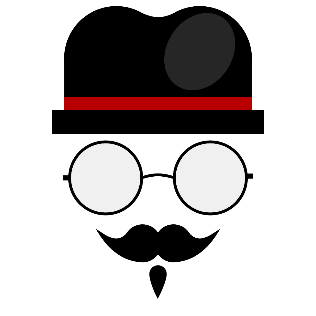
**Ročníková práce**

**Programování – 4.E**



**Blackjack**

**Autor:** Filip Dávidík

**Škola:** Gymnázium, Praha 6, Arabská 14

**Kraj:** Praha

**Konzultant:** Mgr. Jan Lána

**Praha 2020/2021**

**Prohlášení**

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praha dne 2. 3. 2020 ………………………………………………………………………….

**Anotace**

Naprogramuji známou karetní hru Black Jack, ke které dále vytvořím kalkulátor, který bude brát v potaz všechny aspekty(pravidla hry, zbývající karty, možnosti tahu hráče a krupiéra) ovlivňující výsledek hry a bude je zobrazovat uživateli. Hlavní částí projektu je tedy onen kalkulátor, který bude počítat pravděpodobnost výhry v daném bodě hry a zobrazovat hráči všechny jeho možnosti. Jedná se tedy o projekt, velmi spjatý s mojí oblíbenou částí matematiky, kombinatoriky.

Cílem této aplikace by mělo pomoci lépe pochopit jak, tato oblíbená hazardní hra funguje, vytvořit z vlastní hlavy, co nejlepší algoritmy pro kalkulátor a zda-li na Black Jacku opravdu lze, z dlouhodobého hlediska, vydělat.

**Obsah**

# Úvod

Blackjack je v dnešní době druhou nejoblíbenější karetní hazardní hrou na světe, první je poker. Díky této hře přišlo nemálo kasin k nemalému zisku, a to kvůli tomu, že jako u většiny hazardních her v kasinech, má kasino mnohem větší šanci na výhru sázky než hráč (u Blackjacku se jedná o cca. 60% výher kasino 40% výher hráči, záleží na pravidlech). Toto pravidlo sice platí i pro Blackjack, avšak pouze pokud je hráč někdo, kdo hru neumí hrát pomocí strategie zvané „Card counting“, neboli počítání karet. Člověk, co umí a zná všechny metody „Card countingu“ se dokáže dostat přes onu hranici, kde má kasino větší šanci na výhru než hráč, a učinit si tak nemalý profit. Rád bych v této práci tedy probral Blackjack jako hru, metody užívané k vyšší šanci na výhru, a nakonec můj program, který slouží k procvičování hráče v daných metodách.

.

# Blackjack

## Pravidla hry

Jako první je důležité zmínit, že po termínem Blackjack si můžeme představit mnoho různých verzí oné hry, avšak v této práci se budeme zaobírat pouze americkým, jinak referovaným jako klasickým, Blackjackem, který jak z názvu vypovídá potkáte v kasinech nejčastěji. V klasickém Blackjacku se užívají takzvané francouzské hrací karty v angličtině „French-suited playing cards“, jedná se o sadu karet skládajíc se ze 4 barev (srdce, piky, kříže, listy) a 13 ti různých karet (2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A) pro každou barvu (celkově tedy 52 karet pro jeden balíček karet). Každý hráč na začátku hry (kola) obdrží, po vložení sázky, 2 karty, přičemž krupiér si rozdá pouze jednu kartu. Po rozdání karet jsou po sobě na tahu hráči, kteří mají, podle situace několik možností, jak hrát dál. Hráč má vždy k dispozici buďto líznout si kartu „Hit“ nebo předat tah dalšímu na řadě „Stand“. Cílem hry je se dostat součtem hodnot karet v tvé ruce, co nejblíže hodnotě 21 (J, Q, K mají vždy hodnotu 10, zatímco A má hodnotu buďto 11 nebo 1, podle toho, co je pro hráče výhodnější). Hráč má taky možnost provést tah „Double“, který hráči zdvojnásobí sázku a zároveň lízne jednu kartu, avšak po zahrání tahu „Double“ musí hráč přenechat tah dalšímu hráči na řadě. Pokud jsou obě karty rozdané hráči stejné hodnoty, může hráč provést tah „Split“, který rozdělí hráčovu hrací ruku na dvě ruky, se kterými pak následně hraje hráč s každou zvlášť. Nutno dodat, že tah „Split“ také automaticky přidá duplikát počáteční sázky hráče i jeho druhé hrací ruce, a pro každou ruku lze zahrát tah „Double“, takže je potencionální výhra dvakrát větší, než by mohla být jen s jednou hrací rukou.

## Historie

Historie této karetní hry začíná někdy kolem roku 1700 ve francouzských kasinech, v té době tato karetní hra ale nesla jiné jméno (též i trocha jiná pravidla), a to „Vingt-et-Un“, což můžeme do češtiny přeložit jako „Dvacet Jedna“. Věří se, že tato hra vznikla na základu karetní hry „Chemin der Fer“, španělsky „Baccarat“, česky „Železnice“, která v té době byla jednou z velmi populárních hazardních her, jak v kasinech, tak i v menších krčmách a v podobných podnicích. Karetní hra „Vingt-et-Un“ se ve své době stala tak oblíbenou, že se hrála dokonce i na francouzském královském dvoře během vlády krále Ludvíka XV. Během 18. století vzniklo mnoho verzí dnes nazývaného Blackjacku, například v Severní Americe mohl použít Double krupiér, nýbrž hráči. Zatímco v 19. století popularita hry „21“(Vingt-et-Un) v Evropě nerostla, v Americe se jí dostávalo nevídaného úspěchu, a to zejména v kasinech amerického přístavního města New Orleans. Bylo to ve 20. století v Nevadě, když se poprvé místo tradičního názvu naší karetní hry „21“ objevil v kasinech název Blackjack. Nejednalo se pouze o změnu jména, s novým názvem hry přišla i nová pravidla, která šla v souladu s tehdy nově vydanou nevadskou legislativou o fungování hazardních her.

## Současnost

# Vývoj aplikací

Cílem našeho projektu bylo vytvořit ucelený nástroj pro investory, který by jim pomohl usoudit, zda je dobrý čas na investování do Bitcoinů.

Rozhodli jsme naprogramovat server, který bude shromažďovat, zpracovávat a ukládat data. A pak je posílat na požadavek. A dále jsme naprogramovali i webovou stránku, která bude data stahovat ze serveru a zobrazovat do přehledných grafů a listů.

## Výběr technologií

Většina práci jsme dělali v Javascriptu. Jedná se o objektově orientovaný programovací jazyk, který nabízí vše k vývoji interaktivních webových stránek.

Nabízí 3 frameworky k vytváření UI: Vue.js, Angular.js a React.js. My jsme si vybrali **React.js**, protože je nejlépe aktualizován.

U serveru jsem se rozhodli také pro Javascript. Pomoc **Nodej.js** (napsané v [C](https://www.google.com/search?client=firefox-b-d&sa=X&biw=1536&bih=750&sxsrf=ALeKk01R-uKbr3bN8C0sv8Yr23zX6GFuTA:1583078918739&q=C+programming+language&stick=H4sIAAAAAAAAAOPgE-LSz9U3SEqqSLOwVOIAsQ1LzJK0lDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrcqLMktKUvMUMvMWsYo5KxQU5acXJebmZualK-Qk5qWXJqan7mBlBACgJka3WAAAAA&ved=2ahUKEwjK9ejE1PnnAhUR-aQKHZfBAwQQmxMoATAbegQIERAD), [C++](https://www.google.com/search?client=firefox-b-d&sa=X&biw=1536&bih=750&sxsrf=ALeKk01R-uKbr3bN8C0sv8Yr23zX6GFuTA:1583078918739&q=C%2B%2B&stick=H4sIAAAAAAAAAOPgE-LSz9U3SEqqSLOwVOIAsbPSC9O1lDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrcqLMktKUvMUMvMWsTI7a2vvYGUEALM14BxFAAAA&ved=2ahUKEwjK9ejE1PnnAhUR-aQKHZfBAwQQmxMoAjAbegQIERAE), [JavaScript](https://www.google.com/search?client=firefox-b-d&sa=X&biw=1536&bih=750&sxsrf=ALeKk01R-uKbr3bN8C0sv8Yr23zX6GFuTA:1583078918739&q=JavaScript&stick=H4sIAAAAAAAAAOPgE-LSz9U3SEqqSLOwVOIAsY0KLM21lDLKrfST83NyUpNLMvPz9Ivz00rKE4tSrcqLMktKUvMUMvMWsXJ5JZYlBicXZRaU7GBlBACgOdHfTAAAAA&ved=2ahUKEwjK9ejE1PnnAhUR-aQKHZfBAwQQmxMoAzAbegQIERAF)**)** (6), což je prostředí umožňující spouštět JavaScript kód mimo webový prohlížeč. To nám umožnilo psát frontend i backend v Javasciptu. Pro jednodušší obsluhování a fungování serveru jsme použili framework **Express.js** a pro ukládání NoSQL databázi **MongoDB.**

Tento výběr technologií je často používaný a kvůli jednotnému programovacím jazyku velice pohodlný. Obvykle se nazývá zkratkou MERN Stack (první písmeno z každé technologie).

## Backend

Backend má krom stahování a obsluhování klientů zpracovávací funkci. Aby tyto data nemusel zpracovávat každý klient zvlášť, stačí aby tyto data zpracoval jednou právě server a výsledek posílal. Tímto šetříme výkon.

Efektivní optimalizací posílání dat se stalo jejich předchozí stringování. Data mají totiž v databází formát .json. Před každým požadavkem data vždy zkomprimujeme (dochází k ušetření místa v řádech desítek %) z .json do prostého textu. Až data dorazí, klient si sám zase naparsuje do pohodlného .json. Tímto postupem šetříme požadavky na rychlost sítě.

### Historické kurzy

Mít přístup k aktuálním a historickým kurzům zcela zdarma se ukázalo jako komplikovaný cíl.

Mnohé API[[1]](#footnote-1) zdarma sice poskytují data, ale omezují je na nízkou aktualizaci dat a vysokou odezvu. Dále bylo potřeba zjistit kvalitu dat z jednotlivých API, některá totiž obsahovala časová okna úplně bez dat.

Po otestovaní těch nejpopulárnějších se ukázalo, že každá API má nějaké nedostatky a nejlepší bude použití nějaké kombinace. První z nich je Coinbase Pro. Poskytuje rychle se aktualizující data v různých intervalech a částech. Avšak v některých měsících data vykazovaly ztráty. Tuto nevýhodu jsme vyřešili s kombinací BitcoinAverage. Jejich data se sice aktualizují každé asi 3 dny, alespoň v jejich datech nejsou žádné ztráty.

U obou bylo potřeba ošetřit omezení požadavků. U Coinbase Pro je to maximální počet kurzů na 300. Například denní kurzy za celý rok jsme museli rozdělit na 2 požadavky po 300 a 65. To se provádí na pozadí zcela automatizovaně díky aplikované a chytré funkci.

Naše API je teda kombinací dvou zdrojů, které si kompenzují své nedostatky.

### Google Trends

API jsme také vylepšili o zajímavou funkci od Googlu, která poskytuje zájem lidí o klíčové slovo (v našem případě „bitcoin“). Data neobsahují přesná čísla, ale poměry v daném úseku, což pro zobrazení v grafu není žádný problém; postačí jako ukazatel zájmu.

Data se stejně jako kurzy aktualizuje průběžně.



Obrázek 1.- korelace ceny a zájmu

Na obrázku 1. je výrazně vidět jak cena Bitcoinu v průběhu 1 měsíce koreluje se zájmem lidí. Tento zajímavý ukazatel určitě stojí za zaobírání jak pro investora, tak i pro analytiky v oboru. Za zmínku stojí také říct, že zájem v Google Trends, lze výkonnými počítači lehce zmanipulovat.

### Indikátory

Jedná se o ukazatele, které se chovají jako trend. Pokud jde cena nižší než tento indikátor, znamená to, že je cena pod dlouhodobým trendem. Obecně pokud je cena blíže k indikátoru, dá se po překřížení očekávat růst nebo pokles; v závislosti na směru křížení.

Jednoduchý klouzavý průměr je specifický tím, že přikládá všem hodnotám stejnou důležitost, což se dá vnímat jako nevýhoda. Pokud je poslední hodnota extrémní, může se po přidání další hodnoty průměr rapidně změnit opačným směrem, což nemusí znamenat, že nastal prudký vzrůst či spád v ceně. Počítáme jej tedy (cena n + cena n-1 + cena n-2)/n.

U váženého klouzavého průměru přiřazujeme každé hodnotě jinou váhu sestupně od hodnoty bližší momentální hodnotě.

Exponenciální klouzavý průměr dává vyšší váhu posledním hodnotám. Počítáme jej tedy: (EMA n-1) + [x\*(cena n-EMA n-1)], kdy x=2/(n+1).

Triangulární klouzavý průměr dává nejvyšší váhu prostřední/prostředním hodnotě/hodnotám a směrem ke krajům váha klesá. Počítáme jej tedy: (2\*cena n-3 + cena n-2 + cena n-1 + 2\*cena n)/n.

Bollingerova pásma používají klouzavé průměry tak, že vykreslují čáry v určité vzdálenost nad a pod klouzavý průměr. Bollingerova pásma tvoří tři křivky. Středová zelená křivka zobrazuje klouzavý průměr, Kolem středu je vytvořena obálka proměnlivé šířky (modrá). Obálku tvoří násobek r směrodatných odchylek n posledních zavíracích kurzů.

### Strojové učení

Strojové učení bylo do nedávna záležitostí především v jazyce python. Díky velkému množství peněz investovaných do vývoje tohoto oboru, ale došlo k rozšíření v mnoha směrech (7).

Jedním z nich je TensorFlow.js, knihovna disponující neurální sítí. Neurální sít potřebuje co největší množství dát k učení. V našem případě s databází to není však problém.

Formát vstupu do sítě jsme museli rekonstruovat do formátu, který je vhodný a nejvíce efektivní. Normalizování dat pomocí škálování se ukázalo jako špatný nápad, protože ve vybraném časovém úseku se cena často pohybovala v rozmezí maximálně jedné stovky dolarů, což u aktuálního kurzu (ke dni 1. 3. 2020 téměř $8000) nemá dostatečnou páku na penalizaci (learning rate) neurální sítě. Nejnižší hodnoty loss[[2]](#footnote-2) při 400 epochs[[3]](#footnote-3) s přibližně 500 vstupy jsme dosáhli přeformátováním dat na sekvence 0s a 1s. Nastaveno na 0 pokud ve srovnání s předchozím dnem cena klesla, a 1 pokud vzrostla. Pro výstupy jsme nastavili hodnotu následující dne. Pro velikost okna 5 (nakonec jsme zvolili velikost 20) by vstup mohl vypadat například takto: [0,0,1,0,1] a pro výsledek[1]. V „překladu“ to znamená, že první dva dny cena klesala, 3. vzrostla, 4. klesla, 5. vzrostla a 6. den, kterým je výsledek, cena vzrostla. Neurální síť tyto vstupy projede a pokusí se zjistit nějakou spojitost a pravidelnost mezi vstupy a výsledky. Nezbytnou součástí je míchání dát a vybalancování dat, pokud totiž budou mít trénovací data převážně výsledky vzrůstu, tak i predikce vytrénované sítě budou spíše sklouzávat k růstu. Dále bylo potřeba data otestovat, na reálných datech; na 3 pokusech vždy o velikosti 50 testovacích dat jsme dosáhli úspěšnosti 55%, 46%, 56% (v průměru 52,33%), což jsou velice uspokojivé výsledky.

## Frontend

Je něco, čemu jsme kladli veliký důraz; stránka vypadá profesionálně a nabízí příjemný vzhled, logickou navigaci strukturu, interaktivitu s uživatelem a další.

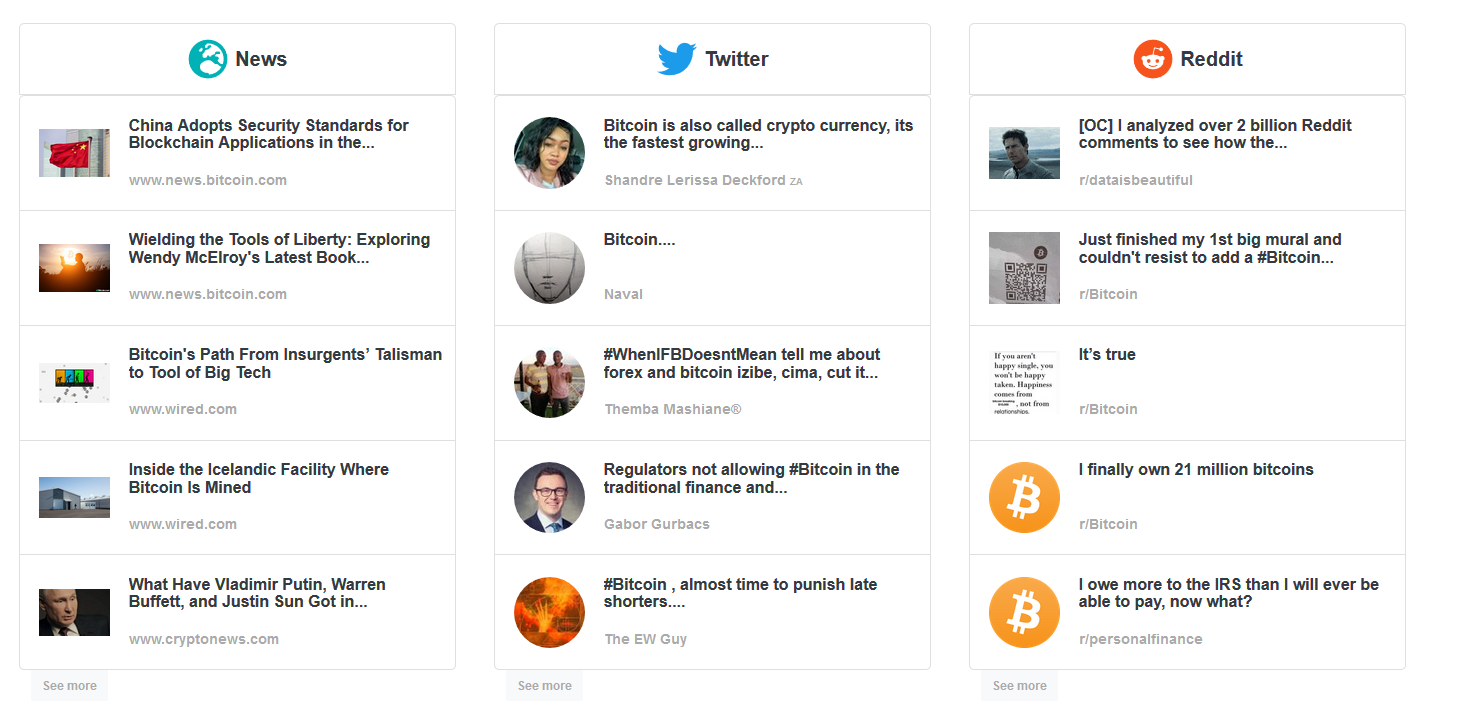
## Použité technologie

Stránka je naprogramovaná pomocí javascriptového frameworku React.js. Dále jsme použili knihovnu Redux, který se stará o uložená data (states) a aktualizuje jen ty komponenty, který to opravdu potřebují. Například cena Bitcoinu se aktualizuje každou minutu, avšak to neznamená, že se musí aktualizovat něco jiného; takové vypracování ve webové stránce, která neustále něco aktualizuje serveru, nesmělo rozhodně chybět.

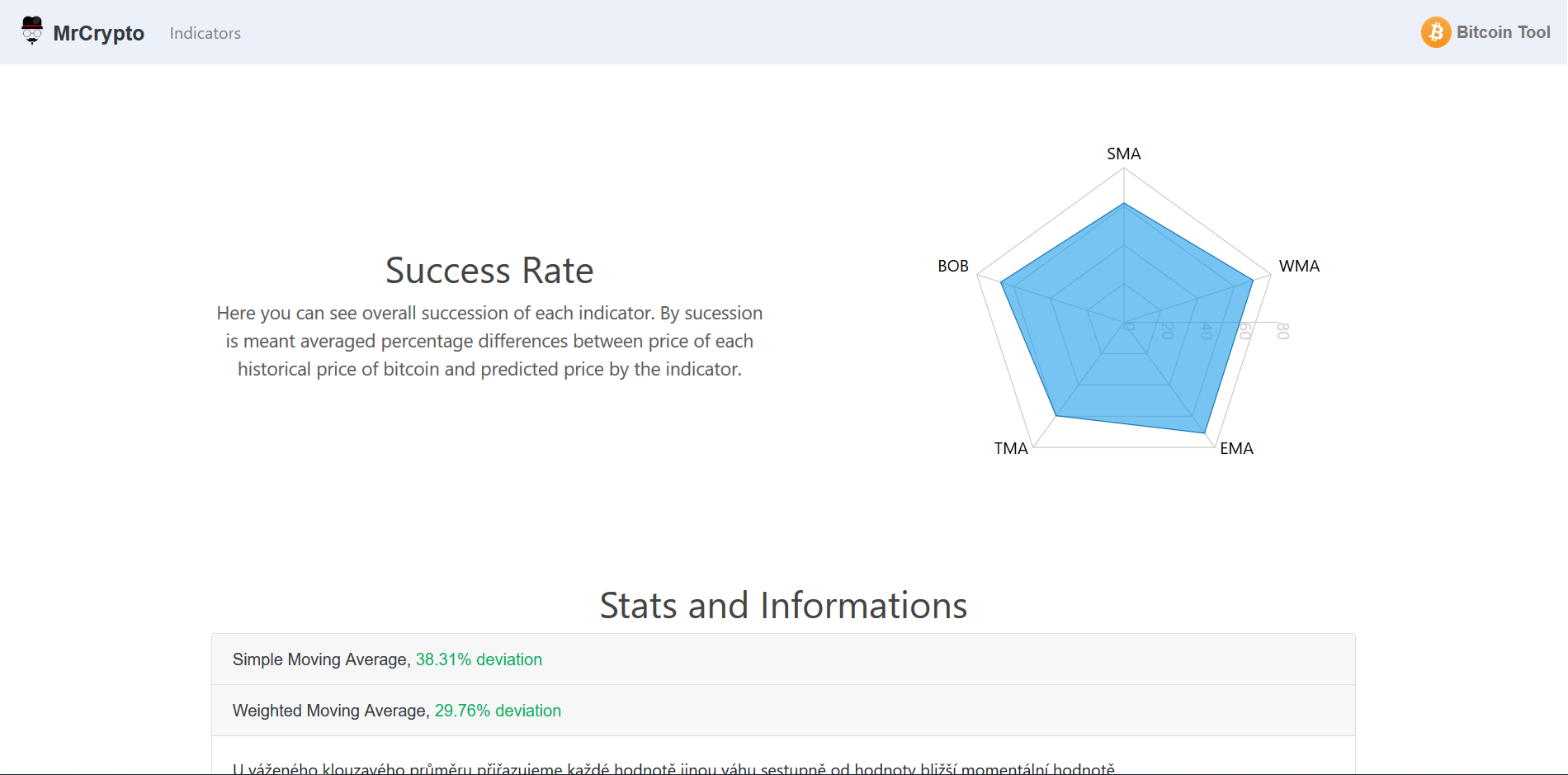
## Výstřižky z webové stránky



Obrázek 2. - úvodní pohled na stránku



Obrázek 3. - struktura zobrazení článků o Bitcoinu



Obrázek 4. - přehled o indikátorech a jejich úspěšností

# Závěr

Tato ročníková práce má mnoho zajímavých podkategorií, které skrývají vysoký potenciál. A i když jsme zadání práce splnili, tak je tu prostor pro další vylepšení a kroky k lepšímu pochopení, jak kryptoměny fungují a reagují různé podněty.

Budoucnost aplikace vidíme spíše ve vzdělávacím směru. Určitě muže být aplikace vnímaná jako realtimový simulátor a být obohacena o kupování a prodávání. A tím sloužit začínajícím investorům jako pískoviště pro jejich budoucí reálné investice.

# Bibliografie

1. **[Online] https://cs.wikipedia.org/wiki/Bitcoin.**

**2. [Online] https://www.investicniweb.cz/pet-faktoru-ovlivnujicich-cenu-bitcoinu/.**

**3. [Online] https://www.buybitcoinworldwide.com/how-many-bitcoins-are-there/.**

**4. [Online] https://blockgeeks.com/guides/bitcoin-halving/.**

**5. [Online] https://connect.zive.cz/clanky/kryptomenove-forky-jak-a-proc-dochazi-k-rozdeleni-kryptomen/sc-320-a-192682/default.aspx.**

**6. [Online] https://en.wikipedia.org/wiki/Node.js.**

**7. [Online] https://www.techworld.com/picture-gallery/data/tech-giants-investing-in-artificial-intelligence-3629737/.**

Seznam obrázků

[Obrázek 1.- korelace ceny a zájmu 7](#_Toc34027468)

[Obrázek 2. - úvodní pohled na stránku 9](#_Toc34027469)

[Obrázek 3. - struktura zobrazení článků o Bitcoinu 10](#_Toc34027470)

[Obrázek 4. - přehled o indikátorech a jejich úspěšností 10](#_Toc34027471)

1. API: rozhraní pro programování aplikací. [↑](#footnote-ref-1)
2. Loss je hodnota na určení přesnosti vytrénované neurální sítě. Čím nižší, tím přesnější. [↑](#footnote-ref-2)
3. Epoch je počet, kolikrát jedny data použijeme pro vytrénování neurální sítě. Pokud máme 10 vstupů a epochs nastavené na 30, tak neurální sítí projde 300 dat. [↑](#footnote-ref-3)