NLP módszerek alkalmazása tőzsdei árfolyamok becsléséhez

Komplex Tervezés

Készítette:
Szemán Balázs
BB89VX

Témavezető:
Dr. Karácsony Zsolt

2024.

Contents

Python Implementáció

1 Imports

Python packagek importálása:

- Seaborn:
  egy matplotlib alapú Python adatvizualizációs könyvtár. Magas szintű felületet biztosít vonzó és informatív statisztikai grafikák rajzolásához.
- Pandas:
  egy Python programozási nyelvre írt szoftverkönyvtár adatkezelésre és -elemzésre. Különösen adatstruktúrákat és műveleteket kínál numerikus táblák és idősorok manipulálásához.
- MatPlotLib:
  egy plotting könyvtár a Python programozási nyelvhez és annak NumPy numerikus matematikai kiterjesztéséhez
- Transformers:
  API-kat és eszközöket biztosít a legkorszerűbb előképzett modellek egyszerű letöltéséhez és betanításához.
- Scipy:
  egy ingyenes és nyílt forráskódú Python-könyvtár, amelyet tudományos és műszaki számítástechnikai célokra használnak. A SciPy modulokat tartalmaz az optimalizáláshoz, a lineáris algebrához, az integrációhoz, az interpolációhoz, a speciális függvényekhez, az FFT-hez, a jel- és képfeldolgozáshoz, az ODE-megoldókhoz és más, a tudományban és a mérnöki tudományokban megszokott feladatokhoz.
- tqdm:
  vizualizálja a loopokat, progress bar-t hoz létre nekik.

[4]:
```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from scipy.special import softmax
from tqdm.notebook import tqdm

plt.style.use('ggplot')
```

2 Adatok Betöltése

A vizsgálni kívánt adatok beolvasása és előkészítése a kívánt műveletek végrehajtásához.
A beolvasás a pandas.read_csv() függvényével történik, ami egy könnyen átlátható és kezelhető DataFrame-et.
Az aapl elnevezésű változók az Apple tőzsdei adatait és a hozzájuk tartozó híreket tárolják.
Az msft elnevezésű változók az Microsoft tőzsdei adatait és a hozzájuk tartozó híreket tárolják.

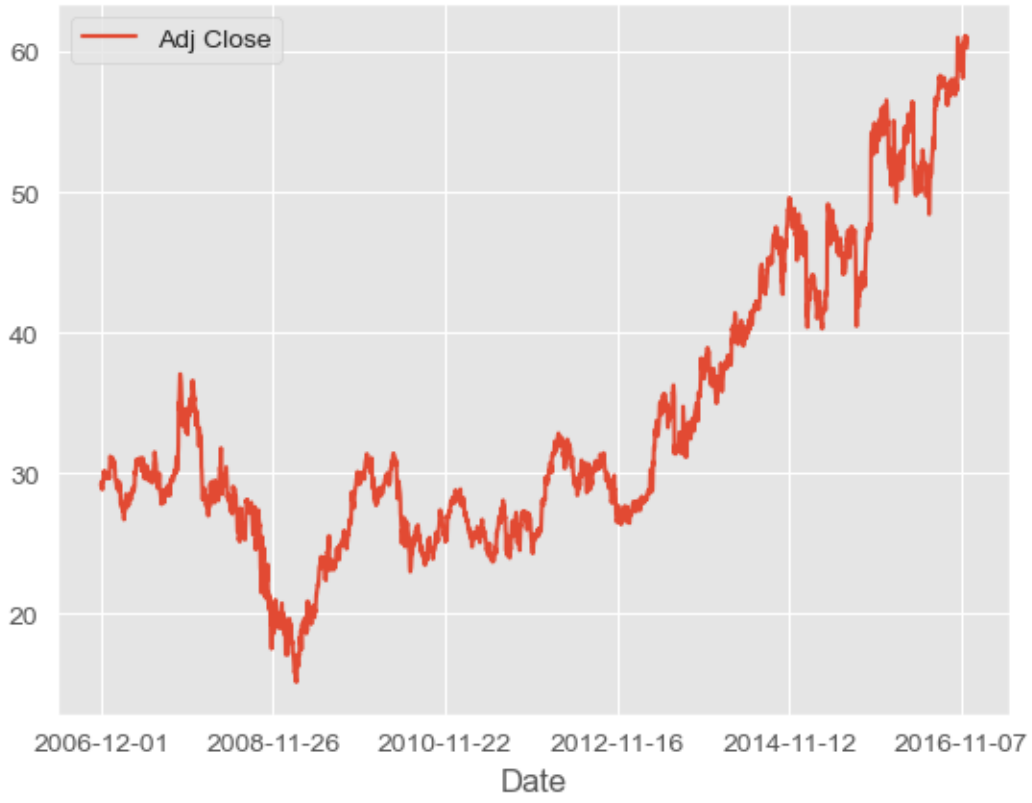
[8]:
```
aapl_news = pd.read_csv('input/source_1/AppleNewsStock.csv')

msft_news = pd.read_csv('input/source_1/MicrosoftNewsStock.csv')
msft_news.drop(
    msft_news.columns[
        msft_news.columns.str.contains(
    'unnamed', case=False)],
    axis=1, inplace=True)
```

[9]:
```
aapl_news.plot.line(y="Adj Close", x='Date')
```

[9]:
```
<Axes: xlabel='Date'>
```

[10]:
```
msft_news.plot.line(y="Adj Close", x='Date')
```

[10]:
```
<Axes: xlabel='Date'>
```



[11]:
```
aapl_news["Tomorrow"] = (aapl_news["High"].shift(-1)+aapl_news["Low"].shift(-1))/
↪2
aapl_news["Target"] = (aapl_news["Tomorrow"] /␣
↪((aapl_news["High"]+aapl_news["Low"])/2) - 1)


msft_news["Tomorrow"] = (msft_news["High"].shift(-1)+msft_news["Low"].shift(-1))/
↪2
msft_news["Target"] = (msft_news["Tomorrow"] /␣
↪((msft_news["High"]+msft_news["Low"])/2) - 1)
```

3 Sentiment Analysis

A Sentiment Analysis model letöltése a huggingface.co-ról.
A huggingface.co egy természetes nyelvi feldolgozó alkalmazásokhoz készült transzformátorkönyvtár, amely platformja lehetővé teszi a felhasználók számára, hogy megosszák a gépi tanulási modelleket és adatkészleteket, és bemutassák munkájukat.
A letöltött modelt a transformers AutoTokenizer és AutoModelForSequenceClassification moduljaival feldolgozuk tokenizer és model-re.

[ ]:
```
MODEL = f"mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

A model és tokenizer felhasználásra kerül a "News" azaz "Hírek" elnevezésű oszlop elemeinek kiértékelésére készült polarity_scores_roberta(text) függvényben.
A kiértékelés megadja, hogy az adott hír cikk pozitív, negatív vagy semleges véleményű.

[ ]:
```
def polarity_scores_roberta(example):
    example = example[:514]
    encoded_text = tokenizer(example, return_tensors='pt')
    output = model(**encoded_text)
    scores = output[0][0].detach().numpy()
    scores = softmax(scores)
    scores_dict = {
        'roberta_neg': scores[0],
        'roberta_neu': scores[1],
        'roberta_pos': scores[2]
    }
    return scores_dict
```

A get_results(dataframe, news, debug=False) függvény paraméterként kap egy feldolgozni kívánt DataFrame-et és annak "Hírek"-et tartalmazó oszlopának nevét és sentiment analízist végez az összes elemére a polarity_scores_roberta(text) függvénnyel. A kapott értékek(pozitiv, negatit, neutral) közül a legnagyobbat kíszűri, majd előjelezi és hozzácsatolja az eredeti DataFrame-hez "sentiment score" néven, utána vissza tér vele.

[ ]:
```
def get_results(df, news, debug=False):
    res = {}
    fail = {}
    n = 0
    for i, row in tqdm(df.iterrows(),
                       total=len(df)):
```

```
        text = row[news]
        try:
            if pd.isna(text):
                roberta_result = {
                    'roberta_neg': 0,
                    'roberta_neu': 1,
                    'roberta_pos': 0
                }

            else:
                roberta_result = polarity_scores_roberta(text)

        except (IndexError, RuntimeError):
            if pd.isna(text) & debug:
                fail[n] = i.astype(str)
            elif debug:
                fail[n] = text
                n += 1
            roberta_result = {
                'roberta_neg': 0,
                'roberta_neu': 0,
                'roberta_pos': 0
            }
            pass
        res[i] = roberta_result

    res = pd.DataFrame(res).T

    res['sentiment score'] = res.apply(
        lambda x: -1
        if max(x['roberta_neg'],
               x['roberta_pos'],
               x['roberta_neu']) == x['roberta_neg']
        else (1 if max(x['roberta_neg'],
                       x['roberta_pos'],
                       x['roberta_neu']) == x['roberta_pos']
              else 0) * max(x['roberta_neg'], x['roberta_pos'],␣
↪x['roberta_neu']), axis=1
    )
    res = pd.merge(df, res['sentiment score'], left_index=True,␣
↪right_index=True, suffixes=('_original', ''))

    if debug:
        return res, fail
    else:
        return res
```

A check_prediction(dataframe, target, score, nautral) függvény, a már "sentiment score"-okkal ellátott DataFrame-et vizsgálja meg, ahol a target és a score iránya megegyezik (növekvés, csökkenés vagy semleges) ott 1, ahol nem ott 0 értéket állít be a "predictions" oszlopba. A semleges érték határát be lehet állítani a neutral elnevezású paraméterrel (0-1 közötti érték).
A függvény visszatérési értéke az eredeti tömb, amihez a "predictions" oszlop hozzá lett csatolva.

[ ]:
```
def check_prediction(df, tar, sco, neutral, debug=False):
    predict_dict = {}
    fail = {}
    n = 0
    for i, row in tqdm(df.iterrows(),
                       total=len(df)):
        try:
            target = row[tar]
            score = row[sco]
            if abs(target) > neutral:
                if (target / score) > 0:
                    predict_dict[i] = 1
                elif (target / score) < 0:
                    predict_dict[i] = 0
            elif score == 0:
                predict_dict[i] = 1
            else:
                predict_dict[i] = 0
        except KeyError:
            predict_dict[i] = 'null'
        except:
            if debug:
                fail[n] = i
                n += 1
            predict_dict[i] = 0
            pass
    predictions = pd.DataFrame(predict_dict, index=[0]).T
    predictions.columns = ['predictions']
    predictions = pd.merge(df, predictions['predictions'], left_index=True,␣
↪right_index=True)
    if debug:
        return predictions, fail
    else:
        return predictions
```

3.1 -Microsoft

A Microsoft tőzsdei értékeinek feldolgozása a Sentiment Analyzis fügvényekkel.

[13]:
```
msft_result = get_results(msft_news,'News')
```

```
  0%|          | 0/2517 [00:00<?, ?it/s]
```

[109]:
```
msft_check_prediction = check_prediction(msft_result,'Target','sentiment score',␣
↪0.06)
```

```
  0%|          | 0/2517 [00:00<?, ?it/s]
```

[110]:
```
msft_check_prediction[['Target', 'sentiment score', 'predictions']]
```

[110]:

	Target	sentiment score	predictions
0	0.008419	0.000000	1
1	-0.004430	0.000000	1
2	-0.007359	0.000000	1
3	-0.002069	0.000000	1
4	0.005529	0.000000	1
...	...	...	...
2512	-0.005686	0.999217	0
2513	0.004724	0.000000	1
2514	0.005774	0.000000	1
2515	-0.004347	0.000000	1
2516	NaN	0.000000	1

[2517 rows x 3 columns]

A kapott "predictions" oszlop pontosságának kiszámítasa.

[111]:
```
msft_prediction_accuracy = (sum((msft_check_prediction['predictions']))/
↪len(msft_check_prediction.index))
msft_prediction_accuracy
```

[111]:
```
0.7167262614223282
```

A kapott értékből látható, hogy a becslés módja nem tökélet csak $\tilde{7}2\%$ -ban megízható.

3.2 -Apple

Az Apple tőzsdei értékeinek feldolgozása a Sentiment Analyzis fügvényekkel.

[37]:
```
aapl_results = get_results(aapl_news,'News')
```

```
  0%|          | 0/2517 [00:00<?, ?it/s]
```

[83]:
```
aapl_check_prediction = check_prediction(aapl_results,'Target','sentiment␣
↪score', 0.06)
```

```
  0%|          | 0/2517 [00:00<?, ?it/s]
```

[85]:
```
aapl_check_prediction[['Target', 'sentiment score', 'predictions']]
```

[85]:

	Target	sentiment score	predictions
0	0.000658	0.000000	1
1	0.003561	0.000000	1
2	-0.011681	0.000000	1
3	-0.020214	0.000000	1
4	-0.005693	-1.000000	0
...	...	...	...
2512	0.004418	0.000000	1
2513	0.004667	0.628444	0
2514	-0.007862	0.000000	1
2515	0.001666	0.000000	1
2516	NaN	0.000000	1

[2517 rows x 3 columns]

A kapott "predictions" oszlop pontosságának kiszámítasa.

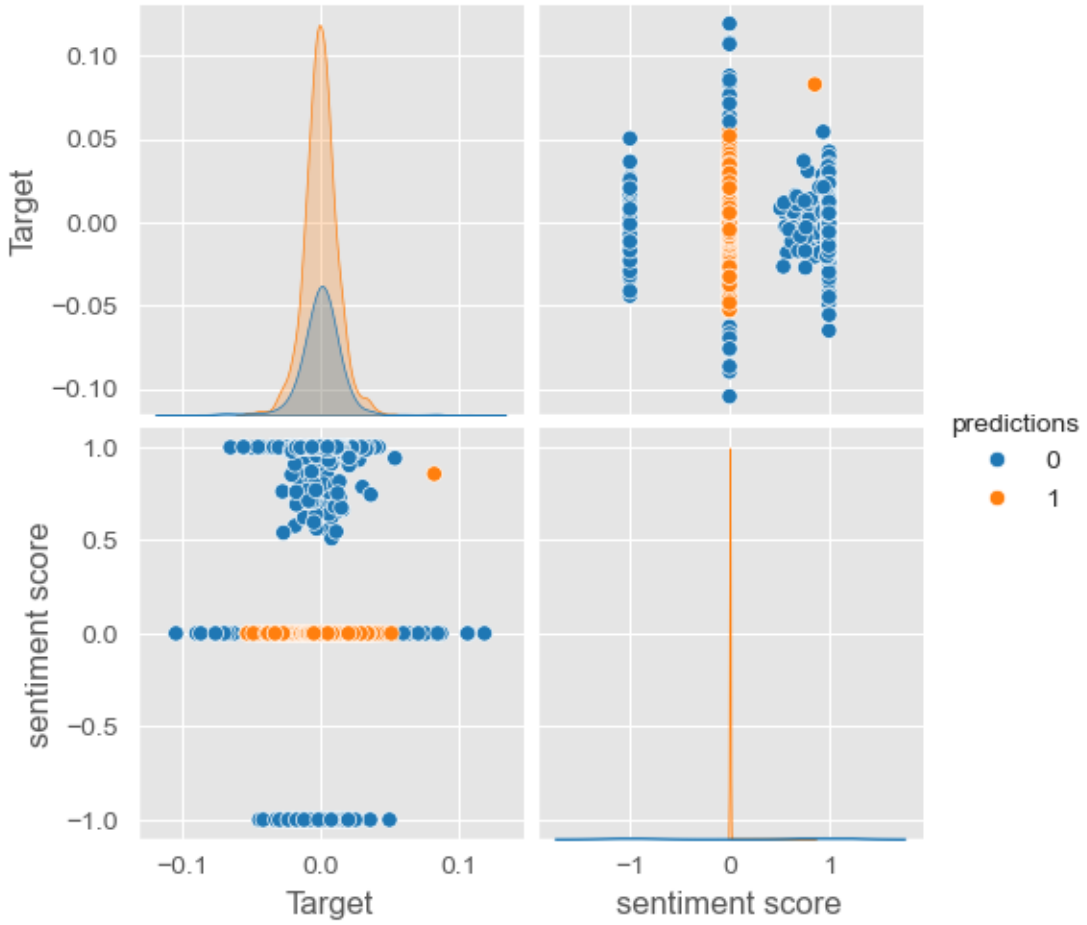
[86]:
```
aapl_prediction_accuracy = (sum((aapl_check_prediction['predictions']))/
↪len(aapl_check_prediction.index))
aapl_prediction_accuracy
```

[86]:
```
0.5490663488279698
```

A kapott értékből látható, hogy a becslés módja nem tökélet, nem eléggé megbízható csak $\tilde{5}5\%$. Ezt az értéket befojásolhatja a hírek relevanciája az adott részvénzhez, mivel a kapott szövegek a Apple-höz kapcsolódó híreken kívül más értékeket is tartalmaztak. Ezt lehet a kapott adatok szűrésével javítani.
Továbbá a becslések nem veszik figyelembe az árfolyam jelenlegi tendenciáit, erre a kód továbbfejlesztésével lehet opciót adni, a releváns időintervallum beállításával.

4 Plot Result Comparison

[115]:
```
sns.pairplot(data=msft_check_prediction,
             vars=['Target','sentiment score'],
             hue='predictions',
             palette='tab10')
plt.show()
```



[116]:
```
sns.pairplot(data=aapl_check_prediction,
             vars=['Target','sentiment score'],
             hue='predictions',
             palette='tab10')
plt.show()
```