tak lze použít historické výsledky i starší než těchto pět ročníků pro tyto výpočty.

5.10.1 Vstupy predikce

Klíčovou oblastí celé predikce a celé práce bude navržení a správné ohodnocení vstupů. Dle ladění samotného modelu predikce pak bude vybrána podmnožina těchto vstupů. Jako původní atributy byly vybrány tyto vstupy:

- **Pevně dané ohodnocení týmu:** číslo v rozmezí 0–100, získáno z databáze.
- **Aktuální forma týmu:** koeficient, počítaný z posledních deseti zápasů (dva body za výhru, bod za remízu).
- **Průměrný počet vstřelených branek:** tento atribut je především důležitý z hlediska toho, kolik branek v zápasech padne. Bez jeho využití by se odhad výsledku zřejmě zdařil také na základě forem a hodnocení obou týmů, ale například u dvou týmů s vyšším průměrem vstřelených branek by mohlo dojít k zaměnění remízy 2:2 za 0:0.
- **Forma na domácím nebo venkovním hřišti:** výpočet posledních pěti zápasů, dle místa kde se odehrává konkrétní zpracovávaný zápas.
- **Aktuální forma proti soupeři:** výpočet posledních pěti zápasů proti soupeři (2 body za výhru, 1 bod za remízu).
- **Hodnocení hráčů týmu:** v potaz se zde bere průměr hodnocení členů A-týmů (standardně nejlépe hodnocených 23 hráčů v týmu).
- **Aktuální forma hráčů:** zaměřuje se na střelce a obránce, vypočteno z počtu jimi vstřelených či obdržených branek.

5.10.2 Výstupy predikce

Výstupem procesu predikce bude dvojice. Tato dvojice pak reprezentuje skóre obou týmů. Tyto výstupy predikce pak budou ovlivněny náhodou.

5.10.3 Proces predikce

Výstupem predikce je výpočet očekávaných branek. V tomto typu výsledku se neuvažuje náhoda, čili při stejných vstupech bychom měli dostat pokaždé stejný výsledek. Tuto predikci pak bude realizovat model predikce mimo jádro systému, kterému předám odpovídající vstupy pro konkrétní zápas.

Pokud bude model správně natrénován, tak by měl generovat reálné výsledky vzhledem k historickým zápasům. Cílem zde není získat absolutně přesné výsledky (často ovlivněno náhodou), ale především reálné výsledky vzhledem k síle a formě obou týmů. Náhodné faktory budou poté již generovány simulátorem. Realnost predikce pak bude testována na několika faktorech, odhadnutý výsledek by ale měl odpovídat kurzům sázkových kanceláří.

Z historických dat je nutno vytvořit množinu dat vhodnou pro trénování modelu. Na těchto historických datech pak proběhne učení daného modelu dle procesu učení s učitelem (čili máme dané vstupy a výstupy). Poté bude možno tento model využívat k predikci statické části výsledku i ve výsledném systému.

5.10.4 Návrh měření úspěšnosti predikce

Pro ověření úspěšnosti daného modelu jsem vytvořil testovací kritéria. Nejdůležitějším kritériem je reálnost odhadu daného zápasu. Tato reálnost je dána buď kurzem na vítězství týmu nebo silou daného týmu. Síla (hodnocení) týmu je dána z databáze Football Manageru.

Proběhnou dva typy testování:

- **Testování na trénovacích datech:** odhad historických zápasů, které nejsou obsaženy v trénovacích datech, ověření je dle síly týmů.
- **Testování na budoucích zápasech:** odhad zápasů, které se budou konat v blízké budoucnosti, ověření je dle kurzů sázkových kanceláří, viz. sekce 3.9.1.

Obecně jsou zápasy rozděleny na tři typy, konkrétněji v tabulce 5.10:

- **Zápas s jasným favoritem:** zde by favorit měl jasně porazit svého soupeře.
- **Zápas s mírným favoritem:** zde se předpokládá mírné vítězství favorita.
- **Vyrovnaný zápas:** zápas dvou srovnatelných týmů, může skončit jakkoliv.

Pro upřesnění je pak rozmezí pravděpodobností a rozmezí kurzů dáno týmem, který má na svou výhru vyšší pravděpodobnost a tím pádem nižší kurz. Co se týče výpočtu hodnocení, písmeno *A* značí hodnocení (sílu týmu), které je z dvojice vyšší, písmeno *B* pak hodnocení slabšího týmu. Klíčový je pak jejich rozdíl.

Název skupiny	Rozmezí pravděpodobností	Rozmezí kurzů	Rozmezí hodnocení	Příklad (hodnocení)	Příklad (kurz)
Jasný favorit	$\geq 75\%$ na výhru	1.0 - 1.33	Rating A - Rating B > 7	Manchester City (81) - Aston Villa (73)	Manchester City - Watford (1.10 - 12.40 - 28.60)
Mírný favorit	$\geq 60\%$ a $< 75\%$ na výhru	1.34 - 1.66	A - B ≤ 7 a A - B > 3	Liverpool (79) - Southampton (74)	Everton - Liverpool (6.22 - 4.01 - 1.62)
Vyrovnaný zápas	$< 60\%$ a $\geq 33.3\%$ na výhru	1.67 - 3.0	A - B ≤ 3	Chelsea (80) - Tottenham (79)	West Ham - Newcastle (2.21 - 3.38 - 3.58)

Obrázek 5.10: Kategorizace zápasů

V tabulkách 5.11 a 5.12 jsou požadavky na úspěšnost predikce v rámci daného modelu. Požadovaná procenta jsou pak vyšší u detekce reality odhadnutí zápasu než u predikce samotného výsledku. Toto bude klíčová část predikce, která signalizuje uvěřitelnost daných zápasů. Proto jsou na ní kladeny vyšší požadavky. Požadavky na přesnost výsledku pak jsou nižší vzhledem k náhodným faktorům ovlivnění zápasu. Účelem predikce je totiž odhadnout pravděpodobný standardní výsledek, nikoliv brát v potaz faktory náhody.

Typ	Název	Požadovaná úspěšnost	Očekávaný výsledek
Realita výsledku	Jasný favorit	90%	Výhra favorita o 2 a více branek
	Mírný favorit	80%	Výhra favorita
	Vyrovnaný zápas	70%	Zápas může dopadnout jakkoliv, žádný z týmů by ale neměl zvítězit o více než 2 branky

Obrázek 5.11: Požadavky na úspěšnost predikce reality odhadnutí zápasu

Typ	Název	Požadovaná úspěšnost
Odhad výhra/remíza/prohra	Jasný favorit	80%
	Mírný favorit	65%
	Vyrovnaný zápas	30%

Obrázek 5.12: Požadavky na úspěšnost predikce výsledku zápasu

5.11 Systém generování náhody

Každý zápas obsahuje různé náhodné, či pseudonáhodné prvky. Tyto prvky mohou plynout ze starších výsledků, případně přímo ze systému. Je třeba tyto náhodné faktory brát v potaz, protože i v reálných fotbalových zápasech v nemalé míře výsledky ovlivňují. V další podkapitole pak rozepisují tyto jednotlivé faktory.

5.11.1 Faktory náhody

Náhoda je ve fotbalových zápasech závislá i na pevně daných faktorech. Snažil jsem se tudíž vybrat takové faktory, které nejvíce ovlivňují výkony hráčů a tím pádem i výsledky zápasů. Jako náhodu si lze představit i zvýšenou motivaci hráčů ve speciálních případech (třeba ve velkém derby nebo proti oblíbenému soupeři). Z těchto faktorů se pak vypočítá celková pravděpodobnost změny skóre oproti statickému výsledku z modelu predikce zápasu. Proces tvorby takového výsledku pak je popsán v další kapitole.

Faktory náhody tedy budou:

- **minulé vzájemné zápasy – extrémny:** tento prvek je zastoupen i při přesné predikci, zde má ale rozdílný význam. Hledáme extrémny. Tyto extrémny si lze představit jako velké množství proher, či výher se stejným soupeřem. Počítat se bude opět procentuálně vzhledem k maximálním získaným bodům. Pokud je dosaženo extrému pak dojde k vygenerování pozitivní, či negativní motivace, která ovlivní celkovou náhodu týmu.
- **faktor derby:** velká derby jako například El Clásico (souboj Realu Madrid a FC Barcelony) generují často v reálném světě výsledky, které až tak nezávisí na současné

formě, ale jsou více nečekané. Pro hráče mají tyto velké zápasy vyšší důležitost kvůli prestiži a sledovanosti, tudíž je výsledek více náhodný. Tento faktor pak budu simulovat pomocí stupně derby (rozmezí $0-n$), čím vyšší stupeň tím větší nevyzpytatelnost – zvýšení náhody jako simulace zvýšené motivace hráčů.

- **faktor důležitosti zápasu** – tento faktor simuluje určitou motivaci při speciálních zápasech (kupříkladu zápas v 1. kole národního poháru bude generovat více náhodný výsledek než semifinále Ligy mistrů). Opět se zde jedná o zvýšenou koncentraci hráčů, každopádně zde vyšší motivace a soustředění bude vyšší, tudíž se tým více přiblíží ke svým standardním výsledkům.

Tyto tři faktory náhody pak indikují náchylnost zápasu vzhledem k náhodným jevům. Náhodné číslo se generovat nebude, ale bude simulováno při vytváření konečného výsledku (popsáno v další sekci).

5.11.2 Konečný výsledek

Základem tvorby konečného výsledku je pole změn skóre. Každá z položek pole signalizuje změnu skóre oproti predikci. Na začátku budou v tomto poli uloženy především změny o 0 branek. Tedy nízkou vygenerovanou náhodou. Základní pole obsahuje i změny skóre, jejich zastoupení je ale minoritní. Tyto změny mají simulovat nepředvídatelnou náhodou.

Na základě již zmíněných faktorů náhody se postupně do pole budou přidávat další položky. Tyto položky budou obsahovat změnu skóre. V největším zastoupení zde bude změna o 1 branku. V případě více náhodných faktorů mohou být přidány do pole i výraznější změny.

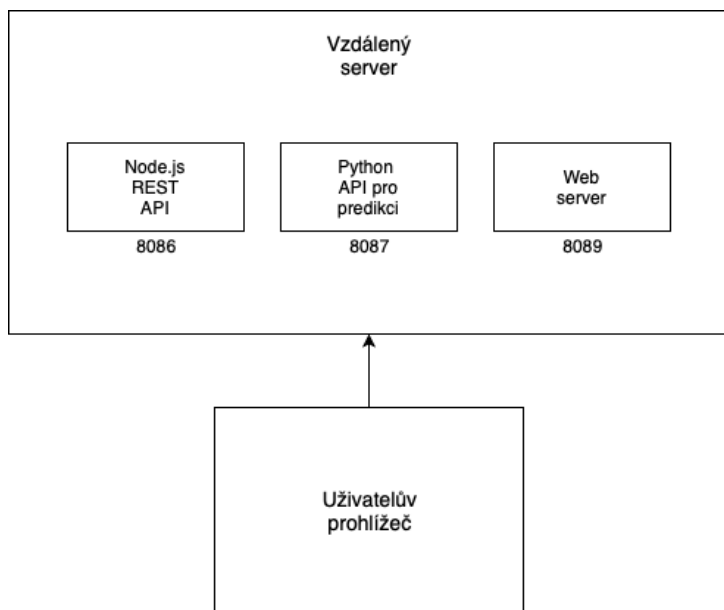
Celkovým výstupem a výsledkem bude tedy výsledek z modelu predikce ovlivněný změnou skóre díky náhodným faktorům. Tím pádem budeme mít dynamické výsledky, které se mění v čase a systém se tak regulovaně bude vyvíjet. Simulátor bude ale zároveň i přinášet nečekané výsledky, tak jak tomu je i v reálném fotbalovém světě.

Kapitola 6

Implementace

Na počátku implemetace byla oddělena logika severu, predikce a klientské části aplikace. Každý tento modul tedy má samostatný repozitář na verzování a z principu je implementován jako samostatný systém. Výhodou takové implementace je, že tyto součásti jsou na sobě navzájem nezávislé a případná chyba tak ovlivňuje jen právě jeden takový modul. Všechny součásti ale běží na jedné serveru, pod stejnou IP adresou, každý ovšem na rozdílném vyhrazeném portu. I samotný webový server má své místo, při vývoji jsem pak skrze vývojové nástroje byl připojení pomocí ssh klíče přímo na tento server a při každém uložení souboru prošlo nahrání těchto změn na server. Strukturu celého systému pak lze vidět na obrázku 6.2.

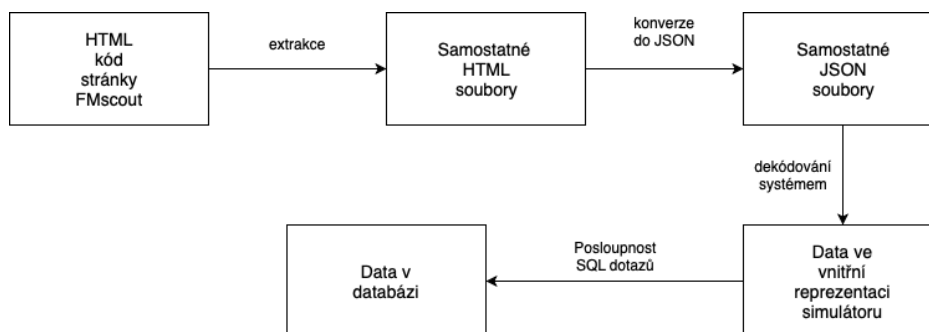
- **Klientská část:** port 8089, webový server v PHP/Nette
- **Serverová část:** port 8086, Node.js REST API
- **Serverová část s predikcí:** port 8087, Python REST API



Obrázek 6.1: Modularita celého systému

6.1 Získávání fotbalových dat

Vzhledem k tomu, že zdroj fotbalových dat nebyla daná databáze, bylo nutné vytvořit soubor skriptů pro naplnění databáze aplikace. Skripty jsou napsány ve dvou jazycích: Python a PHP. Poslouponost činností těchto inicializačních skriptů lze vidět na obrázku 6.2.



Obrázek 6.2: Postup získání a plnění dat v systému

6.1.1 Zisk dat z webu

Časově a zdrojově náročnou operací bylo již dekódování dat z webu FMScout. Ze stránky národů bylo postupně hierarchicky otevřeno a dekódováno jednotlivé webové stránky dle jejich elementů. Daný PHP skript vždy otevřel dekódovanou stránku, uložil daná data a dále otevřel všechny podstránky nutné k zisku dat. Jako příklad můžeme uvést tok **stránka soutěže – otevření všech stránek týmu – pro všechny týmy otevření všech stránek hráčů**. Tento skript trval několik desítek minut, ale výsledkem byla nezpracovaná data z většiny světových fotbalových lig. Obrázky jsou uloženy jako odkaz na soubor, nikoliv jako samotný soubor, z čehož plynula výrazná úspora velikost celé aplikace a tím pádem jednodušší manipulace s jednotlivými moduly.

6.1.2 Zpracování dat

Původním zdrojem je tedy HTML soubor, uložený vždy s identifikátorem stránky (tedy identifikátor týmu, hráče, soutěže apod.). Každý soubor obsahuje tu část původní stránky webu, kde jsou uložena data potřebná pro databázi simulátoru. Dále pak proběhla konverze použitím jazyka Python a knihovny `htm2json`¹ na soubory ve formátu JSON.² Ačkoliv tato konverze nebyla nutná pro uložení dat, chtěl jsem mít uchované soubory k datům ve standardizovaném formátu z důvodu záloh a případného dalšího použití těchto dat.

6.1.3 Integrace dat do databáze aplikace

Když jsou již k dispozici soubory ve formátu JSON, je nutné tyto soubory v rámci vnitřní implementace systému dekódovat (využití PHP funkce `json_decode`.³) Z této vnitřní implementace pak mohly být provedeny jednotlivé SQL skripty, které naplní příslušné tabulky v databázi daty. Tyto inicializační skripty vkládají data do tabulek, které se v průběhu času

¹<https://pypi.org/project/html2json/>

²<https://www.json.org>

³<https://www.php.net/manual/en/function.json-decode.php>

nebudou měnit. Na základě těchto dat se pak každému uživateli při registraci vytvoří mělká kopie uživatelských dat, která se vyvíjí v průběhu simulace.

Tento statický stav původních dat je pak výhodný při resetování simulace uživatelem, kde se kopírují právě tato původní data. V budoucnu pak při každé aktualizaci na novou sezónu po uzavření hlavních přestupových oken (pravděpodobně v průběhu října každého roku) proběhne celá tato procedura znovu a simulátor si tak aktualizuje svá statická data. Uživatelé budou mít k dispozici po určitý čas zálohy svých dat a budou moci ještě pár týdnů po aktualizaci fungovat na datech z minulé sezóny.

6.2 Struktura statických a uživatelských dat

Jak jsem již výše zmínil, klíčovou oblastí tvorby systému byla implementace a rozdělení dat statických a dat uživatelských. Na diagramu 6.7 lze vidět propojení tabulek *hráčů*, *týmů* a *soutěží*. Klíčovým faktorem zde je pak simulační identifikátor, který je shodný s uživatelským identifikátorem (v budoucnu bude pozměněno a uživatel bude moci mít více svých simulací zároveň). Tento identifikátor se pak využívá při detekci konkrétní simulace a především v požadavcích v rámci Node.js REST API.



Obrázek 6.3: Provázání statických a uživatelských tabulek

Jednotlivé tabulky tak byly rozděleny na tabulku statickou a tabulku s ní provázanou, která ve své původní podobě nese kopii některých hodnot atributů tabulky statické. Při průběhu simulace každého uživatele se pak mění stav pouze tabulek vazebných a návrat k původní podobě je kdykoliv možný. Tudíž je možné, aby se každému uživateli simulace vyvíjela jinak. Jako příklad můžeme například uvést hodnocení týmu. To je na počátku pevně dané ze zdroje, ale může se po každé sezóně vyvíjet jak pozitivně (výkony nad očekávání, vysoký potenciál), tak negativně (výkony pod očekávání, nižší potenciál). Je tedy potřeba mít tuto hodnotu uloženou jak ve svém původním stavu, tak mít možnost ji v průběhu času měnit.

V budoucím vývoji simulátoru pak budou data například v tabulkách hráčů oproti statickým datům více rozdílná, protože dojde k nahrazování končících hráčů dorostenci. Budou zde také probíhat například přestupy mezi týmy. Je tedy pravděpodobné, že čištění stavu databáze v budoucnu bude náročnější na čas i zdroje.

6.3 Architektura a závislosti systému

Po vytvoření inicializačních skriptů bylo potřeba navrhnout architekturu samotného samotného jádra systému. Toto jádro bylo postaveno především na frameworku *Nette*⁴ a jazyce PHP s verzí 7.2. Tento framework poskytuje řadu nástrojů, knihoven a již předpřipravených komponent pro tvorbu webových systémů a aplikací.

Jádro tohoto frameworku je postaveno na technologii *Model-view-presenter*, dále zkráceně MVP. Struktura této architektury je postavena na těchto podčástech:

- **Modely:** tato část aplikace slouží především ke komunikaci se zdroji přes REST API, slouží k práci s daty.
- **Presentery:** slouží zde jako prostředník mezi datovou vrstvou a vrstvou prezentační, zpracovávají data z modelů a předávají do šablon k požadované podobě.
- **Šablony:** slouží k vykreslení zpracovaných dat uživateli.

6.3.1 Řešení závislostí

Návaznost modelů a presenterů je pak řešena pomocí tzv. *dependency injection*. Jedná se o předávání závislostí třídám, kterým díky tomu můžeme předat jednoduše bez statického vytváření nových objektů zevnitř této třídy. Lze pak jasně a exaktně předávat závislosti. Tyto závislosti se pak dají předávat různými způsoby, o správné předávání se ale při vhodném nastavení konfiguračních souborů stará *Dependency injection container* [4].

Pro řešení závislostí vzhledem k externím nástrojům a knihovnám se využívá nástroj *Composer*. V projektu existuje soubor `composer.json`, v němž jsou pod klíči jednotlivých knihoven uloženy dané verze. Pomocí příkazu `composer install` dochází k instalaci daných definovaných knihoven. Pomocí příkazu `composer update` dochází k vynucení aktualizace daných knihoven. Knihovny není nutné mít uloženy lokálně, stačí pouze daný soubor a při nové instalaci se tyto knihovny nainstalují nezávisle. Tento postup tedy šetří místo v daném repozitáři.⁵

⁴<https://nette.org/cs/>

⁵<https://getcomposer.org/doc/01-basic-usage.md>

6.3.2 Konfigurační soubory

Pro architekturu jsou důležité i konfigurační soubory aplikace, které jsou samostatné pro různá prostředí (vývojové a produkční), zde pak lze nastavit například rozdílný zdroj databáze, nastavení maximální odezvy serveru nebo právě vložit, či změnit odkazy na jednotlivé modely. Každý model díky tomu funguje jako samostatná služba a jeho vložení (především do presenterů) je jednoduché a rychlé. Při změně umístění, či parametrů modelu pak stačí změnit jako vlastnosti právě v těchto souborech. Konfigurační soubory jsou pak napsány ve standardizovaném formátu *NEON* (*nette object notation*).⁶

6.4 Serverová část

Ze systému pomocí jeho modelů přicházejí požadavky na čtení, úpravy a vkládání dat. Pro komunikaci slouží centrální třída `RestApi`, která obsahuje metody pro provedení požadavku GET, PUT, POST a DELETE. Z této třídy pak dědí všechny ostatní modely, z nichž se předává URL pro provedení požadovaného požadavku. Modely pak většinou odpovídají daným entitám v serverové části aplikace.

6.4.1 Nastavení serveru

Pro běh serverové části aplikace se využívá *Virtuální privátní server*, zkráceně dále jen VPS. Na tomto serveru pak běží na daném portu REST API, které odpovídá na požadavky z klientského serveru. Tato služba pak běží v nekonečném cyklu a v případě pádu nebo pádu serveru se pomocí nástroje *forever.js*⁷ dokáže sama obnovit nebo restartovat. Na serveru je pak nastaven přehledný logovací systém, který umožňuje v případě fatálních chyb zpětně trasovat problém a pomoci tak s jeho vyřešením.

6.4.2 Node.js REST API

Architektura serverové částí většinou odpovídá modelům aplikace. V podkapitolách této části pak popisují jednotlivé entity a mapují všechny požadavky této součásti. Jak již vychází z návrhu, klíčovými částmi jsou pak modely (serverové části aplikace, nespojují se systémem), v nichž jsou nativní dotazy do databáze. Databáze pak běží na stejném virtuálním serveru, ale na jiném portu. Klíčové pak v této části bylo implementování co nejoptimálnějších dotazů, tak aby i byla odezva přijatelná i při rozsáhlejší množině dat. Pro každý model zde popisují jeho funkci a implementované zdroje. U každého zdroje je pak uveden typ metody, URL metody a slovní popis metody. Pro dokumentaci zdrojů tohoto API jsem pro zobrazení využíval nástroj *Swagger*.⁸ Pro instalaci knihoven a dalších nástrojů do tohoto typu systému je využíván balíčkovací systém *npm*.⁹

Model globálních informací

Tento model slouží k získávání globálních informací o stavu simulace. Jako typy získaných dat zde můžeme uvést například aktuální sezónu nebo aktuální týden probíhající simulace.

⁶<https://github.com/nette/neon>

⁷<https://github.com/foreverjs/forever>

⁸<https://swagger.io>

⁹<https://www.npmjs.com>

Model	Metoda	URI	Popis
Global	GET	/global/{simulationId}	získ globálních informací pro konkrétní simulace
	PUT	/global/reset/{simulationId}	původní stav globálních informací
	PUT	/global/addOneWeek/{simulationId}	přidá jeden týden ke globálnímu stavu simulace
	PUT	/global/resetAfterSeason/{simulationId}	nastaví stav globálních informací na novou sezonu

Obrázek 6.4: Struktura modelu s globálními informacemi

Model soutěže

Tento model především spravuje informace o konkrétních soutěžích, souvisí pak výrazně s modelem pro země a využívá se například i při nulování simulace. Data z tohoto modelu jsou pak v systému využita především na obrazovce ke konkrétní soutěži.

Model	Metoda	URI	Popis
Competitions	GET	/competitions/{simulationId}	získ všech soutěží
	GET	/competitions/specific/{competitionId}/{simulationId}	získ konkrétní soutěže
	GET	/competitions/nation/{nationId}/{simulationId}	získ soutěže pro konkrétní zemi
	PUT	/competitions/set/{actualRound}/{competitionId}/{simulationId}	změní aktuální kolo soutěže
	PUT	/competitions/reset/{simulationId}	nastaví původní stav soutěže před startem simulace

Obrázek 6.5: Struktura modelu soutěže

Model země

Jednoduchý model, obsahuje pouze jednu metodu pro získání informací o daném národu, strukturu lze vidět v tabulce 6.6. Tato tabulka se v průběhu simulace nemění a pro evoluci dat se přistupuje k jednotlivým soutěžím, na které je z každé entity národ odkazováno. Do budoucna se model rozroste, jakmile budou přidány mezinárodní reprezentační soutěže a reprezentace budou mít v simulátoru svoji roli.

Model	Metoda	URI	Popis
Nations	GET	/nations	získá informace o všech zemích

Obrázek 6.6: Struktura modelu země

Model tým

V rámci poměrně rozáhlého modelu s týmu se kromě získávání informací o týmech používají hlavně metody pro manipulaci s daty týmu. Jsou zde například požadavky pro aktualizaci formy týmu po odehrání zápasu, která je důležitá pro predikci dalších zápasů. Dále se zde pak po zápasech přepočítávají statistiky týmu. Probíhá zde i přepočítání hodnocení týmu po odehrané sezóně nebo v případě postupu, či sestupu do jiné soutěže změna identifikátoru soutěže. Strukturu modelu lze vidět v tabulce níže.

Model	Metoda	URI	Popis
Teams	GET	/teams/{teamId}/{simulationId}	získá konkrétního týmu
	PUT	/teams/setForm/{teamId}/{simulationId}/actualForm/{actualForm}/actualHomeForm/{actualHomeForm}/actualAwayForm/{actualAwayForm}/actualAvgGoals/{actualAvgGoals}	po rekalkulaci nastavení formy konkrétního týmu
	PUT	/teams/setStats/{teamId}/{simulationId}/wins/{draws}/{losses}/{goalsFor}/{goalsAgainst}	aktualizace statistik daného týmu
	PUT	/teams/rating/{newRating}/{teamId}/{simulationId}	změna síly týmu po sezóně
	PUT	/teams/nullTeams/{simulationId}	resetování týmu do původního stavu
	GET	/teams/competition/{competitionId}/{simulationId}	získá týmů v konkrétní soutěži
	GET	/teams/competitionWinner/{competitionId}/{simulationId}	získá vítěze soutěže po jejím konci
	PUT	/teams/changeLeague/{teamId}/{simulationId}/{newLeagueId}	změna soutěže pro tým po postupu nebo sestupu

Obrázek 6.7: Struktura modelu týmu

Model hráč

Data z tohoto modelu se především týkají informací o hráčích vzhledem k danému týmu, ale jsou využity například i u seznamu nejlepších střelců dané soutěže. Používá se také například při simulaci samotného zápasu, kde postupně přidává střelcům branek do jejich statistik vstřelené branky. Strukturu tohoto modelu lze vidět v tabulce 6.8.

Model	Metoda	URI	Popis
Players	GET	/players/{teamId}/{simulationId}	zisk hráčů z konkrétního týmu
	GET	/players/{playerId}/{simulationId}	zisk konkrétního hráče
	PUT	/players/nullPlayers/{simulationId}	nulování statistik hráčů (vstřelené branky apod.)
	PUT	/players/addGoal/{playerId}/{simulationId}	přidání vstřelené branky hráči
	GET	/players/topScorers/{teamId}/{simulationId}	zisk nejlepších střelců v rámci týmu
	GET	/players/topScorersLeague/{leagueId}/{simulationId}	zisk deseti nejlepších střelců soutěže

Obrázek 6.8: Struktura modelu hráče

Model uživatel

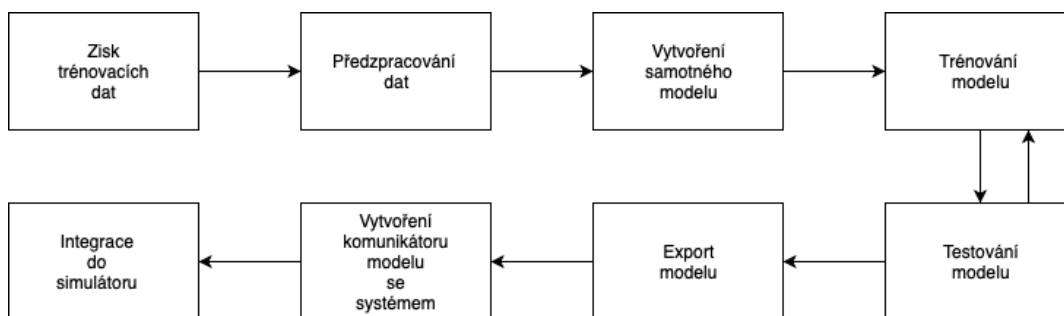
Zde je obdobná situace jako u modelu pro správu dat spojených s národem, nyní je model jednoduchý, jeho strukturu lze vidět v tabulce 6.9. V budoucích aktualizacích ale bude přidána možnost úprav dat přímo uživatelem a správa uživatelů pak bude mít důležitější roli.

Model	Metoda	URI	Popis
User	GET	/users/{userId}	zisk informace o konkrétním uživateli
	GET	/users	zisk všech uživatelů
	POST	/users/save	uložení/změna dat uživatele
	DELETE	/users/{userId}	smazání uživatele

Obrázek 6.9: Struktura modelu uživatele

6.5 Implementace predikce zápasu

V této sekci je popsána implementace samotné predikce, od předzpracování vstupních dat až po integraci výstupu do samotného systému simulátoru. Postup samotné implementace pak lze vidět na obrázku 6.10. Jak je již zmíněno v návrhu na implementaci predikce je použit jazyk Python, hlavní roli zde hraje knihovna *scikit-learn*. Další využití knihovny jsou pak zmíněny v konkrétních příslušných sekcích.



Obrázek 6.10: Postup tvorby predikce zápasu

6.5.1 Zisk relevantních dat

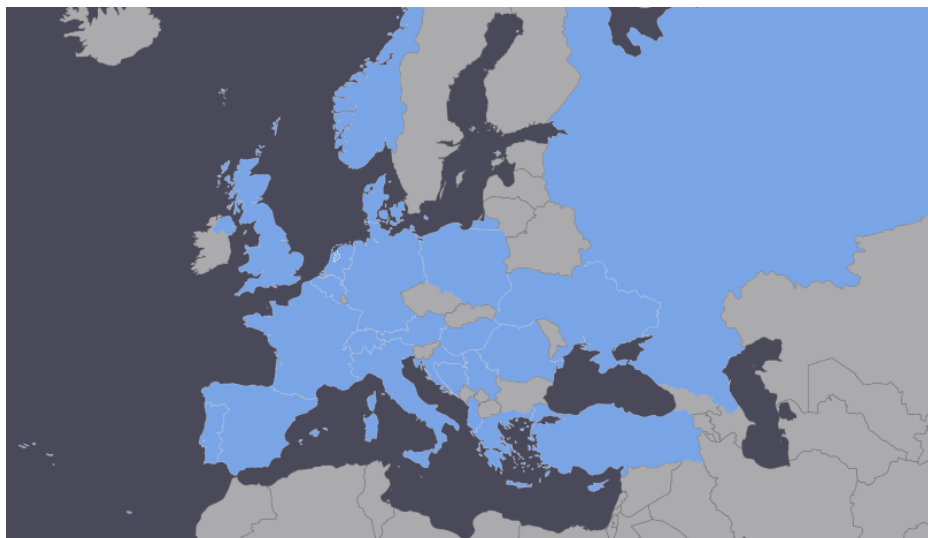
Data pro predikci jsou čerpána ze dvou zdrojů: dat pro systém a z REST API ze služby API Football.¹⁰ Samotné získání dat pro predikci byl poměrně komplexní problém. Nejdříve bylo nutné zmapovat jak široká množina dat bude. Databáze systému má k dispozici data pro týmy za posledních šest let. Zároveň data ze služby API FOOTBALL, byť jsou poměrně rozsáhlá, neobsahují všechny soutěže zemí UEFA. Bylo tedy nutné spojit tato data dohromady.

Ze systému je možné získat údaje o všech týmech nejvyšších evropských soutěží. Klíčovými atributy jsou zde **síla týmu** a **síla hráčů v týmu**. Tato data jsou zde pak uložena i historicky, takže je podstatné v kterém roce se daný zápas hraje. Z externího API (konkrétně 22 lig v rámci UEFA, viz. obrázek 6.11 a Liga Mistrů s Evropskou ligou) se pak načítají původní data pro **formu týmu**, **formu týmu na daném hřišti** a **průměrný počet vstřelených branek**. Tato data jsou v původní podobě, čili například pro formu týmu je potřeba načíst odpovídající počet minulých zápasů daného týmu.

Data z posledních pěti let jsou pak dokumentována jako pokrytá. Nelze ale do množiny dat zahrnout i poslední pátý rok dat, vzhledem k tomu, že by nebyla dostupná data pro výpočet předchozích zápasů a formy. Rozhodl jsem se tedy zahrnout výsledky za poslední čtyři sezóny (od ročníku 2014/2015 po ročník 2017/2018). Celkově pak množina dat čítá tedy čítá okolo **23000** záznamů. Tato množina ovšem není celá určena pro trénování modelu, budou z ní ještě odčerpána testovací data.

V množině dat jsou obsaženy ještě další vstupní atributy, které ale ve finální podobě v modelu nejsou, jsou zde ale uložena pro potenciální budoucí využití, proto dále zmiňují jen vybrané atributy. Redukce atributů proběhla pak v rámci testování a snahy o nejlepší procentuální úspěšnost predikce systému.

¹⁰<https://www.api-football.com>



Obrázek 6.11: Obsažené nejvyšší soutěže zemí z API FOOTBALL

6.5.2 Zpracování dat

Před dalším zpracováním je ještě potřeba zjistit, ve které sezóně se daný zápas hraje. Tento parametr totiž ovlivňuje data s hodnocením hráčů a týmů, zároveň se podle něj určuje počáteční bod počátku výpočtu forem a průměru branek.

Pro každý obsažený zápas v modelu jsou pak data zpracovávána. Pro oba týmy je spočtena jejich celková forma za posledních deset zápasů, forma na hřišti na němž v zápase hrají za posledních pět zápasů, průměrný počet branek za posledních patnáct zápasů a je zprůměrováno hodnocení hráčů týmu. Formy pak reprezentují desetinná čísla, která jsou vypočtena na základě výher, remíz a proher týmu v daném časovém období. Tento postup se pak drobně liší mezi celkovou formou a formou na daném hřišti. Takto zpracovaná data se pak vkládají postupně do množiny již zpracovaných dat. Zároveň se k nim přidávají výsledky daného zápasu. Každý takto zpracovaný zápas obsahuje svůj jedinečný identifikátor, pod nímž je uložen a díky kterému je možné z množiny odstranit zápasy určené k testování.

Do výsledné množiny dat také nejsou vkládány zápasy, ke kterým není dostatek podkladů. Takových příkladů je minoritní množství a vyskytly se u soutěží, které nemají úplnou množinu dat k soutěžím nižších úrovní. Po postupu jednoho z nezpracovaných týmů pak v prvních kolech nového ročníku nebylo možno spočítat minulou formu.

6.5.3 Struktura celého modelu

Jak již zmiňuji, pro model predikce jsem zvolil polynomiální regresní funkce. Samotný model se pak skládá ze dvou regresních funkcí, které jako své vstupy přijímají tato již zpracovaná vstupní atributy dle návrhu (viz. sekce 5.10.1). Každá regresní funkce pak má jeden výstup – jedna vrací počet branek domácích a druhá obsahuje počet branek hostů. Z celkového modelu se pak vrací tato dvojice výsledků dohromady. Po vyexportování modelu pak stačí poslat jeden požadavek na vrácení celkového výsledku.

6.5.4 Trénování modelu

Před samotným trénováním jsou z množiny dat odebrána testovací data pro testování validity predikce. Tato data jsou vybrána tak, aby odpovídala kategoriím pro testování úspěšnosti (viz. sekce 5.10.4). Zbylá data jsou pak rozdělena v poměru 80/20 na data trénovací a data testovací vzhledem k danému modelu. K tomuto účelu se pak využívá funkce `train_test_split`.¹¹

Trénování pak probíhá pomocí transformace vstupních atributů do jejich polynomiální podoby. Pro tuto transformaci slouží funkce `PolynomialFeatures`, která jako klíčový atribut přijímá stupeň polynomu. Zároveň jsem při trénování také využíval pozitivně nastavený parametr `include_bias`. Tento parametr přidává sloupec, jehož všechny parametrické složky jsou nastaveny na nulu (sloupec jedniček) – funguje jako záchytný bod v lineárním modelu. Testování různých stupňů polynomů je popsáno v dalších sekcích, v rámci trénování jsem ale zkoušel rozmezí od stupně 2 po stupeň 20. Finální křivka natrénovaného je pak kubická. Poté již mohu na přestransformovaných datech použít lineární model pomocí funkcí `fit` a `fit_transform` [19]. V kódu 6.1 pak lze vidět postup trénování části modelu od transformace atributů až po evaluaci modelu.

```
# vytvoreni atributu
poly_features = PolynomialFeatures(include_bias=True, degree=3)

# transformace atributu
X_train_poly = poly_features.fit_transform(X_train)

# linearni model
poly_model = LinearRegression()
poly_model.fit(X_train_poly, y_train)

# predikce, trenovaci data
y_train_predicted = poly_model.predict(X_train_poly)

# predikce, testovaci data
y_test_predict = poly_model.predict(poly_features.fit_transform(X_test))

# evaluace modelu na trenovacich datech
rmse_train = np.sqrt(mean_squared_error(y_train, y_train_predicted))
r2_train = r2_score(y_train, y_train_predicted)

# evaluace modelu na testovacich datech
rmse_test = np.sqrt(mean_squared_error(y_test, y_test_predict))
r2_test = r2_score(y_test, y_test_predict)
```

Listing 6.1: Proces trénování

¹¹https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

6.6 Testování validity predikce a integrace

Vzhledem k tomu, že testování validity probíhalo již v rámci implementace a na základě něj byly měněny a zkoušeny různé kombinace atributů, formát atributů nebo stupeň polynomu funkce, rozhodl jsem se testování validity zařadit do této sekce.

6.6.1 Rámcové ověření rozsahu výsledků

Úplně prvním krokem testování bylo experimentální ověření rozmezí stupně polynomu, v němž model generoval smysluplné výsledky. Výsledky tohoto ověření pak lze vidět v následující tabulce. Při tomto prvotním ověření jsem se ještě nezaměřoval na konkrétní úspěšnost, jen bylo potřeba zúžit množinu stupňů polynomů na při použití daných atributů.

Stupeň polynomu	Rozmezí výsledků	Závěr
degree = 2	jediné dva typy výsledků, vyskytující se v tomto modelu bylo skóre 2:1 a 1:2.	nevhodné pro další použití
degree = 3	skóre variabilní a uvěřitelná	vhodné
degree = 4	skóre variabilní a uvěřitelná	vhodné
degree = 5	některé zápasy mají uvěřitelné skóre, některé ovšem končí výsledkem nereálným (např. 12:5)	nevhodné pro další použití
degree = 6 a více	poměr reálných a nereálných skóre je horší než o stupeň níže	nevhodné pro další použití

Obrázek 6.12: Výsledek experimentálního ověření

Dále probíhaly jednotlivé iterace testování. V těchto iteracích probíhalo jak testování na testovacích historických datech, tak posléze dle úspěšnějšího modelu na reálných zápasech v soutěžích, které jsou zahrnuty i v simulátoru. Správnost výsledků pak byla porovnávána i s výsledky odhadu sazkové kanceláře Fortuna (jako tip byl považován nejnižší kurz). Výsledky testování na aktuálních zápasech ale nebyly brány v potaz vzhledem k ověření správnosti predikce. Klíčový faktor zde byl především shoda s kurzem. Každý víkend totiž obsahoval zhruba 50 zápasů a každý překvapivý výsledek pak výrazně mění procentuální úspěšnost. Testování na aktuálních datech pak slouží spíše jako porovnávání modelů mezi sebou a se sázkovou kancelář. Mezi iteracemi pak probíhaly i změny konkrétních atributů, z důvodu stručnosti pak v iteracích popisují jen změny, které vedly k zlepšení úspěšnosti predikce.

6.6.2 První iterace testování

Predikce historických zápasů

Po prvním výběru se stupeň polynomu zúžil na kubický a bikvadratický. Následovalo testování na připravených datech k tomuto účelu. Tato připravená historická testovací data byla rozdělena do kategorií vyrovnaných zápasů, zápasů s jasným favoritem a zápasů s mírným favoritem vzhledem k definici těchto kategorií. Kritéria reálnosti a správnosti predikce pak

vycházejí z jejich návrhu v sekci 5.10.4. Výsledky na těchto historických datech lze vidět v tabulce 6.13. Z tabulky vyplývá, že dle množiny historických tipovaných výsledků dosahoval lepších výsledků model se 4.stupněm funkcí. Kategorizace se zde počítala vzhledem k síle týmů.

Stupeň	Kategorie	Realističnost	Správnost
3	Jasný favorit	89%	80%
3	Mírný favorit	77%	63%
3	Vyrovnaný zápas	62%	34%
4	Jasný favorit	92%	82%
4	Mírný favorit	83%	67%
4	Vyrovnaný zápas	70%	40%

Obrázek 6.13: 1.iterace testování – historická data

Predikce aktuálních zápasů

Oba modely jsem si uložil a provedl predikce víkendových zápasů. Dané kurzy jsem získal ze sázkové kanceláře Fortuna. Kurzy jsem si uložil v pátek dopoledne, čili jsem již poté nezahrnoval jejich drobné změny. Výsledky prvního testování na reálných zápasech lze vidět v tabulce 6.14. Oba modely ovšem zde mají problém i s určováním výsledků zápasů s favority. Dalším problémem tohoto testování byl nadměrný počet branek v zápasech. Výsledky byly uvěřitelné, ale oproti realitě zde padalo mnohem více branek, časté výsledky 0:0 nebo 1:0 se zde vyskytovaly spíše sporadicky. Následovala pak druhá iterace testování. Zároveň pak odehrané zápasy byly přidány do množiny trénovacích dat.

Stupeň	Kategorie	Realističnost	Správnost	Sázková kancelář
3	Jasný favorit	75%	75%	75%
3	Mírný favorit	71%	57%	71%
3	Vyrovnaný zápas	57%	29%	39%
4	Jasný favorit	100%	75%	
4	Mírný favorit	85%	57%	
4	Vyrovnaný zápas	61%	35%	

Obrázek 6.14: 1.iterace testování – aktuální zápasy

6.6.3 Druhá iterace testování

Změny v rámci modelu

Klíčovou částí druhé iterace testování bylo zhodnocení stávajících vstupních atributů a odladění modelu. Testování na historických testovacích datech zde proběhlo několikrát. Nejlepších možných výsledků pak byla výměna původních atributů. Atribut formy týmu proti

soupeři byl nahrazen za průměrný počet vstřelených branek. Poslední zápasy proti danému týmu jsem místo toho zařadil do faktorů náhody. Průměrný počet branek pak značně pomohl aproximovat výsledky do reálnější podoby (ve výsledku ubylo nadměrně vstřelených branek). Model byl zkoušen i s výraznějšími změnami, finální podoba ale zahrnovala pouze tuto změnu. Dále byl ještě při transformaci atributů přidán již zmíněný *bias*.

Predikce historických zápasů

Samotné testování predikce pak proběhlo obdobně jako v první iteraci na stejných množinách kategorizovaných testovacích dat. Z tabulky na obrázku lze vidět výrazně zlepšení modelu s použitím kubických funkcí. Zásadní bylo toto zlepšení především při odhadování správnosti vyrovnaných zápasů. Zlepšení ale nastalo i ve všech ostatních faktorech. Model se 4.stupněm polynomu naopak zaznamenal drobné zhoršení úspěšnosti. Toto zhoršení zde ale přičítám obecnému generování méně branek v zápasech. Výrazným rozdílem oproti minulé iteraci tedy byl hlavně výrazný nárůst úspěšnosti modelu se 3.stupněm polynomu.

Stupeň	Kategorie	Realističnost	Správnost
3	Jasný favorit	96%	86%
3	Mírný favorit	85%	71%
3	Vyrovnaný zápas	71%	49%
4	Jasný favorit	91%	80%
4	Mírný favorit	82%	67%
4	Vyrovnaný zápas	69%	37%

Obrázek 6.15: 2.iterace testování – historická data

Predikce aktuálních zápasů

Zde nastal obdobný postup jako v minulé iteraci. Dvojici modelů jsem testoval na aktuálních víkendových zápasech. U obou modelů nastalo zlepšení odhadu. U modelu s kubickým polynomem bylo toto zlepšení výrazné. Zároveň se také výrazně snížila odchylka celkového počtu vstřelených branek oproti realitě. Vybral jsem tedy tento model pro finální otestování.

Stupeň	Kategorie	Realističnost	Správnost	Sázková kancelář
3	Jasný favorit	100%	80%	80%
3	Mírný favorit	88%	63%	63%
3	Vyrovnaný zápas	77%	51%	71%
4	Jasný favorit	80%	82%	
4	Mírný favorit	75%	50%	
4	Vyrovnaný zápas	69%	39%	

Obrázek 6.16: 2.iterace testování – aktuální zápasy

6.6.4 Finální testování

Finální testování pak proběhlo především na aktuálních zápasech. Účelem tohoto dalšího testování bylo ověření splnění kritérií daným natrénovaným modelem. Na historických výsledcích zůstala úspěšnost stejná u obou modelů. Úspěšnost na aktuálních zápasech vzhledem k realističnosti zůstala také velmi podobná, drobné změny lze přičíst tomu, že se hrály zápasy v jiném složení a náhodným faktorům. Výsledky tohoto testování lze vidět na obrázku 6.17. Byl tedy použit model s kubickými křivkami v regresích.

Ná základě všech těchto srovnání lze říct, že model se třetím stupněm polynomu umí spolehlivě odhadnout reálný výsledek při zápasech výraznějších, či méně výrazných favoritů. Zároveň dosahuje i přijatelné realističnosti při vyrovnaných zápasech. Co se týče srovnání s úspěšností sázkové kanceláře, tak tam model odhaduje s podobnou úspěšností při zápasech favoritů. Sázková kancelář má pak výraznější úspěšnost při vyrovnanějších zápasech, to lze ale přičíst manuálním úpravám bookmakery a širším množstvím dostupných dat.

Stupeň	Kategorie	Realističnost	Správnost	Sázková kancelář
3	Jasný favorit	100%	67%	67%
3	Mírný favorit	88%	63%	75%
3	Vyrovnaný zápas	77%	46%	60%
4	Jasný favorit	83%	67%	
4	Mírný favorit	75%	50%	
4	Vyrovnaný zápas	69%	37%	

Obrázek 6.17: 3.iterace testování – aktuální zápasy

6.6.5 Integrace do systému

Po ověření samotných výsledků a validity predikce bylo nutné daný model vyjmout a uložit jej pro možné použití v simulátoru. Samotný export je poměrně jednoduchý a realizuje ho modul `pickle`. Tento modul umožňuje serializaci a deserializaci objektů v jazyce Python. K těmto procesům pak využívá převod objektů na reprezentaci bajtů [20].

Takto exportovaný model přijímá jako své parametry identifikátory obou týmů a data ze systému (hodnocení hráčů a týmu). Na základě identifikátorů týmů je pak spuštěn skript, který zpracovával trénovací data. Tento skript je schopen získat již zpracované atributy, které natrénovaný model potřebuje doplnit jako své vstupy. Tato logika je pak dále zapouzdřena v jednoduché REST API architektuře. Tato architektura běží na rozdílném portu než rozhraní pro práci s databází a pomocí jediného zdroje vrací výsledek simulace zápasu. Tento výsledek je pak vložen do systému a ten s ním provádí další operace.

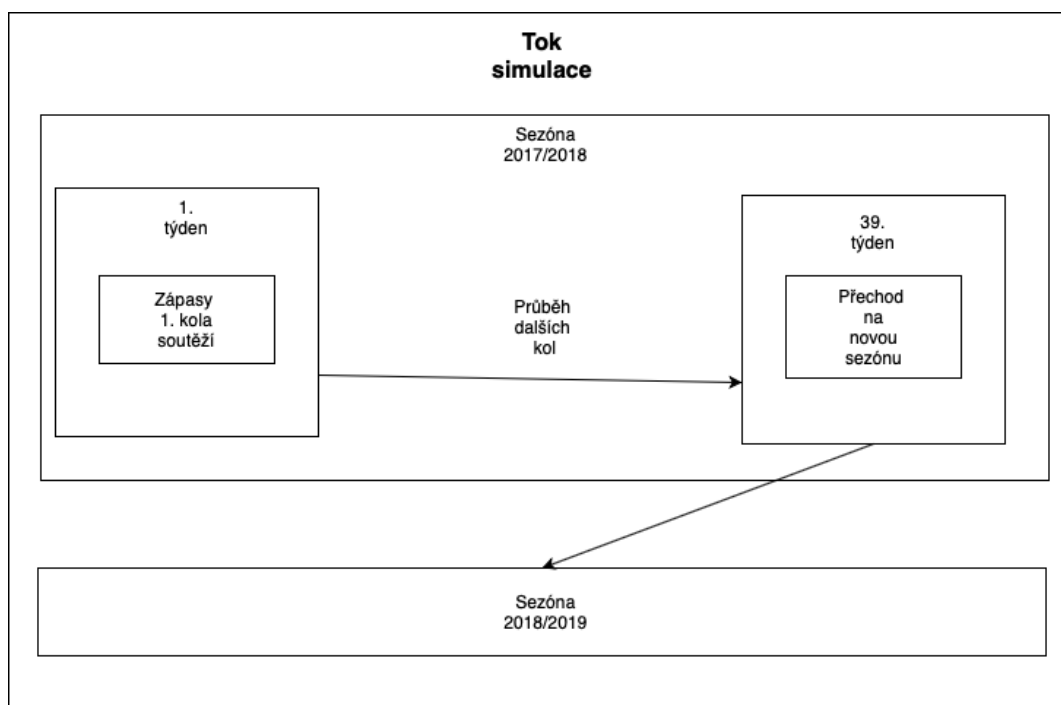
6.7 Implementace simulátoru

V této sekci bude popsána implementace klíčových systému a algoritmů v simulátoru. Těmito algoritmy můžeme rozumět například rozlosování zápasů systémem každý s každým nebo systém postupů a sestupů z daných ligových soutěží. Dále pak zde je popsána klientská

část aplikace, což zahrnuje například postup vytváření asynchronních událostí na popředí nebo tvorbu samotných šablon.

6.7.1 Tok celé simulace

Simulace probíhá po jednotlivých sezónách. Sezóna pak odpovídá jednomu odehranému ročníku soutěže. Tato sezóna je pak rozdělena na jednotlivá kola, respektive zde týdny. Stav simulace je tedy nastaven na počátku na 1.kolo dané sezóny. Poslední týden pak má v současné době číslo 39, zde se již soutěže nehrají, ale tento týden je nazván koncem sezóny. Soutěže začínají všechny shodně a pro ligy s méně než 38 celkovými koly končí dříve než ostatní. V rámci jednoho kola je pak potřeba mít odsimulovány všechny naplánované zápasy. Uživatel může tyto zápasy simulovat buď zvlášť nebo po soutěžích. Přechod do dalšího kola není možný, pokud tyto zápasy nejsou odehrány. V rámci budoucí implementace pak každý týden má i příznak buď ligový nebo pro evropské soutěže. Přehledněji lze pak tok simulace vidět na obrázku 6.18



Obrázek 6.18: Znázornění toku simulace

6.7.2 Ligový systém

Každá implementovaná ligová soutěž vyžaduje aby všechny týmy utkaly mezi sebou dvakrát (jednou na hřišti každého z dvojice). Při generování rozpisu zápasů pak bylo potřeba do paměti ukládat s jakými týmy se daný tým již utkal. Algoritmus vytvoření rozpisu zápasů pak vypadal následovně:

1. Získání všech týmů dané soutěže.
2. Získané pole je pak náhodně přeskládáno pro unikátní pořadí zápasů.
3. Nastavení čítače kol na hodnotu 0.

4. Inkrementace čítače kol.
5. Pro všechny týmy nalezení soupeře s nímž se v rámci soutěže ještě nepotkal a který již v daném kole neparticipuje a vygenerování daného zápasu.
6. Zároveň vygenerování zápasu pro druhou polovinu soutěže s opačným hostitelem utkání.
7. Pokud čítač není roven hodnotě počet týmů - 1, pak návrat na krok 4, jinak pokračování.
8. Ukončení generování zápasů a uložení všech zápasů v daném pořadí.

Po vygenerování seznamu zápasů se nastavuje čítač týdnů na počáteční hodnotu. S každým odsimulovaným kolem soutěží se inkrementuje aktuální kolo soutěže a týden simulace. Týmy jsou v tabulce seřazeny dle získaných bodů, v případě rovnosti bodů je pak řazení nastaveno pomocí následujících kritérií:

1. rozdíl ve skóre,
2. vyšší počet vstřelených branek,
3. vyšší počet výher,
4. nižší počet proher,
5. pořadí v abecedě.

Ligová pravidla se pak řídí standartními fotbalovými pravidly. Každá liga má staticky definovaný počet sestupujících, dále pak postupujících do evropských soutěží (zde dle daného koeficientu národu). K lize je pak přiřazena i její reputace, která v budoucnu bude určovat jak kvalitní hráči mohou do dané soutěže přijít.

6.7.3 Tvorba výsledku zápasu

Klíčovou součástí vygenerovaného výsledku tvoří výsledek z modelu predikce. V systému pak následuje ovlivnění náhodou, jenž je vypočteno tímto postupem:

1. **Vytvoření původního pole:** inicializuje se pole změn, největší zastoupení zde má nulová změna skóre proti predikci, s nižší četností jsou zde přidávány jemné odlišení od predikce, s mnohem nižší četností pak výraznější odchylka.
2. **Kontrola motivace:** detekce minulých zápasů, pokud zde nastává extrém (časté výhry, či časté prohry), je do pole přidáváno více ovlivnění výsledku.
3. **Kontrola důležitosti:** pokud se jedná o předem definované derby, je přidáváno více náhodných výsledků.
4. **Důležitost vzhledem k fázi sezóny:** pokud se nacházíme v poslední třetině sezóny, je přidán náhodný faktor, který simuluje nevyzpytatelný výsledek souboje týmů s velmi podobnými zisky v rámci boje o titul, evropské poháry a záchranu v soutěži, jsou přidány do pole další změny.

5. **Vytvoření původního pole:** inicializuje se pole změn, největší zastoupení zde má nulová změna skóre proti predikci.
6. **Výběr ovlivnění:** pro každý tým se z pole změn poté pomocí náhodného indexu vybírá konečné ovlivnění skóre dané výstupem z predikce.

6.7.4 Přepočítání mezi sezónami

Po konci každé sezóny následuje mezisezónní přepočítání. Úlohou tohoto přepočítání je vynulovat data vztahující se k právě skončené sezóně a provést všechny další odpovídající operace související s ukončením ročníku.

Přepočítání síly týmů

Po každé sezóně v rámci simulátoru dojde ke zhodnocení sezóny každého týmu. Vzhledem k tomu, že každá soutěž má různé rozpětí síly týmů, je potřeba úspěšnost sezóny každého týmu vypočítávat vzhledem k rámci tohoto rozpětí. V rámci přepočítání síly hrají roli tyto faktory:

- **Procentuální úspěšnost v sezóně:** procento udávající úspěšnost daného týmu vzhledem k maximálnímu možnému zisku bodů.
- **Současná síla týmu:** úspěšnost každého týmu se hodnotí dle jeho předpokladů, ty zde udává současná síla.
- **Průměrná síla soutěže:** jaké předpoklady tým má je potřeba rozhodovat i podle průměrné síly všech týmů v soutěži.
- **Rozdíl ve skóre:** kromě úspěšnosti pak hraje roli i rozdíl mezi vstřelenými a obdrženyými brankami.

Všechny tyto případy jsou pak zohledněny v samotném výpočtu nového hodnocení. Z důvodu zábavnosti a vývoje simulace je celý tento proces tvořen podobně jako výpočet ovlivnění skóre zábasu, vliv zde má i náhoda. Síla se ovlivňuje drobně, konkrétně maximálně o více/méně než dva body. V poli je povětšinou uložena změna o 0 bodů, ale výše zmíněné faktory pak do tohoto pole přidávají více změn.

Jako modelový příklad uveďme favorita na získání titulu, který ovšem dosáhne jen nadprůměrného zisku počtu bodů, který by odpovídal pouze nadprůměrnému týmu. Tento tým tedy má větší pravděpodobnost drobného snížení svého hodnocení, které by ho na škále hodnocení všech týmů částečně posunulo směrem od špičkových týmů k průměrným. Opakem může být adept na sestup, s jedním s nejhorších hodnocení. Tým dosáhne průměrného zisku bodu (tedy se umístí zhruba ze středu tabulky). Pak zde roste pravděpodobnost zvýšení hodnocení týmu a jeho zařazení spíše mezi podprůměrné týmy.

Postupy a sestupy

Každá soutěž má ve statické části dat poznamenána počet sestupujících a počet postupujících, pokud se jedná o soutěž nižší. Při sezónním přepočítání, po ukončení všech soutěží, pak probíhá tento přepočítání. Vzhledem k tomu, že v praktické části zatím není nastavena simulace na nižší soutěže, tak pro výpočet nově postupujících je využit podobný postup jako pro změnu skóre náhodou. Týmy jsou srovnány dle jejich síly, nejčtenější zastoupení pak mají

nejsilnější celky. Z tohoto pole je pak vybrán odpovídající počet týmů, které mají postoupit. Tyto týmy jsou pak vyměněny se sestupujícími celky přepsáním cizího klíče v dynamických datech na nový identifikátor soutěže.

Ukládání historie

Jak ukládat historii jednotlivých soutěží a týmů byl z hlediska úspory dat zásadní problém. Důležité bylo nejdřív vybrat ta nejpodstatnější data, která budou stále uložena. Vzhledem k délce simulace pak většina těchto dat narůstá. Pro současnou fázi simulátoru jsem se rozhodl ukládat čtyři části historických dat pro týmy a hráče:

- **Zápasy:** v databázi jsou ponechávány zápasy za posledních pět let, na základě nich se může vypočítávat forma, odpovídá dostupnosti původních historických výsledků.
- **Pořadí v soutěži:** pro každou sezónu se ukládá pořadí týmu v dané soutěži.
- **Šampioni soutěží:** separátně pak uložení mistrů soutěží, pro efektivnější dotazy, jejich výpis probíhá jinde než výpis pořadí jednotlivých týmů.
- **Celkový počet vstřelených branek hráče:** hráč má separátní počítadla ve své tabulce pro sezónní branky a celkové branky.

Z povahy těchto dat pak vyplývá, že zápasy a branky hráče jsou ukládány okamžitě, šampioni a pořadí v soutěži pak po každém dokončení sezóny.

6.7.5 Nulování uživatelské simulace

Pokud se uživatel rozhodne, že chce začít simulovat od původního počátečního bodu, je nutné správně vyčistit nasimulovaná data. Díky rozvržení na statická a simulační data je tato operace rychlá a nepříliš složitá na výpočet. Postup resetování simulace je:

1. **Nulování globálních informací:** nastavení sezóny na původní, nastavení prvního týdne.
2. **Nulování týmů:** nastavení na původní ohodnocení, původní formu, původní soutěž, nulování veškerých statistik a údajů.
3. **Nulování hráčů:** nulování statistik hráče.
4. **Nulování údajů o soutěžích:** nulování pořadí, historických šampionů, ročníku, týdne soutěže a statistik.

Jak jsem již výše zmiňoval, třeba původní síla nebo identifikátor soutěže jsou již uvedeny v tabulce. Aktualizace této hodnoty pak lze provést pomocí jednoduchého dotazu UPDATE jazyka SQL, aniž bychom potřebovali předávat jakékoliv další argumenty. Tento postup je pak opakován při více operacích, z toho tedy vyplývá rychlost provedení této akce.

6.7.6 Uživatelské a asynchronní akce na popředí

Pro asynchronní načítání a provádění požadavků bylo využito konceptu *AJAX* (asynchronous JavaScript and XML) [55]. Pro realizaci tohoto konceptu bylo využito knihovny

`nette.ajax.js`¹², která umožňuje propojit komponenty z Nette s klientskou částí aplikace. Při provádění asynchronních akcí se uživateli ukazuje loader, který trvá do ukončení asynchronní operace. Po skončení operace a znovunačtení dat tedy nedochází a obnově celé stránky, ale jen její části. Zároveň pak uživatel nemá možnost spouštět další blokující akce, má ale možnost prohlížení dalších údajů na stránce a jejich podstránkách.

6.7.7 Tvorba šablon a vizuální stránka aplikace

Pro tvorbu šablon a stylizaci jednotlivých komponent bylo využito klasických webových technologií jako jsou HTML, CSS a jQuery. Významnou roli v této oblasti pak má také framework Materialize.¹³ Tento framework pak poskytuje základní stylizaci pro řadu komponent a zároveň pak již předdefinované třídy. Tyto třídy pak obsahují nastavení například pro jednotlivé typy rozlišení. V rámci psaní HTML kódu je pak možné tedy rovnou skládat rozložení zároveň pro více velikost displejů [15].

6.8 Použité nástroje pro vývoj

Pro vývojové prostředí jsem v jednotlivých implementačních podčástech využil nástroje od firmy JetBrains.¹⁴ Konkrétně se jedná o vývojová prostředí PHPStorm (PHP), PyCharm (Python), WebStorm (JavaScript a Node.js) a DataGrip(databáze). Vývoj povětšinou probíhal v prohlížeči Safari a Chrome, testování webu pomocí nástrojů těchto prohlížečů. Pro běh jednotlivých částí bylo použito virtuální VPS server. Samotný vývoj pak probíhal na operačním systému macOS Mojave.

¹²<https://componette.com/vojtech-dobes/nette.ajax.js/>

¹³<https://materializecss.com>

¹⁴<https://www.jetbrains.com>

Kapitola 7

Testování

Kromě testování samotné predikce (zmíněno již v implementace modelu), pak probíhaly ještě dva druhy testování. Při tvorbě uživatelského rozhraní bylo již toto rozhraní testováno na svou použitelnost uživateli (část návrhu a testování proběhla v rámci projektu do školního předmětu UX1a), poté pak výsledek testován znovu vzhledem k vzhledu rozhraní v jednotlivých prohlížečích a škále zařízení. Po dokončení prototypu simulátoru pak skupina uživatelů znalých fotbalovému prostředí testovala průběh a reálnost samotného vývoje fotbalového světa. Funkční prototyp byl konzultován i s některými zájemci na konferenci Excel@FIT 2019.

7.1 Uživatelské testování GUI

Testování rozhraní probíhalo v rámci sezení se skupinou uživatelů a to v rámci dvou iterací. V každé iteraci pak byli jiní uživatelé. Změna uživatelů pak byla z toho důvodu, že část uživatelů jsou mi známí a měl jsem obavu z objektivitu zpětné vazby (ze zkušenosti, kdy uživatelé něchtějí více kritizovat věci, které by měly být opraveny). Důležité je zde říci, že testování probíhalo již na vytvořeném rozhraní, uživatel tedy měl možnost proklikávat samotný systém. Tento systém neměl implementovanou veškerou funkcionalitu. Zároveň chybějící data ze systému byla nahrazena aspoň daty vzorovými vzhledem k dané oblasti [36].

Testování probíhalo zadáním cílů uživateli a během každé instance s jedním uživatelem probíhala neustálá diskuse. To mi umožnilo lépe identifikovat chování uživatele na stránce a také lépe analyzovat silné a slabé stránky tohoto rozhraní. Testované byly následující (z obou iterací):

- Přihlaš se do systému.
- Najdi současnou tabulku ligy X.
- Zjistí výsledky z minulého kole soutěž X.
- Najdi tým X.
- Najdi pět hráčů z týmu X.
- Najdi jak tým X hrál v kole N.
- Najdi střelce zápasu mezi X a Y.

- Odsimuluj zápas mezi X a Y.
- Pro tým X najdi tři soupeře, s nimiž bude v následujících třech kolech hrát.
- Zjisti pozici hráče X.
- Zjisti nejlepších pět střelců v soutěži X.
- Vynuluj svou simulaci.
- Odhlaš se ze systému.

Dosažení cílů jsem s uživateli konzultovat a poznamenal si jejich případné připomínky. Někteří uživatelé dokonce sami navrhovali přidání některých částí do rozhraní.

Ještě před oběma iteracemi jsem se snažil sám analyzovat místa, která mi přišla problematická a nastítnit některé scénáře změn. Ty pak byly konzultovány i s uživateli pokud jsme na daný problém také narazili. Kromě oprav menších chyb z tohoto testování vyplynuly obecné úpravy rozhraní systému a to následující:

- **Navigace na stránce:** pokud se uživatelé dostávali na zanořenější podstránky, měli problém se zpětnou navigací, jediný způsob byl pomocí lokální navigace, přidal jsem tedy na všechny stránky příslušné *breadcrumbs*¹, ty obsahují seznam zanoření konkrétní stránky s odkazy na každou nadřazenou sekci.
- **Shlukování dat:** došlo k redukci počtu obrazovek. Některá data byla zbytečně na separátních obrazovkách, byla tedy přidána na stránku jako záložka s proklikem pomocí lokálních navigací.
- **Částečné odstranění scrollování:** obecně byl v rozhraní problém s tím, že elementy měly široké okraje a mezery. Bylo nutné tyto mezery zredukovat. Nyní je tak rozhraní více kompaktní, aby zde nebylo více scrollování, než je nutné.

7.2 Testování průběhu simulací

Po dokončení praktické části jsem menší skupině uživatelů, kteří mají povědomí o fotbalovém světě, zadal otestování tohoto prototypu. Testování bylo zaměřeno především na vývoj fotbalového světa, zajímala mě především variace výsledků a vývoje jednotlivých týmů. Úkolem bylo odsimulovat aspoň 10 (ale ideálně více) sezón a evidovat výsledky soutěží. Zároveň v rámci tohoto testování proběhla kontrola v rámci prohlížečů a evidence některých chybějících, či špatných dat (chybějící hráč, jeho jméno, či národnost).

7.2.1 Validace reálnosti

Samotné ověření validity vývoje systému bylo poměrně složité. Vzhledem k mnoha různým faktorům ovlivňující jednotlivé simulace nebylo ani možné exaktní nastavení očekávaných výsledků. Každý uživatel má jinou simulaci. Cílem tedy bylo ověření těchto faktorů:

¹<https://www.smashingmagazine.com/2009/03/breadcrumbs-in-web-design-examples-and-best-practices/>

1. **Plynulost změn:** každá liga by se měla vyvíjet plynule. Není příliš pravděpodobné aby špičkové týmy hned po sezóně či dvou nahradily týmy průměrné. Pokud nějaký tým upadá, mělo by se to také vlivem změn hodnocení a formy stát postupně a neměly by zde povětšinou nastat extrémy.
2. **Extrémní výsledky:** zde šlo o ovlivnění testování náhody, uživatelé reportují vyšší množství vysokých výsledků.
3. **Reálnost střelců:** v rámci systému je také generování střelců zápasu, toto bylo také ověřováno. Nejvíce branek by měli mít střelci a ofenzivní hráči. Například obránci by měli střílet branky spíše sporadicky.
4. **Reálnost postupujících:** bylo nutné také zkoušet, jací postupující postupují z nižších soutěží. Překvapivá jména by se měla také objevovat v menším rozsahu. Postupovat by hlavně měli favorité nižší soutěže, případně týmy, které sestoupily v minulé sezóně.

7.2.2 Výstup testování

Z výsledků testování plyne závěr takový, že s postupujícími sezónami se dokáží některé celky dostat do absolutní špičky. Zároveň pak občas některý z favoritů ztratí svoji pozici a spadne i do nižší soutěže. Tato evoluce ale trvá delší dobu a aspoň v blízké budoucnosti jsou výsledky reálné. Na obrázku 7.1 lze vidět výstup nejlepších čtyř týmů v každé z pěti simulovaných soutěží. Co se týče jednotlivých výsledků, ty vypadají poměrně reálně. Občas se (jako v reálném světě) vyskytne nečekaný a divoký výsledek, tato skutečnost zde ale není pravidlem. Tabulky střelců pak povětšinou ovládne střední útočník. Díky tomu, že zde hráči nejsou zatím nahrazováni dorostenci, tak v případě zlepšení nějakého týmu se pak jeho hlavní útočná hvězda posouvá ve statistikách výrazně výše. Nově postupující jsou pak většinou hlavní favorité nižších soutěží. V prvních sezónách pak tito nováčci soutěží mají problém v nejvyšší soutěži udržet. Tato skutečnost ale odpovídá realitě.

ID testera	Sezóna	Anglie	Německo	Španělsko	Francie	Itálie
1	2017/18	Man City, Man United, Chelsea, Liverpool	Bayern, Dortmund, Lipsko, Schalke	Real, Barcelona, Atlético Sevilla	PSG, Monako, Lyon, Marseille	Juventus, Inter, AC Milán, AS Řím
2	2017/18	Man United, Man City, Arsenal, Chelsea	Bayern, Dortmund, Koln, Lipsko	Barcelona, Real, Atlético, Valencia	PSG, Monako, Lyon, Nantes	Juventus, AS Řím, Lyon, AC Milán
3	2017/18	Tottenham, Man City, Liverpool, Man United	Bayern, Dortmund, Leverkusen, Schalke	Atlético, Real, Barcelona, Bilbao	PSG, Monako, Lyon, Marseille	Juventus, AS Řím, Inter, Lazio
1	2022/23	Arsenal, Man City, Man United, Chelsea	Lipsko, Bayern, Schalke, Dortmund,	Atlético, Real, Barcelona, Valencia	Monako, PSG, Nantes, Lyon	AC Milán, PSG, Sampdoria, Inter
2	2022/23	Man City, Chelsea, Man United, Tottenham	Bayern, Lipsko, Leverkusen, Koln	Real, Atlético, Barcelona, Betis	PSG, Monako, Marseille, Lille	Juventus, Inter, Neapol, Atalanta
3	2022/23	Man United, Arsenal, Chelsea, Liverpool	Dortmund, Lipsko, Bayern, Leverkusen	Barcelona, Sevilla, Atlético, Real	Monako, Lyon, PSG, Bordeaux	Inter, Juventus, Lazio, Neapol
1	2027/28	Arsenal, Chelsea, Burnley, Man United	Bayern, Lipsko, Schalke, Dortmund	Atlético, Real, Barcelona, Sevilla	Nantes, Monako, PSG, Lyon	Juventus, Sampdoria, AC Milán, AS Řím
2	2027/28	Man City, Arsenal, Tottenham, Watford	Lipsko, Bayern, Hoffenheim, Schalke	Real, Atlético, Valencia, Villareal	PSG, Marseille, Monako, Lyon	Inter, Juventus, AS Řím, Atalanta
3	2027/28	Arsenal, Chelsea, Liverpool, WBA	Bayern, Leverkusen, Frankfurt, Stuttgart	Sevilla, Real, Atlético, Barcelona	Monako, Lyon, PSG, Nantes	Juventus, AC Milán, Inter, Chievo
1	2032/33	Man United, Burnley, Chelsea, Arsenal	Lipsko, Wolfsburg, Frankfurt, Bayern	Barcelona, Atlético, San Sebastián, Real	PSG, Nantes, Monako, Guingamp	Juventus, Sampdoria, Inter, AC Milán
2	2032/33	Man City, Watford, Southampton, Man United	Bayern, Schalke, Lipsko, Leverkusen	Barcelona, Real, Sevilla, Bilbao	Monako, Marseille, Lyon, PSG	Inter, AC Milán, Atalanta, AS Řím
3	2032/33	Arsenal, Liverpool, WBA, Watford	Bayern, Frankfurt, Lipsko, Stuttgart	Sevilla, Barcelona, Real, Levante	Monako, Bordeaux, Lyon, PSG	Juventus, Chievo, Inter, AS Řím

Obrázek 7.1: Výsledky lig na příkladu simulace třech testerů

Celkově pak hodnocení průběhu simulací hodnotím jako úspěšné. Toto testování proběhne v budoucnu znovu. Stane se tak po zapracování aktualizace se stárnutím, přestupy a generováním nových hráčů. V průběhu tohoto testování se pak objevily nějaké chyby v systému. Tyto chyby jsem průběžně opravoval. Konkrétně šlo o chybu v přepočtu síly týmů po sezóně nebo nesrovnalosti mezi prohlížeči.

Kapitola 8

Budoucí vývoj

Samotnou diplomovou práci vývoj simulátoru nekončí. Již nyní mám naplánovanu řadu funkcionalit, rozšíření a vylepšení pro celý systém. Díky modulárnímu návrhu všech podčástí je možné efektivně přidávat nová rozšíření.

V současné době mám rozpracován princip Ligy Mistrů, Evropské Ligy a Evropského Superpoháru. Poté bude rozšířena možnost hratelných lig na většinu evropských zemí. Tato data jsou již v databázi dostupná, tudíž nebude složité zpracovat tyto systémy. Uživatel si také bude moci volit, které ligy si vybere (v současné době má zpracované ligy povinně). Následovat bude implementace mezinárodních soutěží. Poté je naplánována aktualizace dat na nadcházející sezónu 2019/2020.

Po těchto aktualizacích nastane čas zaměřit se na vylepšení hráčů. V současné době již mám dostupné databáze jmen pro největší země. Na základě těchto jmen se budou generovat dorostenci. Jejich schopnosti se budou počítat především dle hodnocení týmu a dle potenciálu týmu. Poměrně složitou operací pak bude smysluplné generování přestupů mezi týmy. To bude souviset i s vývojem samotných hráčů. Pokud se hráči bude v průměrném celku výrazně dařit a bude se zlepšovat, měl by nastat přestup do celku s vyšší úrovní kvality. Tento koncept mám zatím rozpracován, jeho implementace ovšem bude poměrně náročná.

Zpracováním hráčů pak bude hotov první cyklus a nastane přesun systému i na další platformy. Prioritou jsou platformy iOS a Android. Výrazné ulehčení migrace i na tyto platformy přináší univerzální serverová část aplikace, z níž bude možné efektivně získávat a pracovat s daty, stejně jako navázání na model predikce.

Následovat bude aktualizace pro možnost uživatele měnit data. Bude možné měnit téměř jakákoliv data nebo například přidávat nové hráče a týmy. To vše bude probíhat v rámci modifikované simulace. Uživatel si tuto část bude moci do svého účtu za drobný poplatek dokoupit. Tato funkcionalita by měla zajistit pokrytí nákladů (běh serveru, hosting a placené API) aplikace do budoucna.

Další plán pak zatím není jasný, po ukončení přestupového okna na konci letních měsíců pak každým rokem vyjde nová aktualizace, která bude přinášet aktualizaci soupisek a dat pro novou sezónu. Uživatel tak bude mít motivaci se opět k simulátoru vrátit.

Kapitola 9

Závěr

Cílem této práce bylo navrhnout, implementovat a otestovat webový simulátor fotbalových lig a turnajů. Proběhla analýza pravidel fotbalového světa, existujících řešení a studium technik strojového učení. Dále byl vytvořen návrh všech podčástí, klíčovou oblastí návrhu byly především vstupní atributy predikce výsledku. Další důležitou oblastí bylo nastavení validity predikce. Takto bylo již od návrhové fáze jasné, jaké úspěšnosti odhadu reálnosti a správnosti výsledků chci v rámci práce dosáhnout. To mi pomohlo dále ladit jak systém, tak model. Výsledná predikce pak byla testována mimo jiné i na reálných fotbalových zápasech. V několika iteracích pak srovnána s tipováním sázkové kanceláře.

Na základě návrhu se podařilo vytvořit simulátor, který využívá predikci výsledků dle reálných historických fotbalových dat a to použitím modelu s využitím polynomiální regrese. Tento systém se pak dále člení na tři podčásti – samotnou aplikaci, API pro práci s daty a API pro propojení systému s modelem predikce. Tento simulátor je použitelný na široké škále zařízení. Všechny části běží odděleně na virtuálním VPS serveru a komunikují mezi sebou.

Systém byl testován již v průběhu návrhu a raného vývoje (testování uživatelského rozhraní), pak také v rámci trénování modelu (testování validity predikce). Závěrečně pak proběhlo testování skupinou domluvených uživatelů. V této části testování jsem se zaměřil na reálnost a rychlost vývoje jednotlivých lig, reálnost generování střelců a reálnost samotných výsledků po ovlivnění predikce náhodou. Prototyp byl také prezentován na konferenci Excel@FIT 2019.

Systém byl navržen modulárně a nic nebrání jeho budoucímu rozšiřování. Na následující měsíce je pak plán již nastaven. Bude se především rozšiřovat funkcionalita a simulátor bude implementován i jako aplikace na mobilní platformy. Simulátor také bude každý rok aktualizován vzhledem k datům nové nadcházející sezóny po uzavření přestupových oken.

Literatura

- [1] FIFA official website. [Online; navštíveno 22.12.2018].
URL <http://fifa.com>
- [2] Bookmaker a kurzy. [Online; navštíveno 2.4.2019].
URL <http://www.jaknasazky.cz/clanky/bookmaker-kurzy>
- [3] Bundesliga. [Online; navštíveno 20.2.2019].
URL <https://www.dfb.de/en/leagues/bundesliga/history/>
- [4] Dependency Injection. [Online; navštíveno 4.5.2019].
URL <https://doc.nette.org/en/2.4/dependency-injection>
- [5] Dropping odds strategy. [Online; navštíveno 3.4.2019].
URL <https://www.bookmakers.bet/8167/dropping-odds/>
- [6] Europa League - winners and history. [Online; navštíveno 19.2.2019].
URL <https://www.footballhistory.org/tournament/europa-league.html>
- [7] European Qualifiers for UEFA EURO 2020: how it works. [Online; navštíveno 5.5.2019].
URL <https://www.uefa.com/uefaeuro-2020/news/newsid=2502936.html>
- [8] Football - Players their Positions. [Online; navštíveno 12.3.2019].
URL https://www.tutorialspoint.com/football/football_players_and_positions.htm
- [9] Football Player Transfers Explained. [Online; navštíveno 10.3.2019].
URL <https://www.football-stadiums.co.uk/articles/football-player-transfers-explained/>
- [10] French Ligue 1 - facts, stats and hisotry. [Online; navštíveno 19.2.2019].
URL <https://www.footballhistory.org/league/ligue-1.html>
- [11] Google Assistant and Artificial Intelligence – The future of simulated communication. [Online; navštíveno 12.5.2019].
URL <https://campaignsoftheworld.com/digital/google-assistant-and-artificial-intelligence/>
- [12] Host Country Selection. [Online; navštíveno 14.4.2019].
URL <https://www.topendsports.com/events/worldcupsoccer/hosts/selection.htm>

- [13] How Does The Spanish Football League System Work? [Online; navštíveno 11.2.2019].
URL <https://www.erttheo.com/blog/en/spanish-football-league-system/>
- [14] Italian Serie A - facts, stats and history. [Online; navštíveno 19.2.2019].
URL <https://www.footballhistory.org/league/serie-a.html>
- [15] Materialize Grid. [Online; navštíveno 1.5.2019].
URL <https://materializecss.com/grid.html>
- [16] Overfitting in Machine Learning: What It Is and How to Prevent It. [Online; navštíveno 16.3.2019].
URL <https://elitedatascience.com/overfitting-in-machine-learning>
- [17] Polynomial Regression. [Online; navštíveno 20.4.2019].
URL <https://towardsdatascience.com/polynomial-regression-bbe8b9d97491>
- [18] Polynomial Regression : Machine Learning in Python. [Online; navštíveno 27.4.2019].
URL <https://www.techeos.com/posts/polynomial-regression-machine-learning/>
- [19] Polynomial regression: extending linear models with basis functions. [Online; navštíveno 12.3.2019].
URL https://scikit-learn.org/stable/modules/linear_model.html#polynomial-regression-extending-linear-models-with-basis-functions1
- [20] Python object serialization. [Online; navštíveno 24.4.2019].
URL <https://docs.python.org/3/library/pickle.html>
- [21] Regulations of the Europa League (2018-21 cycle). [Online; navštíveno 1.3.2019].
URL https://www.uefa.com/MultimediaFiles/Download/Regulations/uefaorg/Regulations/02/55/82/82/2558282_DOWNLOAD.pdf
- [22] Regulations of the UEFA European Football Championship. [Online; navštíveno 10.4.2019].
URL https://www.uefa.com/MultimediaFiles/Download/Regulations/uefaorg/Regulations/02/54/36/05/2543605_DOWNLOAD.pdf
- [23] UEFA Nations League: all you need to know. [Online; navštíveno 3.5.2019].
URL <https://www.uefa.com/uefanationsleague/news/newsid=2079553.html>
- [24] What Is Machine Learning? [Online; navštíveno 1.4.2019].
URL <https://www.mathworks.com/discovery/machine-learning.html>
- [25] World Cup. [Online; navštíveno 4.5.2019].
URL <https://www.britannica.com/sports/World-Cup-football>
- [26] Úvod do Node.js. [Online; navštíveno 10.5.2019].
URL <https://www.itnetwork.cz/javascript/nodejs/uvod-do-nodejs>
- [27] Agarwal, A.: Polynomial regression. [Online; navštíveno 2.4.2019].
URL <https://towardsdatascience.com/polynomial-regression-bbe8b9d97491>

- [28] Bezděk, J.: *Aplikace metod strojového učení na predikci výsledků fotbalových zápasů*. 2018.
- [29] Bhatia, R.: When not to use Neural Networks. [Online; navštíveno 13.4.2019].
URL <https://medium.com/datadriveninvestor/when-not-to-use-neural-networks-89fb50622429>
- [30] Blažej, A.: *Vliv investic na nákup hráčů na sportovní výsledky u vybraných klubů Premier League*. 2018.
- [31] Brownlee, J.: Overfitting and Underfitting With Machine Learning Algorithms. [Online; navštíveno 14.3.2019].
URL <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>
- [32] Coursera: Online Courses From Top Universities: Machine Learning. [Online; navštíveno 2.1.2019].
URL <https://www.coursera.org/learn/machine-learning>
- [33] Donges, N.: Pros and Cons of Neural Networks. [Online; navštíveno 4.4.2019].
URL <https://towardsdatascience.com/hype-disadvantages-of-neural-networks-6af04904ba5b>
- [34] Gerald, P.: Guide to the Leagues and Cups of English Football. [Online; navštíveno 11.2.2019].
URL <https://englishsoccerguide.com/guide-leagues-cups-english-football/>
- [35] Goulet, J. A.: Regression Analysis. [Online; navštíveno 17.4.2019].
URL https://users.wpi.edu/~goulet/MME523/my_dm_regression.htm
- [36] Herout, A.: User Design and Experience of User and Interfaces Services. [Online; navštíveno 10.4.2019].
URL https://wis.fit.vutbr.cz/FIT/st/cwk.php?title=Main_Page&src=UXI-05-Testing.pdf&ns=UXIa&action=download&cache=no&csid=674893&id=130519
- [37] Honzík, P.: Strojové učení. *Elektronická skripta VUT Brno*, 2006.
- [38] Johnson, J.: *Designing with the mind in mind: simple guide to understanding user interface design guidelines*. Elsevier, 2013.
- [39] Kassies, B.: UEFA Coefficients calculation method. [Online; navštíveno 22.12.2018].
URL <https://kassiesa.home.xs4all.nl/bert/uefa/calc.html>
- [40] Ústav formální a aplikované lingvistiky, U. K.: Strojové učení: úvod, regrese. [Online; navštíveno 11.5.2019].
URL https://ufal.mff.cuni.cz/~vernerova/2016/docs/prednaska_04_strojove_uceni.pdf
- [41] Malý, M.: REST: architektura pro webové API. [Online; navštíveno 4.5.2019].
URL <https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>
- [42] Marcotte, E.: *Responsive web design: A book apart n 4*. Editions Eyrolles, 2017.

- [43] Mel, L.: Polynomial linear regression. [Online; navštíveno 2.4.2019].
URL <https://medium.com/data-py-blog/polynomial-linear-regression-1f8117f6de2e>
- [44] Milde, D.: Polynomická regrese. [Online; navštíveno 9.5.2019].
URL <http://ach.upol.cz/user-files/intranet/13-pokrocilaregrese-2017-1505723153.pdf>
- [45] Montti, R.: What is Google's Neural Matching? [Online; navštíveno 13.5.2019].
URL <https://www.searchenginejournal.com/google-neural-matching/271125/#close>
- [46] Novák, M.: *Umělé neuronové sítě: teorie a aplikace*. CH Beck, 1998.
- [47] Pal, S. K.; Mitra, S.: *Multilayer Perceptron, Fuzzy Sets, Classification*. 1992.
- [48] Pant, A.: Introduction to Linear Regression and Polynomial Regression. [Online; navštíveno 16.3.2019].
URL <https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb>
- [49] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; et al.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.
- [50] Ruizendaal, R.: Deep Learning 3: More on CNNs Handling Overfitting. [Online; navštíveno 16.3.2019].
URL <https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d>
- [51] Schánělec, J.: *Liga mistrů - historie a současnost*. 2012.
- [52] Selinger, T.: *Sportovní talent a diagnostika psychických vlastností ve fotbale*. 2018.
- [53] Sentance, R.: 15 examples of artificial intelligence in marketing. [Online; navštíveno 3.3.2019].
URL <https://econsultancy.com/15-examples-of-artificial-intelligence-in-marketing/>
- [54] Sokol, V.: *Predikce výsledků sportovních utkání s použitím neuronových sítí*. 2013.
- [55] Spínola, D.: JavaScript—from callbacks to async/await. [Online; navštíveno 16.5.2019].
URL <https://medium.freecodecamp.org/javascript-from-callbacks-to-async-await-1cc090ddad99>
- [56] Volná, E.: *Neuronové sítě 1. Ostrava: Ostravská univerzita v Ostravě. Vydání: druhé, 2008.*
- [57] Vomlelová, M.: Strojové učení. [Online; navštíveno 4.1.2019].
URL <http://kti.mff.cuni.cz/~marta/uvod.pdf>
- [58] Vysloužilová, L.: Předzpracování dat. [Online; navštíveno 7.5.2019].
URL https://cw.fel.cvut.cz/old/_media/courses/xp33ppd/ppd2014-03-predzpracovani_dat.pdf