KIV/VSS

Skrytý markovský model pro predikci cen akcií

Martin Forejt - A20N0079P
mforejt@students.zcu.cz

22. 1. 2021

Obsah

1 Zadání

Cílem je podívat se na stávající literaturu a vyprat si pro vás zajímavý simulační problém, který zkusíte reimplementovat. V minulých letech se objevila řada věcí - celulární automat pro modelování šíření požáru, vzniku mraků nebo pohybu kapaliny v terénu, predikce cen akcií postavené na markovských modelech, simulace různého chování hmyzu a podobně. Není nutné přijít s vlastním, originálním řešením, může jít (a očekává se že půjde) o replikační studii, moc se jich nedělá.

Součástí zadání pak bude přehled state-of-the-art - co v dané oblasti už existuje, jaké nástroje je možné využít a podobně.

1.1 Skrytý markovský model pro predikci cen akcií

Cílem semestrální práce by měla být analýza odborné literatury na téma predikce cen akcií za pomoci skrytého markovského modelu (dále HMM) a jeho následná implementace.

Práce bude vycházet ze článku Hidden Markov Model for Stock Trading (Nguyen N., 2018) [1], který představuje aplikaci HMM pro obchodování s akciemi (jako příklad je použit index S&P 500) na základě jejich predikce. Autor začíná použitím 4 kritérií pro odhad predikční chyby, aby určil optimální počet stavů pro HMM, vybraný čtyřstavový HMM implementovaný dle [2] je použit k predikci měsíčních uzavíracích cen indexu S&P 500.

Výsledkem práce by měla implementace modelu v podobě desktopové aplikace (v pythonu), která dle historických dat (ať už poskytnuta uživatelem, či získána automaticky např. z finance.yahoo.com) natrénuje model a následně zobrazí v grafu jak trénovací data, tak predikovaná data.

V první části by měl mít implementovaný HMM fixní počet stavů, následně by mohl být rozšířen tak, aby dle kritérií pro odhad predikční chyby vybral vhodný počet stavů, který se může lišit dle konkrétního akciového titulu.

2 Úvod

Cílem obchodníků s akciemi je nakoupit akcie za nízkou cenu a prodat je za cenu vyšší. Nicméně, kdy je nejlepší čas na nákup nebo prodej akcií je náročná otázka. Investice do akcií tak mohou mít obrovský výnos nebo výrazné ztráty v důsledku vysoké volatility cen akcií.

Dle [2] bylo v poslední době publikováno mnoho výzkumů pro nalezení optimálního modelu pro predikcy chování akciových trhů. Většina výzkumů využívala techniky jako autoregression moving area (ARMA) nebo multiple regression modely. Objevují se také techniky založené na umělé inteligenci využívající neuronové sítě, fuzzy logiku a další.

V této práci bude použit skrytý markovský model (dále HMM), který pro předpověď ceny akcií použili Hassan a Nath [2] nebo Nguyet Nguyen [1] a naše práce bude z těchto dvou vycházet.

Hassan a Nath ve své práci [2] použili HMM pro předpověď ceny akcie na konci následujícího dne pro několik akcií aerolinek. Nguyet Nguyen svou prací [1] na tuto práci navazuje a přidává použití kritérií pro odhad predikční chyby, podle kterých vybírá počet stavů HMM.

3 Skrytý markovský model

Skrytý markovský model (HMM) je konečný automat s fixním počtem stavů, byl představen v 70. letech jako nástroj pro rozpoznávání řeči. Tento model je založen na statistických metodách a najde mnoho aplikací jako je rozpoznávání řeči, analýza sekvence DNA, rozpoznávání ručně psaného textu a další.

3.1 Definice

HMM je charakterizován:

- N - počet stavů modelu
- M - počet symbolů na stav
- T - délka pozorované sekvence
- O - pozorovaná sekvence $O_1, O_2, ...O_T$
- $A = \{a_i j\}$ - pravděpodobnost přechodu ze stavu i do stavu j
- $B = \{b_j(O_t)\}$ - pravděpodobnost pozorování O_t ve stavu j
- $\pi = \{\pi_i\}$ - pravděpodobnost bytí ve stavu i v čase t=1
- $\lambda = \{A, B, \pi\}$ - HMM model

Pokud jsou pravděpodobnosti spojitě rozděleny, máme spojitý HMM. V této práci předpokládáme, že model je spojitý a pravděpodobnost pozorování je Normální (Gaussovo) rozdělení. Potom $b_i(O_t) = N(O_t = \mu_i, \sigma_i)$, kde μ_i a σ_i jsou střední hodnota a rozptyl rozdělí pro stav S_i a parametry HMM jsou definovány jako:

$$\lambda = \{A, \mu, \sigma, \pi\}$$

3.2 Problémy a jejich řešení

Při řešení problémů pomocí HMM narazíme na tři problémy:

1. Pro pozorování $O = \{O_t, t = 1, 2, ...T\}$ a model $\lambda = \{A, \mu, \sigma, \pi\}$ najít pravděpodobnost pozorování $P(O|\lambda)$
2. Pro pozorování $O = \{O_t, t = 1, 2, ...T\}$ a model $\lambda = \{A, \mu, \sigma, \pi\}$ najít sekvenci stavů, která nejlépe popisuje pozorování

3. Pro pozorování $O = \{O_t, t = 1, 2, ...T\}$ zkalibrovat (natrénovat) parametry modelu $\lambda = \{A, \mu, \sigma, \pi\}$

1. Pravděpodobnost pozorování najdeme pomocí algoritmu Forward nebo Backward

2. Sekvenci stavů najdeme pomocí algoritmu Viterbi

3. Model zkalibrujeme pomocí algoritmu Baum-Welch

V naší práci využijeme algorimus Forward pro řešení prvního problému a algoritmus Baum-Welch pro řešení třetího problému (tento alogritmus využívá Forward a Backward). Druhý problém řešit nepotřebujeme.

4 Existující nástroje

Pro použití HMM v pythonu (ve kterém bude aplikace implementována) existuje několik nástrojů, zde následuje seznam vybraných:

- hmmlearn[1]
- pomegranate[2]
- PyHHMM[3]

4.1 hmmlearn

Knihovna pro práci s HMM v pythonu, prvotní verze programu byly testovány s touto knihovnou, ale celkem často padala.

4.2 pomegranate

Knihovna pro práci s HMM v pythonu, narozdíl od hmmlearn fungovala spolehlivě a byla použita v prvotních fázích vývoje. Její integrace v programu zůstala a lze ji použit na místo našeho implementovaného modelu (viz uživatelská příručka). Je napsána v cythonu a obsahuje paralelizace, proto je mnohom výkonnější než naše implementace.

4.3 PyHHMM

Další knihovna pro práci s HMM v pythonu s jednoduchou a transparentní implementací, některé její části byly použity i v naší implementaci.

[1] https://github.com/hmmlearn/hmmlearn
[2] https://github.com/jmschrei/pomegranate
[3] https://github.com/fmorenopino/HeterogeneousHMM

5 Predikce cen akcií

V této části je popis, jak využít HMM k predikci cen akcií dle práce Nguyena [1], která vychází z metody z práce Hassana [2].

5.1 Výběr modelu

Nguyen navrhuje pro výběr počtu stavů modelu použít čtyři různá kritéria pro odhad predikční chyby:

- AIC - Akaike information criterion
- BIC - Bayesian information criterion
- HQC - Hannan-Quinn information criterion
- CAIC - Bozdogan Consisten Akaike information criterion

$$AIC = -2ln(L) + 2k$$
$$BIC = -2ln(L) + kln(M)$$
$$HQC = -2ln(L) + kln(ln(M))$$
$$CAIC = -2ln(L) + k(ln(M) + 1)$$

kde L je pravděpodobnost modelu (likelihood), M je počet prvků v pozorované sekvenci a k je počet parametrů modelu, kde v případě Normálního rozdělení $k = N^2 + 2N - 1$, kde N je počet stavů modelu.

Nguyen ve své práci testoval model pro predikci měsíčních cen indexu S&P 500 pro všechna výše zmíněná kritéria (kde nižšího hodnota znamená lepší model) a dle jejich výsledku se rozhodl použít model se čtyřmi stavy. Jak autor ale sám zmiňuje, pro každý akciový titul, mohou vyjít kritéria různě. Proto do naší aplikace přidáme automatické nalezení nejlepšího modelu, kdy budeme testovat modely pro 2,3,4,5 a 6 stavů. Pro každý model spočteme všechna kritéria a uděláme z nich průměr, model s nejnižším průměrem vybereme jako nejlepší a použijeme ho pro následnou predikci cen akcií.

5.2 Inicializace parametrů

Prvotní parametry modelu A a π inicializujeme dle [1] takto:

$$A = (a_i j), a_i j = 1/N$$
$$\pi = (1, 0, ...0)$$

Parametry μ a σ inicializujeme dle ukázkové pozorované sekvence pomocí algoritmu KMeans, jako v případě knihovny PyHHMM.

5.3 Predikce

Pozorovaná sekvence je složena ze čtyř sekvení: otevírací, nejnižší, nejvyšší a zavírací cena:

$$O = \{O_t^{(1)}, O_t^{(2)}, O_t^{(3)}, O_t^{(4)}\}$$

Pro predikci zavírací ceny v čase T+1, použijeme jako první trénovací sekvenci

$$O = \{O_t^{(1)}, O_t^{(2)}, O_t^{(3)}, O_t^{(4)}, t = T - D, T - D + 1, ...T\}$$

kde D je tzv. tréninkové okno (konstanta). Spočteme pravděpodobnost tohoto pozorování $P(O|\lambda)$ a posuneme testovací sekvenci o jeden den zpět a dostaneme:

$$O^{new} = \{O_t^{(1)}, O_t^{(2)}, O_t^{(3)}, O_t^{(4)}, t = T - D - 1, T - D, ...T - 1\}$$

a spočteme pravděpodobnost pozorování $P(O^{new}|\lambda)$. Takto budeme posouvat až na počátek pozorované sekvence a následně vybereme takovou sekvenci O^*, kde $P(O^*|\lambda) \simeq P(O|\lambda)$. Zavírací cenu v čase T+1 určíme takto:

$$O_{T+1}^{(4)} = O_T^{(4)} + (O_{T^*+1}^{(4)} - O_{T^*}^{(4)}) \times sign(P(O|\lambda) - P(O^*|\lambda))$$

Vždy předpovídáme cenu pouze jeden den dopředu a k terénování používáme pouze data od tohoto dne zpět. Pro předpovězení dalšího dne poustupujeme stejně jen k trénování použijeme jeden den navíc (ten co už jsme předtím předpovídali). V aplikaci je implementovaná předpověď pro D dnů (D je parametr programu viz uživatelská příručka).

6 Implementace

Aplikace je implementována v pythonu, projekt je rozdělen takto:

- *data/data.py* - funkce pro načítání dat z csv souboru a finance.yahoo
- *error/error.py* - funkce pro výpočet chyb viz výsledky
- *hmm/criteria.py* - funkce pro výpočet kritérií viz výše
- *hmm/hmm.py* - vlastní implementace hmm modelu
- *plot/plot.py* - funkce pro vykreslení grafů
- *utils* - pomocné funkce (parsování argumentů, logování)
- *forecast.py* - funkce pro předpověď cen za využití hmm modelu
- *main.py* - vstup programu

6.1 HMM

Implementace modelu vychází ze článku [1], části jsou inspirovány knihovnou **PyHHMM**.

6.1.1 Rozhraní

Model má stejné rozhraní jako knihovna **pomegranate** a proto je možná modely měnit viz uživatelská příručka. Rozhraní definuje dvě funkce:

```
def fit(self, obs) -> None
def log_probability(self, obs) -> float
```

Funkce *fit* slouží k natrénování modelu pomocí pozorované sekvence, funkce *log_probability* slouží ke zjištění pravděpodobnosti (jejímu logaritmu) pozorování.

6.1.2 Forward algoritmus

Pro zjištění pravděpodobnosti pozorování je využit algoritmus Forward, který je definovaný v článku [1]. Je použita mírná modifikace (jako v knihovně **PyHHMM**), která počítá v logaritmickém měřítku.

6.1.3 Backward algoritmus

Dále je implementován algoritmus Backward, který pracuje na podobném principu jako předchozí a je použit u algoritmu Baum-Welch viz dále.

6.1.4 Baum-Welch algoritmus

Pro natrénování modelu je použit algoritmus Baum-Welch, který je popsán v článku [1].

6.2 Predikce

Predikce cen akcií je implementována přesně dle postupu popsaného v sekci 5.3.

7 Ověření modelu

Pro ověření modelu, je spočítána střední absolutní procentní chyba (MAPE) a koeficient determinace (r2) stejně jako v článku [2].

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}|\frac{A_t - F_t}{A_t}|$$

kde A_t je aktuální hodnota a F_t je předpoveď v čase t.

8 Výsledky

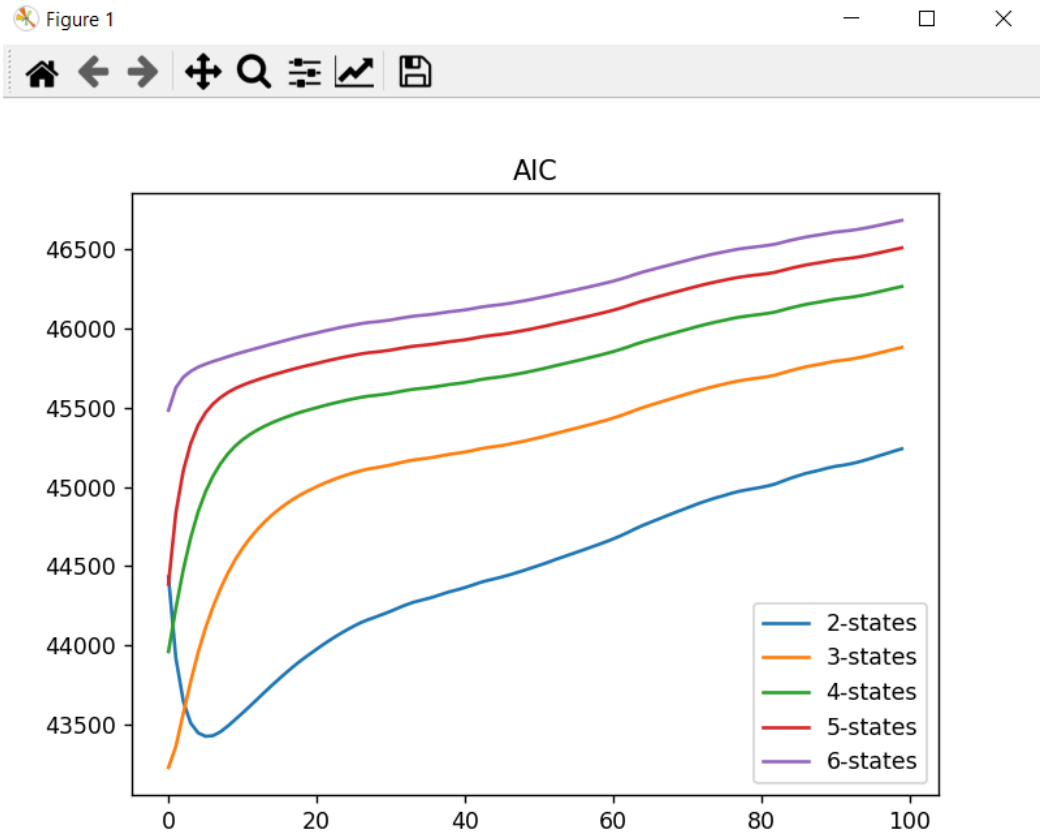
8.1 Testovací scénář

Pro otestování modelu byl použit index S&P 500 jako v případě [1] (ticker SPY), ale byly použity historické denní ceny od 1.1.2019 do 1.1.2022 na časové okno 100 dní, tedy na 2.1.2022 a 100 (obchodních) dní zpět.

```
python -m hmm_stock_forecast.main -s 2019-01-01 -e
2022-01-01 -t SPY -w 100
```

8.2 Optimální počet stavů

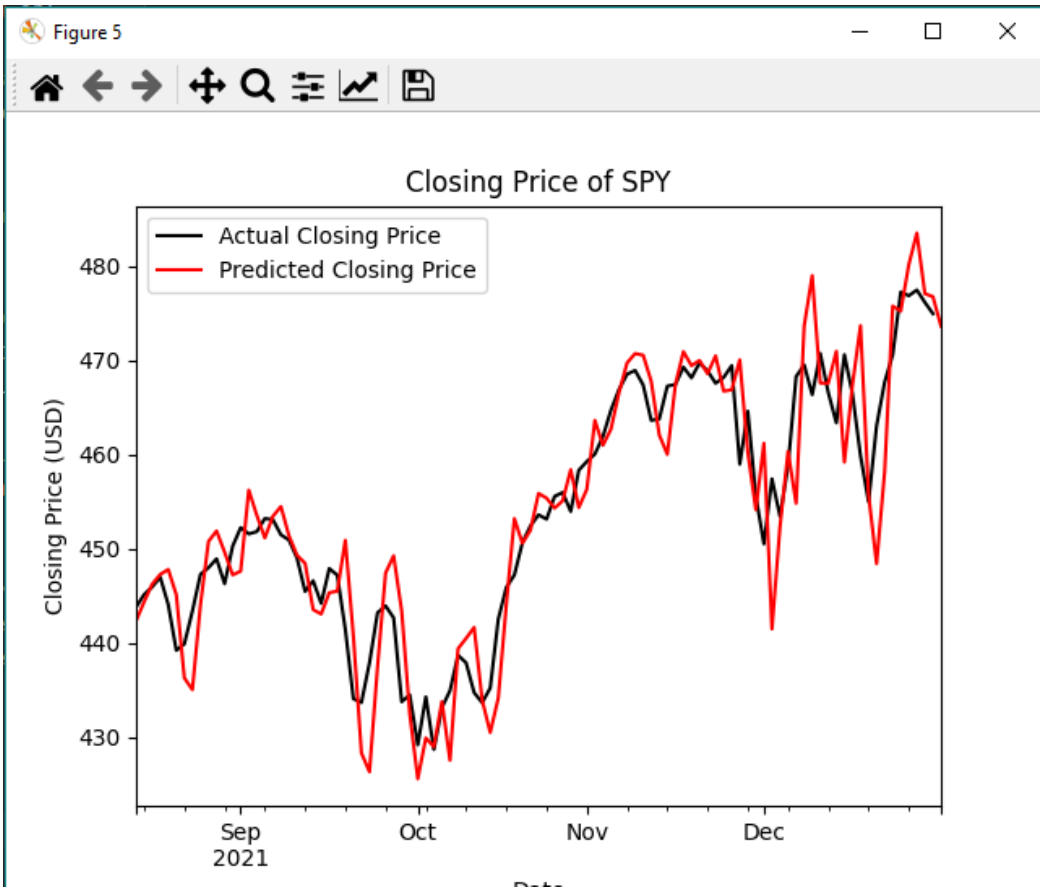
Program našel optimální počet stavů dle zmíněných kritérií na 2 stavy a vykreslil grafy pro jednotlivá kritéria viz obrázek 1 pro AIC.



Obrázek 1: AIC

8.3 Predikce cen

Na obrázku 2 je vidět graf predikovaných a reálných cen na časové okno 2.1.2022 a 100 (obchodních) dní zpět.



Obrázek 2: Výsledek

8.4 Chyby

Pro náš testovací scénář vyšly chyb takto:

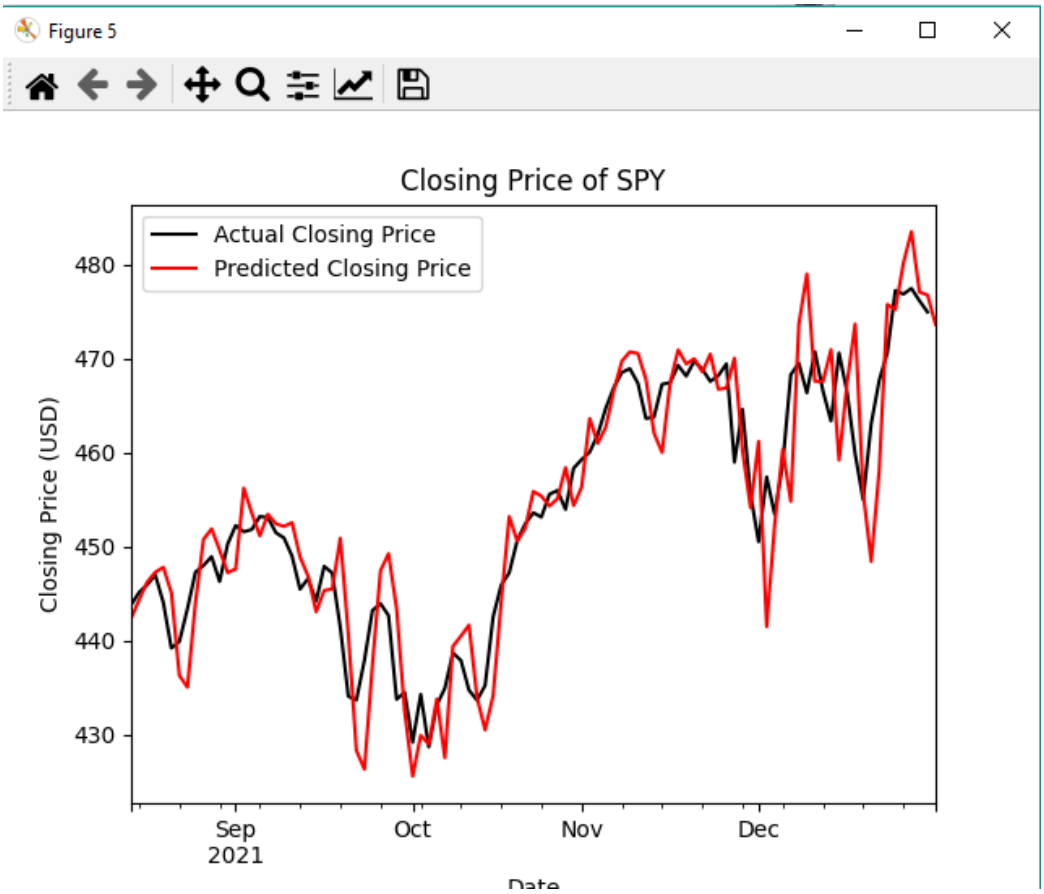
$$MAPE = 0.861$$

$$r2 = 0.822$$

8.5 Porovnání s pomegranate

Stejný test byl proveden také s modelem z knihovny **pomegranate**, který vybral také jako optimální dva stavy a výsledek predikce je vidět na obrázku 3.

```
python -m hmm_stock_forecast.main -s 2019-01-01 -e
2022-01-01 -t SPY -w 100 -m pomegranate
```



Obrázek 3: Výsledek

$$MAPE = 0.860$$

$$r2 = 0.822$$

Je také nutno podotknout, že model z knihovny **pomegranate** běží mnohem rychleji z důvodu, že části jsou psané v cythonu a využívá paralelizace.

9 Uživatelská příručka

9.1 Požadavky

Pro spuštění aplikace je potřeba mít nainstalovaný python 3.

9.2 Spuštění ze zdrojových kódů

Projekt využívá pro správu zavíslostí nástroj Poetry, který je možné nainstalovat pomocí příkazu:

```
pip install poetry
```

Následně pomocí poetry nainstalujeme všechny závislosti definované v souboru *pyproject.toml* pomocí příkazu:

```
poetry install
```

Vlatní aplikaci můžeme spustit pomocí příkazu (parametry viz níže):

```
poetry run start
```

9.3 Spuštění z wheel balíčku

K dispozici je také *.wheel* soubor *hmm_stock_forecast-1.0-py3-none-any.whl*, který můžeme nainstalovat id bez Poetry pomocí příkazu:

```
python -m pip install hmm_stock_forecast-1.0-py3-none-any.whl
```

A následně spustit (parametry viz níže):

```
python -m hmm_stock_forecast.main
```

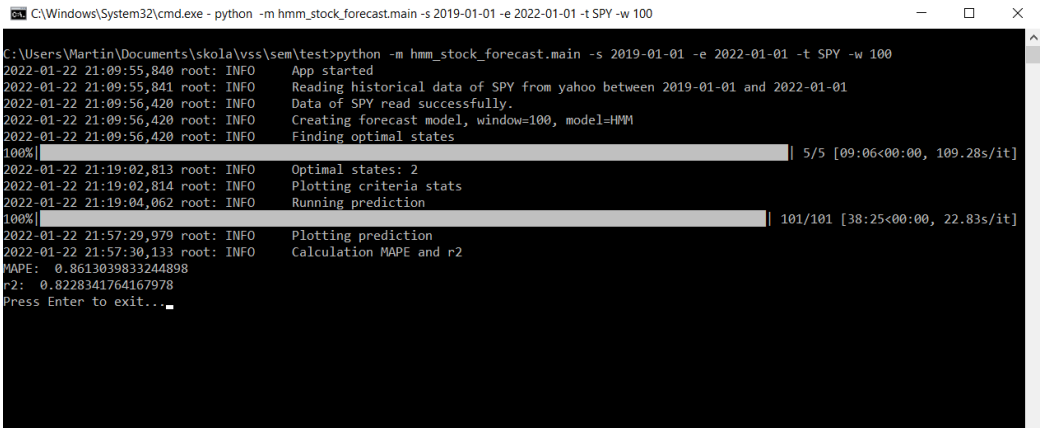
9.4 Parametry programu

Aplikace očekává následující parametry:

- -t ticker akciového titulu (např. AAPL)
- -f cesta k csv souboru (4 sloupce: Open, Low, High, Close)
- -s datum počátku dat, která se mají stáhnout
- -e datum konce dat, která se mají stáhnout
- -w velikost okna viz výše (defaultně 120)

- -m typ modelu:
  - HMM - naše implementace modelu (defaultně)
  - pomegranate - využití modelu knihovny pomegranate

Spuštění je tedy možné buď s -f nebo -t, v tom případě jsou povinné i datumy pro stažení dat z yahoo finance. Na obrázku 4 je vidět výstup aplikace.



Obrázek 4: Výstup aplikace

10 Závěr

Implementoval jsem skrytý markovský model, který jsem použil pro predikci cen akcií pomocí metody vycházející ze článku Hidden Markov Model for Stock Trading [1]. Model jsem otestoval pro predikci cen indexu SPY 500 a porovnal s existující knihovnou pro práci s HMM - pomegranate.

Respositář je dostupný zde: https://github.com/MFori/HMM-Stock-Forecast

Reference

[1] Nguyen N. *Hidden Markov Model for Stock Trading.* International Journal of Financial Studies. 2018; https://doi.org/10.3390/ijfs6020036.

[2] M. R. Hassan and B. Nath, *Stock market forecasting using hidden Markov model: a new approach* 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005, pp. 192-196, doi: 10.1109/ISDA.2005.85.