Stock market prediction (early signal model) for most trending S&P 500 stocks on Reddit

Project presentation: Python for finance II

University of Vienna

Bernd Prostmaier

Agenda

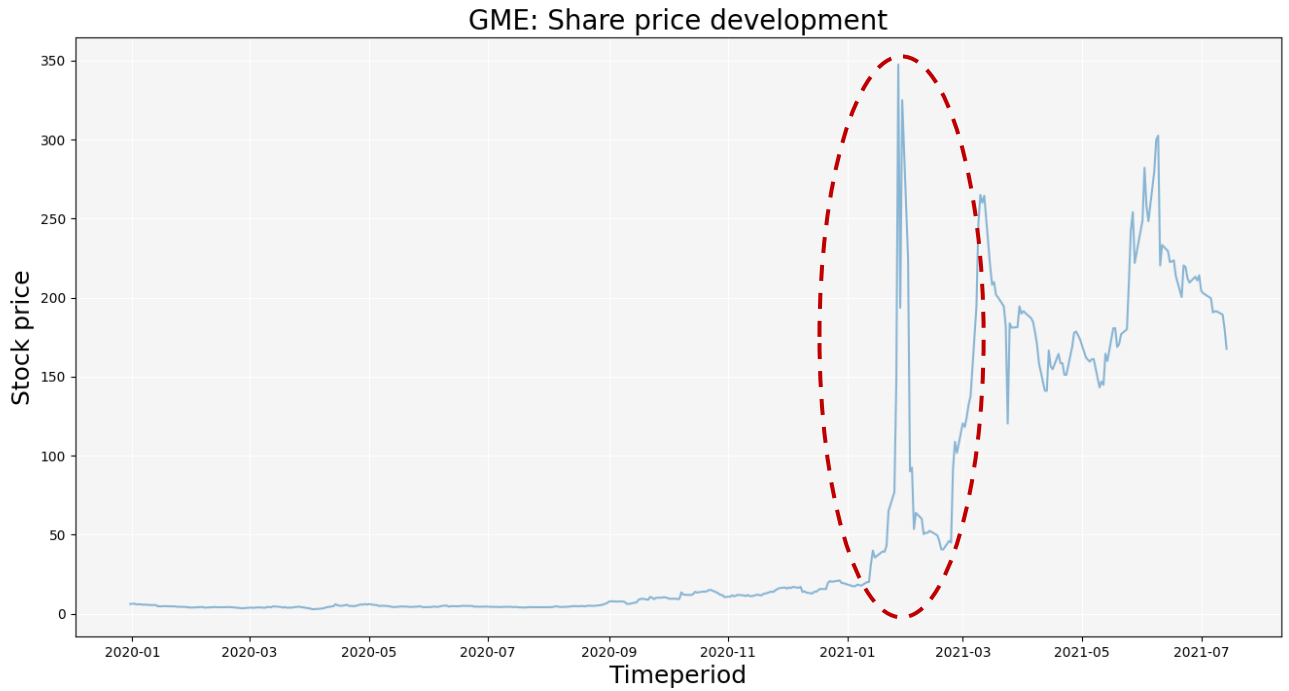
1. Introduction
2. Research objective
3. Data and methodology
    1. Social media publications
        1. Reddit
        2. Sentiment analysis
    2. Machine learning
        1. Random forest
        2. Model preparation
        3. Prediction & early signal model
4. Output and Conclusion

INTRODUCTION

Introduction

Why choosing stock market prediction for most trending stocks on social media?

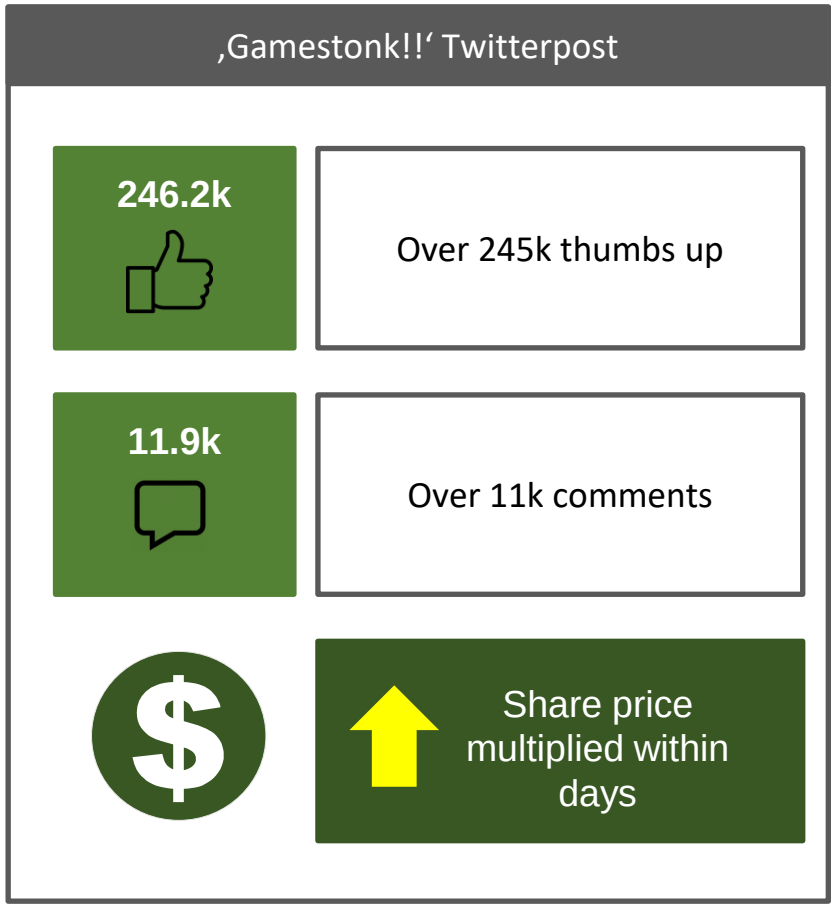
‚Gamestonk!! r/wallstreetbets' *(Twitter: Elon Musk, 26 January 2021)*



Source: yfinance library (Python); https://www.cnbc.com/2021/01/29/elon-musks-tweets-are-moving-markets.html

Introduction

Why choosing stock market prediction for most trending stocks on social media?



One post from Elon Musk was enough to cause the share price of GameStop to skyrocket within one day!

It could be seen that social media sentiment can have substantial economic value.

The question is whether social media publications are an indicator of whether the share price of a particular stock is more likely to increase or decrease in the near future.

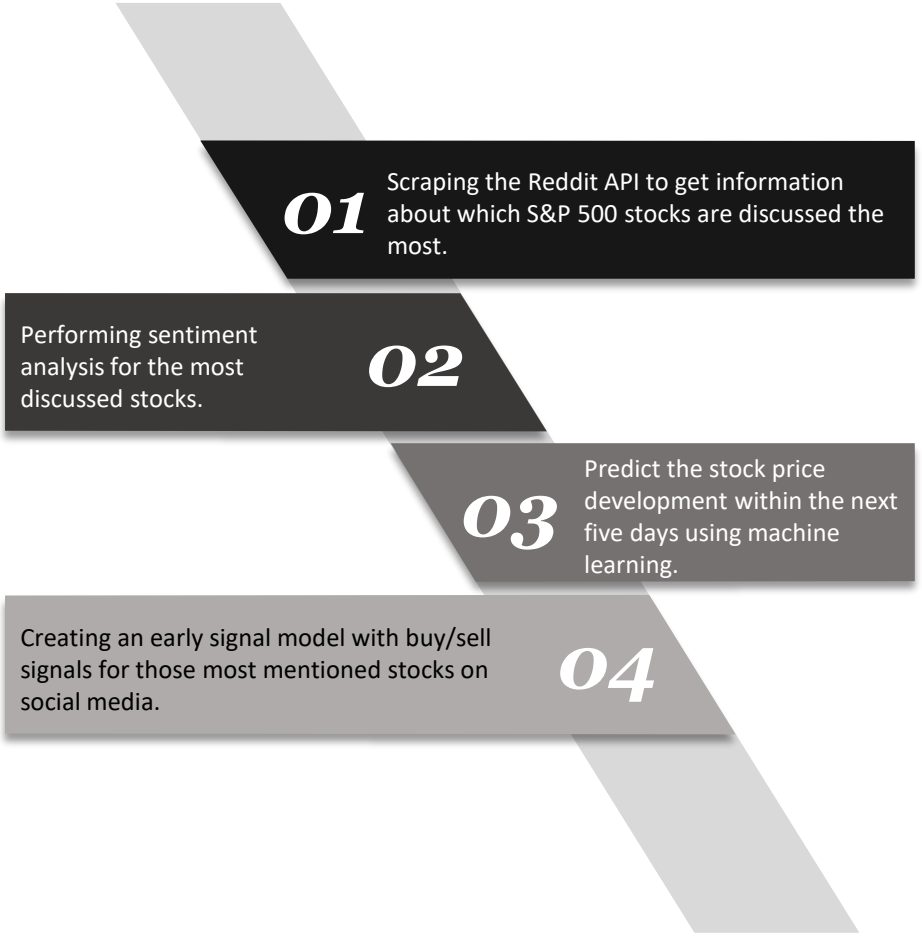
Source: https://www.cnbc.com/2021/01/29/elon-musks-tweets-are-moving-markets.html

RESEARCH OBJECTIVES

Research objectives

There are primarily four goals for this project

RESEARCH OBJECTIVES



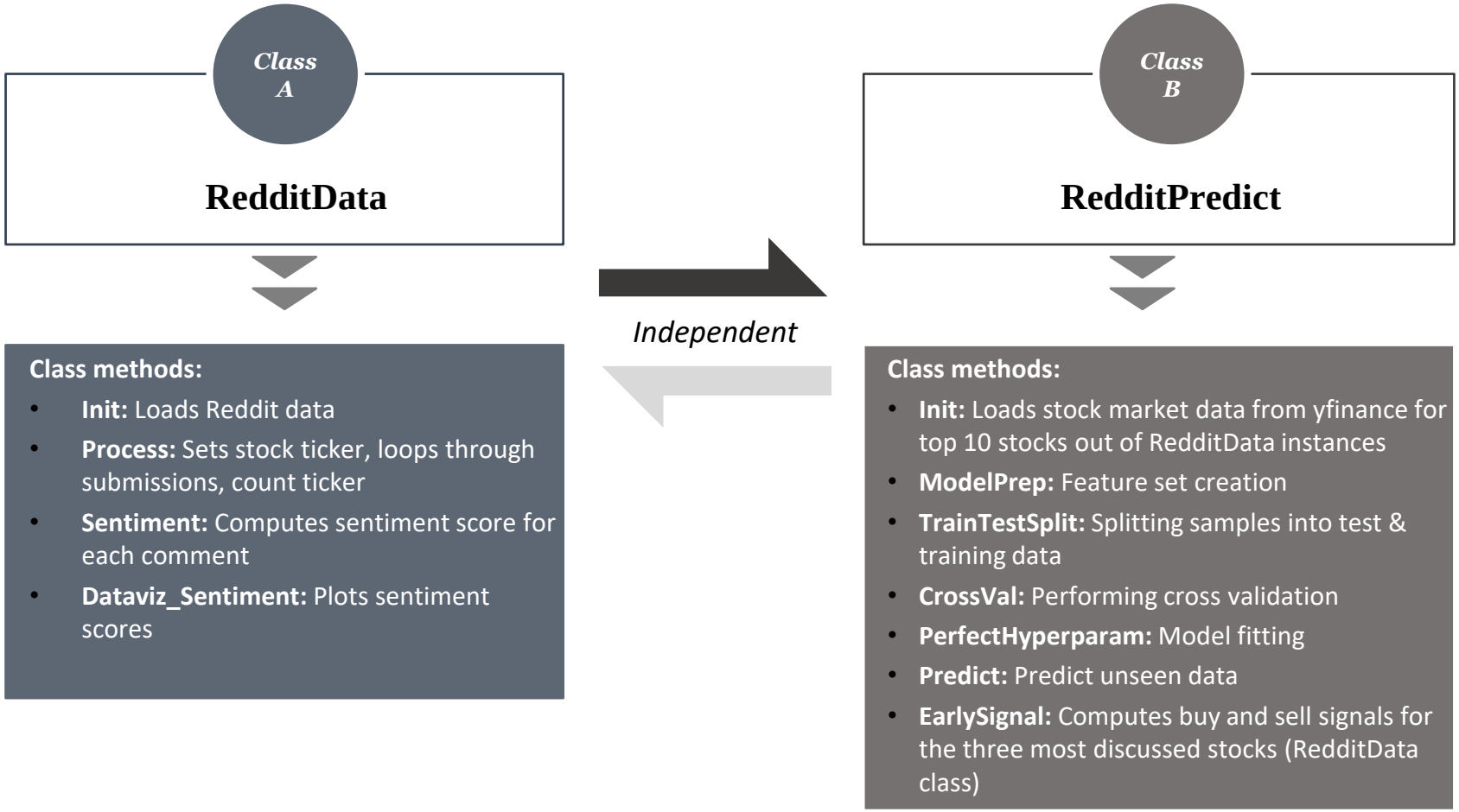
DATA AND METHODOLOGY

Data and methodology

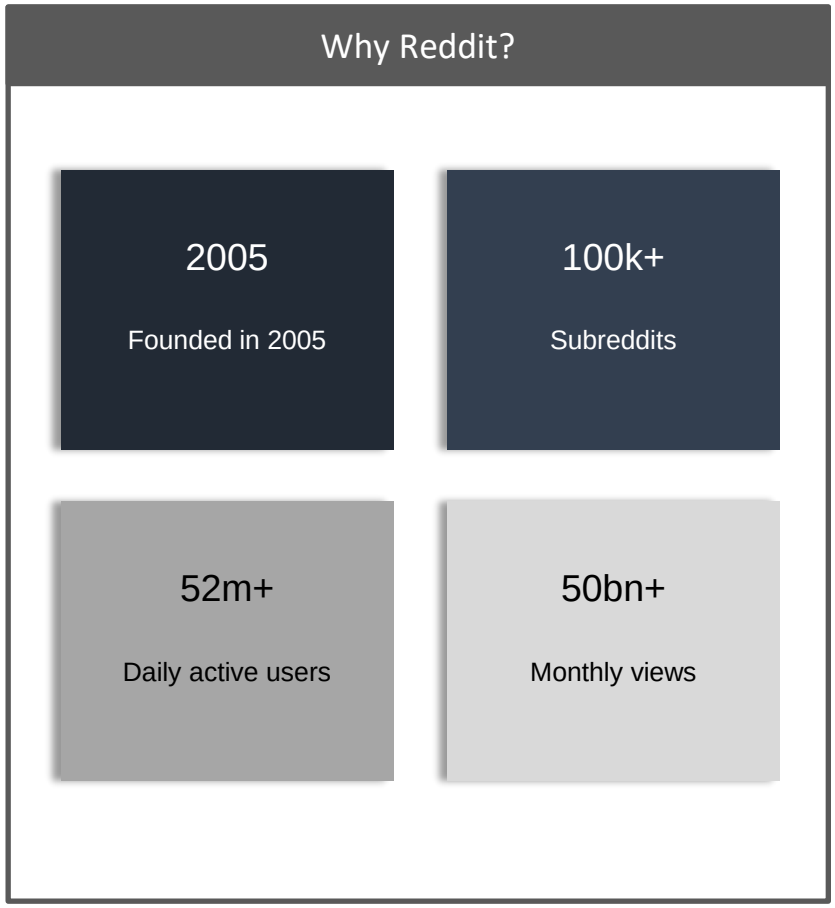
1. Social media publications
    a) Reddit
    b) Sentiment analysis
2. Machine learning
    a) Random forest
    b) Model preparation
    c) Prediction & early signal model

Python code

Divided this project into two python classes



Social media publications
Why choosing Reddit?



Why choosing Reddit instead of other social media plattforms (e.g. Twitter):

- With over 52m daily active users and 50bn monthly views Reddit is one of the top five most visited websites in the U.S.
- One place (i.e. subreddit) for people sharing investment opportunities
- Easy to scrap API (‚praw') via python

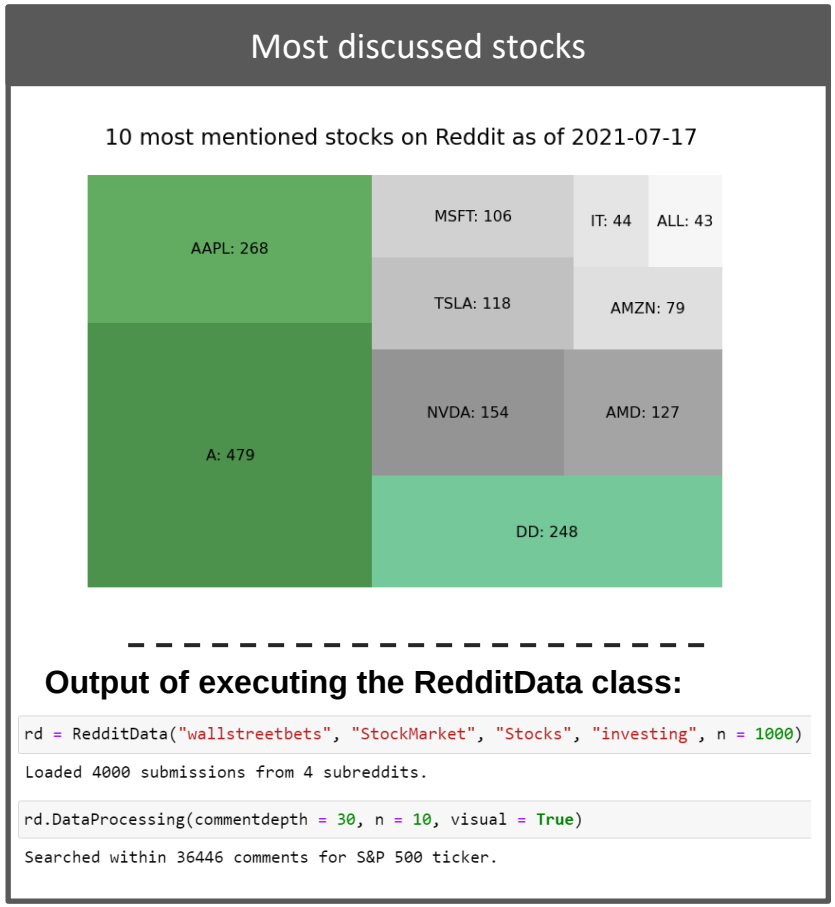
Scraping the ‚hottest' submissions from the following subreddits:

- Wallstreetbets
- Stocks
- StockMarket
- Investing

Analyzed 4,000 submissions with more than 35,000 comments!

Source: https://www.redditinc.com/

Social media publications
1. Step: Ticker counting

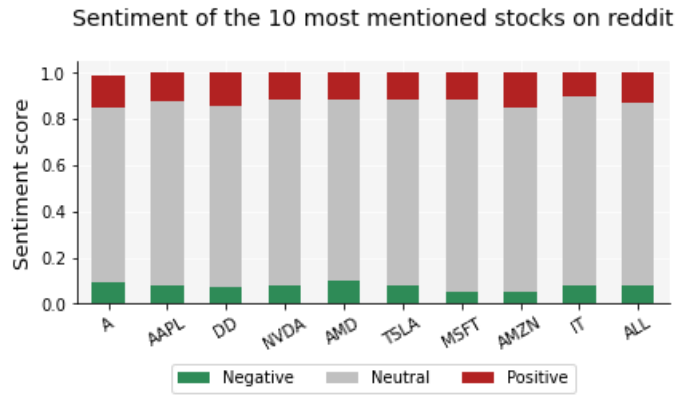


Ticker counting procedure:

- Scraping S&P 500 ticker from Wikipedia
- Looped through the comment body of every submission, resulting in analyzing over 35,000 comments
- Splitted every comment into words and checked whether a ticker was written within the comment or not
- If there was a match, the ticker counter goes up and the comment as a whole was saved for later sentiment analysis
- Executing the RedditData class method ‚DataProcessing‘ prints the plot on the left side showing the top 10 most mentioned stocks from that particular day

Social media publications
2. Step: Sentiment analysis

Most discussed stocks



Output of executing RedditData methods:

```
rd.Sentiment(threshold = 0.5)
```

```
rd.Dataviz_Sentiment()
```

Sentiment analysis procedure:

Using ‚Vader Sentiment Analyzer' for performing sentiment analysis. Vader already supports:

- Conventional use of punctuation (e.g. ‚Good!!!')
- Utf-8 encoded emojis
- Slang words (e.g. ‚sux'; ‚kinda'; etc.)

Vader provides over 9,000 token features, rated on a scale from -4 to 4 (extremely negative to extremely positive). Additionally, the lexicon was extended with a number of ‚Reddit specific terms' like: moon (4.0), stonk (2.5), gtfo (-4.0), hindenburg (-4.0).

The Figure on the left side shows the output of scoring the sentiment after extending the lexicon and cleaning up comments (lowercase, remove line break characters, etc.).

Social media publications
2. Step: Sentiment analysis

Sentiment examples

Sentiment	Ticker	Comment
Positive sentiments		
positive	A	i'm very bullish on a and have been since it was oac. companies like these are the future of healthcare and telemedicine.
positive	AAPL	aapl underpriced compared to peers and ready for upside move.
positive	DD	knock the millionaire next door. should have added more dd to the moon.
Negative sentiments		
negative	A	watch a crashing. hindenburg alert
negative	AAPL	aapl gets slapped!!!!
negative	DD	dd really? ohhhh boy please nooo
Neutral sentiments		
neutral	A	early a was the best a
neutral	AAPL	anyone here thinking opex might be priced in for aapl and this thing could fly?
neutral	DD	one dd and suddenly ketchup hype?

Attribute of RedditData instances:

```
rd.reportdf
```

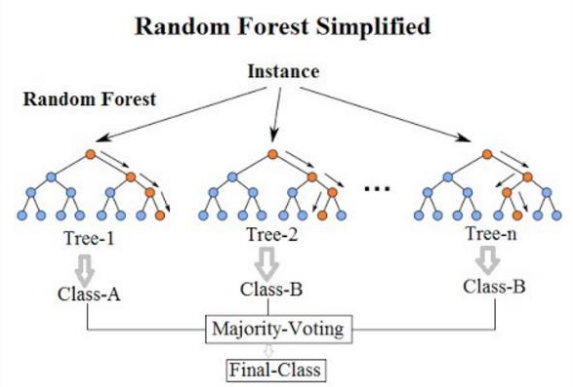
Using compound score for sentiment classification:

Vader classifies the final sentiment according to the so-called ‚compound' score, which is computed by summing the valence scores of each word in the lexicon and normalizes to be beweteen -1 and 1. It could be understood as a ‚normalized, weighted composite score'.

- **Positive sentiment:** $Compound\ score > 0.5$
- **Negative sentiment:** $Compound\ score < -0.5$
- **Neutral sentiment:** $-0.5 \leq compound\ score \leq 0.5$

Machine learning
Why choosing random forests?

Random forest



Source: https://en.wikipedia.org/wiki/Random_forest

Simple classification task:

+1
Stock price will **increase** in 5 days

-1
Stock price will **decrease** in 5 days

Why choosing random forests?

- Invariant under scaling
- Robust to inclusion of irrelevant features
- Produces inspectable models

Why choosing random forests instead of simple classification trees or bagged trees?

- Very deep grown trees tend to learn highly irregular patterns
  → Often leads to overfit the training set, i.e. very high variance, low bias
- Random forests averages multiple trees in order to reduce the variance at the expense of some loss of interpretability.

At each split we consider *m* of *p* features as split candidates. This decorrelates the trees, making the average of the resulting trees less variable.

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{b}(x)$$

Source: Hastie, T., James, G., Tibshirani, R., Witten, D. An Introduction to Statistical Learning, 2013

Machine learning

Model preparation – Data download & feature creation

Stock market data

```
rpred = RedditPredict(startdate = "2020-01-01")
```

```
rpred.AAPL
```

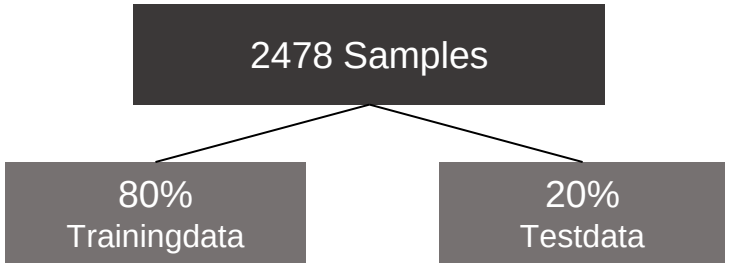
Date	Open	High	Low	Close	Adj Close	Volume
2019-12-31	72.482498	73.419998	72.379997	73.412498	72.552094	100805600
2020-01-02	74.059998	75.150002	73.797501	75.087502	74.207466	135480400
2020-01-03	74.287498	75.144997	74.125000	74.357498	73.486023	146322800
2020-01-06	73.447502	74.989998	73.187500	74.949997	74.071579	118387200
2020-01-07	74.959999	75.224998	74.370003	74.597504	73.723213	108872000
...	...	...	...	...	...	...
2021-07-12	146.210007	146.320007	144.000000	144.500000	144.500000	76299700
2021-07-13	144.029999	147.460007	143.630005	145.639999	145.639999	100698900
2021-07-14	148.100006	149.570007	147.679993	149.149994	149.149994	127050800
2021-07-15	149.240005	150.000000	147.089996	148.479996	148.479996	106820300
2021-07-16	148.460007	149.759995	145.880005	146.389999	146.389999	93100300

389 rows × 6 columns

```
rpred.TrainTestSplit(trainsize = 0.8, info = True)
```

Stock market data:

1. Download stock market data for the 10 most discussed stocks from yfinance library within the RedditPredict class
2. Using ‚TA-Lib‘ library to perform technical analysis and create additional features for every sample (i.e. relative strength index, on balance volume, rate of change, etc.)
3. Combining stock market data from 4. – 10. most mentioned stocks to one large trainingset



Machine learning

Model preparation – Cross validation & model fitting

10-fold cross validation

```
rpred.CrossVal()
```

```
Fitting 10 folds for each of 100 candidates, totalling 1000 fits
```

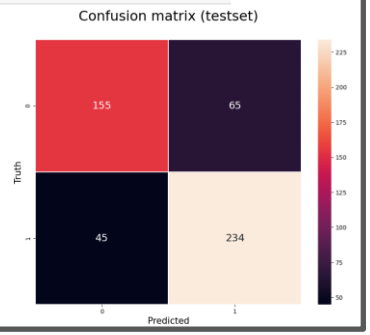
```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=-1)]: Done  17 tasks      | elapsed:    3.2s
[Parallel(n_jobs=-1)]: Done 138 tasks      | elapsed:   15.8s
[Parallel(n_jobs=-1)]: Done 341 tasks      | elapsed:  1.1min
[Parallel(n_jobs=-1)]: Done 624 tasks      | elapsed:  1.8min
[Parallel(n_jobs=-1)]: Done 1000 out of 1000 | elapsed:  3.2min finished
```

```
{'n_estimators': 750, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'log2', 'max_depth': 1500}
```

Model fitting

```
rpred.FitPerfectHyperparam(best_n_estimators = 750,
                           best_min_samples_split = 2,
                           best_min_samples_leaf = 1,
                           best_max_features="log2",
                           best_max_depth=1500,
                           confusion = True)
```

```
Correct predictions:      78.0%
Missclassified samples:   110/499
```



Cross Validation:

I used 10-fold cross validation to test 1,000 different models in order to optimize hyperparameters.

Using the set of hyperparameter for which:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

is lowest!

Model Fitting:

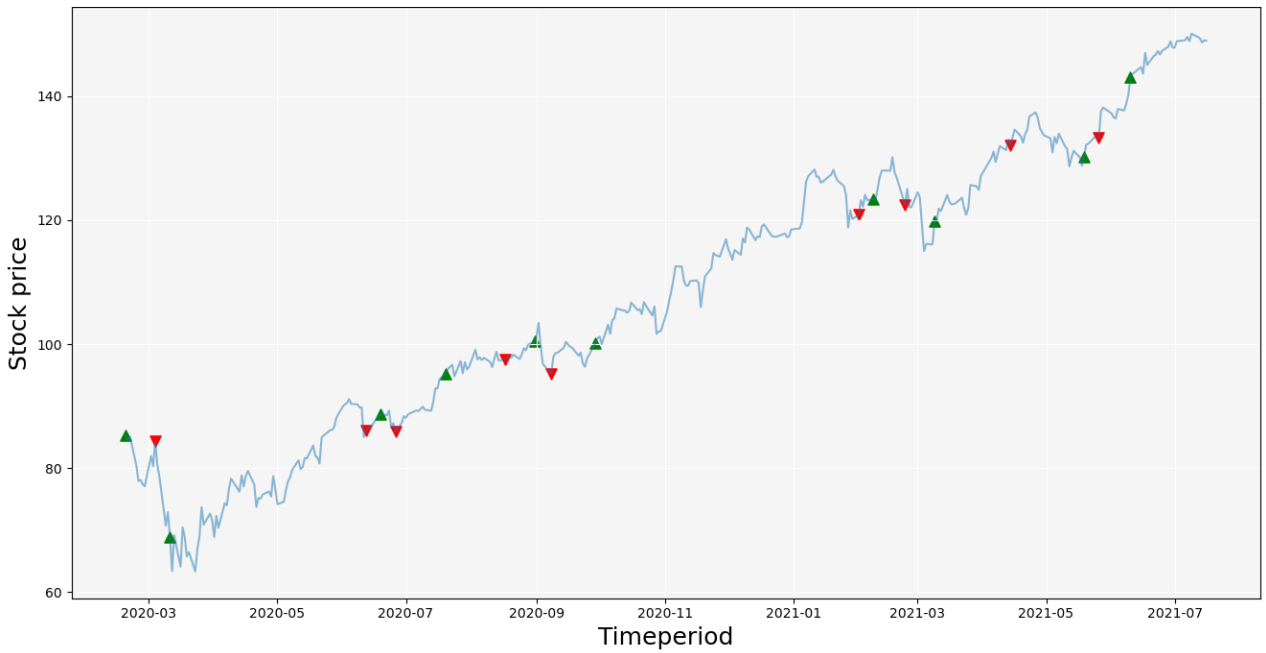
Determining the random forest model with the ‚best' hyperparameter resulting from cross validation leads to an **accuracy of over 75%**!

Machine learning

Model preparation – Early signal model

Early signal model

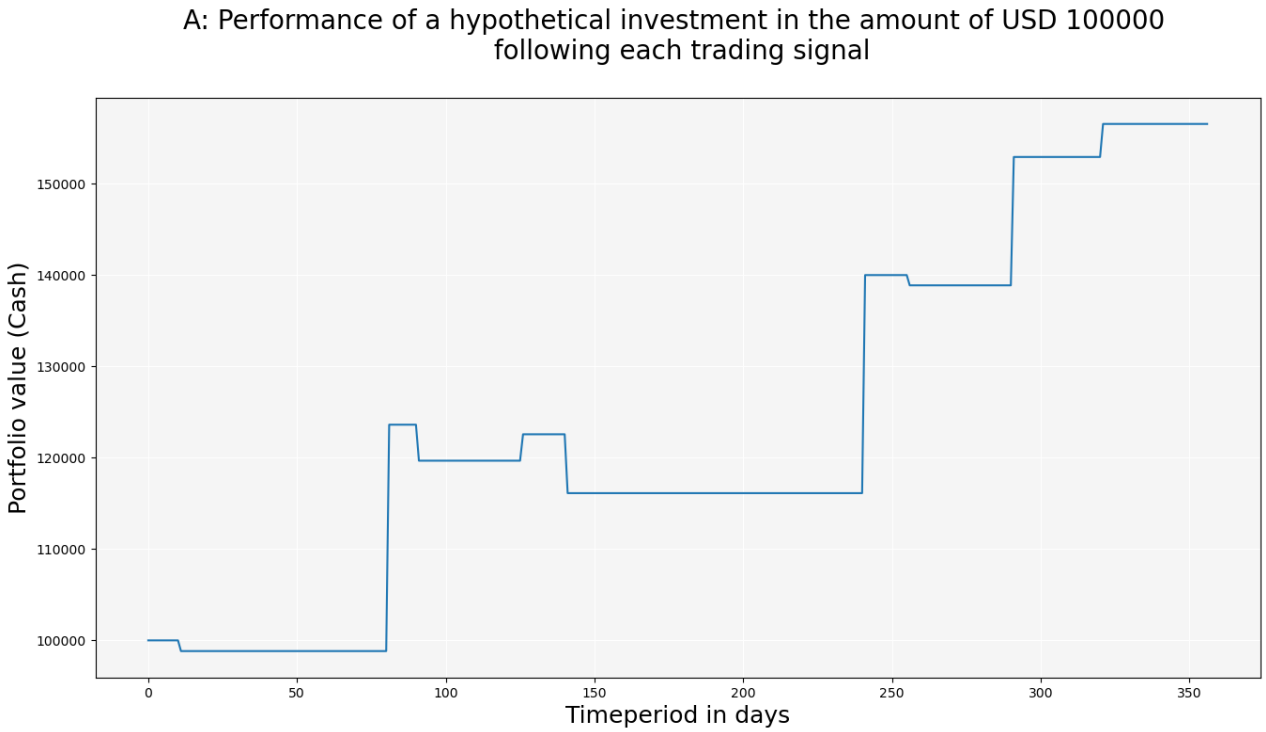
A: Stock price development incl. early signals from random forest algorithm



Machine learning

Model preparation – Early signal model

Development of a hypothetical investment in the amount of USD100.0k



OUTPUT AND CONCLUSION

Output and conclusion

Early signal model works pretty well. Adding sentiment data to the machine learning process would require months of sentiment analysis beforehand.

Reconciliation of machine learning output vs. sentiment analysis results

Ticker	Sentiment	Random forest prediction*
Top three discussed stocks from Reddit		
A	positive	1
AAPL	positive	1
DD	neutral	-1

** 1 = Increasing stock price; -1 = decreasing stock price*

Monitored sentiments + predictions over 10 days.

Overall, the sentiment from the most discussed stocks on Reddit serves as a good indicator for an intuition whether the stock price is more likely to increase or decrease.

Conclusion

This research is fully applicable within two python classes and provides the user with information about the most mentioned stocks on Reddit and their sentiment as well as a prediction for near-term performance (incl. early signal model). The data from the RedditData class can serve as a good initial guide for stock picking

In order to show truly valid evidence for the statistical significance of sentiment data, the sentiment data would have to be collected on a daily basis over a period of several months. However, for the time frame of this project, that would be impossible.

Thanks for your attention!