AAPL Stock Price Prediction

Capstone Project #1
Springboard Data Science Career Bootcamp

Mei Liu

Outline

- Introduction/Objective
- Data Wrangling
- Exploratory Data Analysis
- Building/Testing Prediction Models
- Final Results

Introduction

Stock Prices are **unpredictable**:

- **Efficient Market Hypothesis**:
  - Weak form
  - Semi-strong form
  - Strong form

- **Random Walk Theory**

Stock Prices may be **somewhat predictable**:

- **Behavioral Finance**

Objective

Using a 10 day prediction window on AAPL closing stock price:

- Compare the prediction accuracy of different models
  - **Simple moving average, Auto ARIMA, FB Prophet, XGBoost**
- Determine which features are most important in the prediction

Data

- **Sample Period**: Jan 2013 – Dec 2018
- **Stock market data** from CRSP database
  - AAPL: (*daily*) Open, Close, Bid, Ask, Vol
  - AAPL: (*once per quarter*) dividend announcement/payout
  - S&P500: (*daily*) Return
- **Earning's announcement surprise data** from StreetInsider.com (*once per quarter*)
- **Google trend data** from PyTrends (*monthly*)

Data Wrangling

Data Wrangling Steps:

1. **Data Type**
   - Date as DateTime, other variables as int64 or float64

2. **Missing Variables**
   - DCLRDT: dummy variable where =1 on the declaration date and =0 otherwise
   - DIVAMT: null values are set to 0

3. **Variable Adjustments**
   - All relevant variables adjusted for the 7-1 stock split on June 9, 2014

4. **Feature Creation**

Feature Creation *(for EDA)*

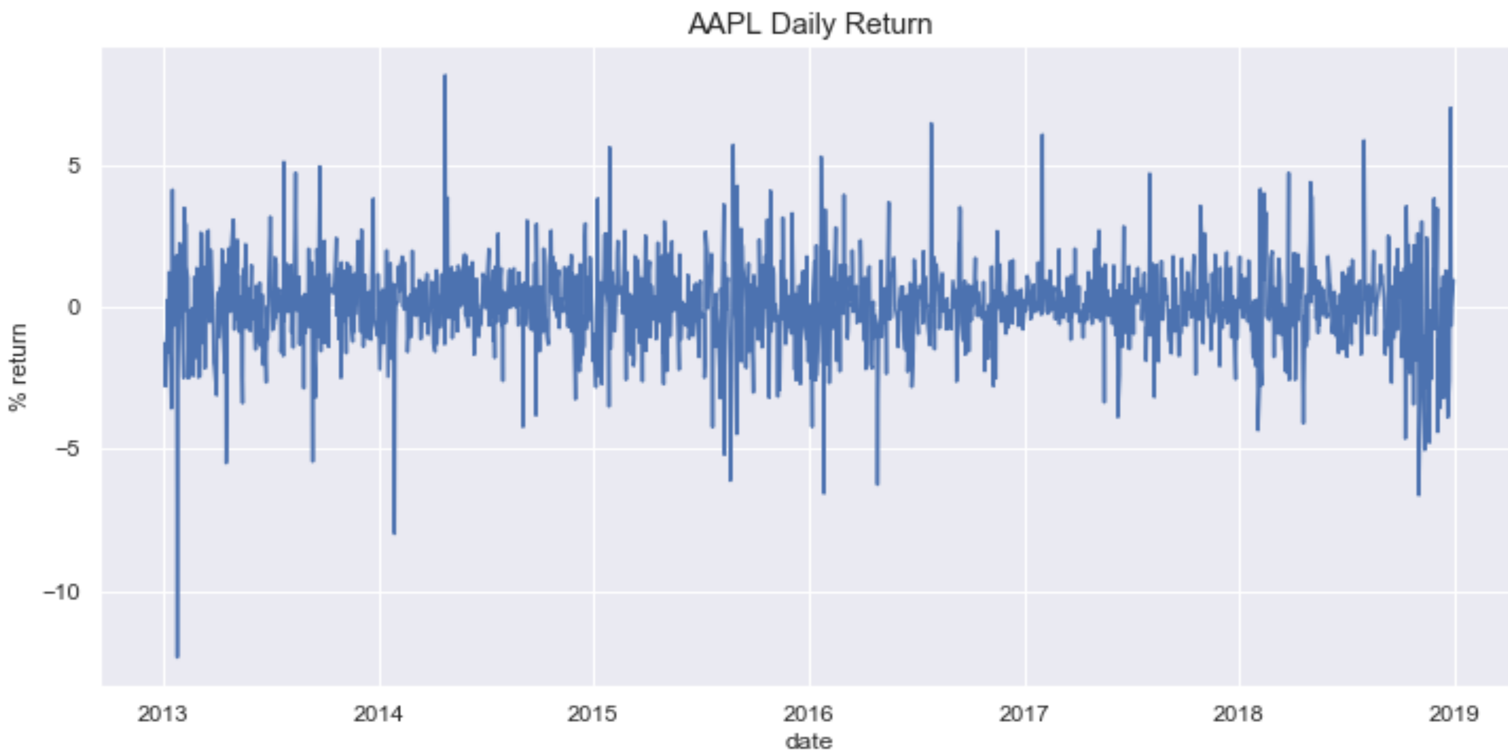
- **Stock Data**
  - Day, month and year for each date
  - Bid ask spreads
  - Difference between the opening and closing price
- **Google Trend Data**
  - Previous month's '**TREND**' is merged to current month in the stock data (e.g. 02/05/2011 trading day has 'TREND' value from 01/2011)
- **Earning Announcement Data**
  - '**Announcement**' variable: where announcement day = 30, and decays by 1 until *Announcement* = 0
  - '**Surprise**' variable: percentage earnings surprise, forward-filled

EDA

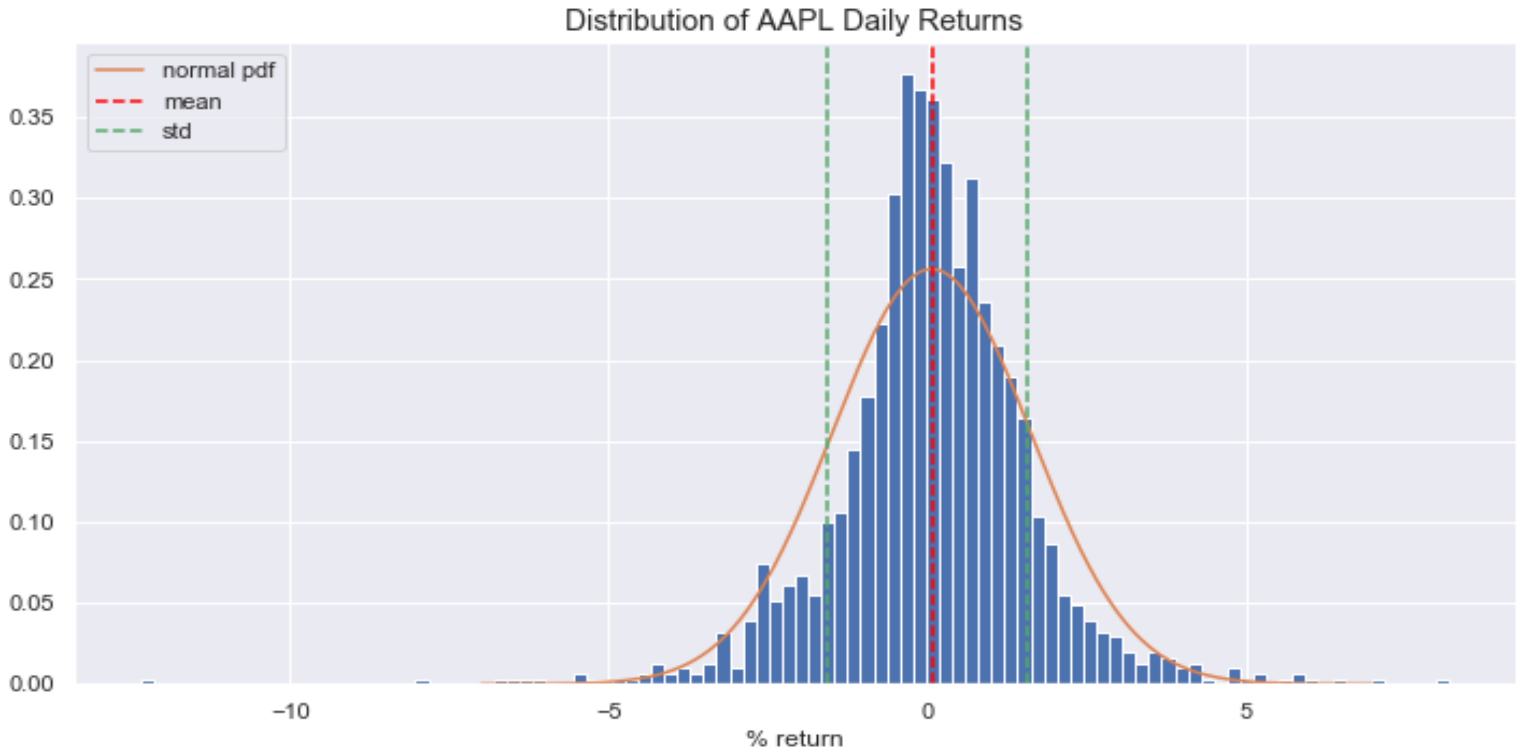
How does the stock **price change over time**?

EDA *(cont.)*

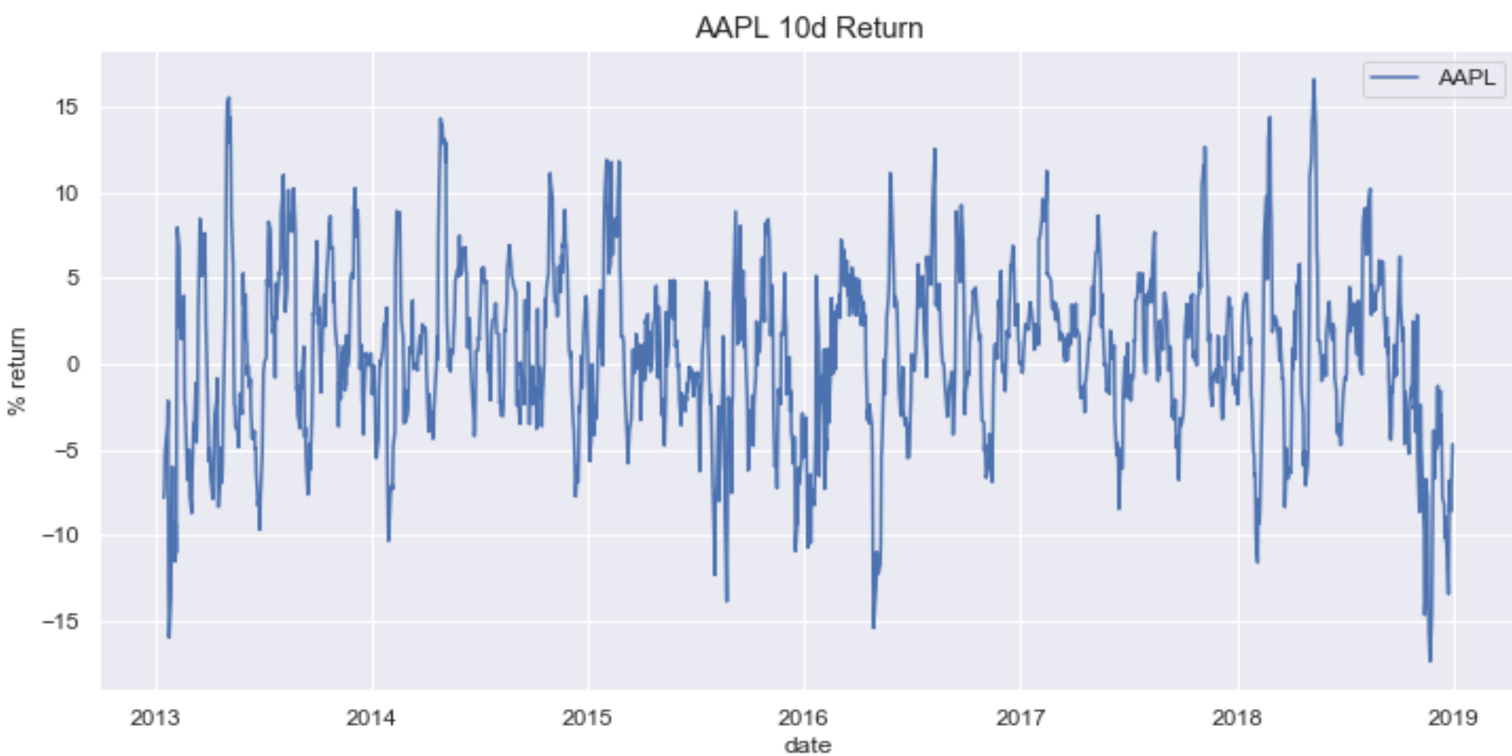
What does the **distribution of historical returns** look like?



EDA *(cont.)*

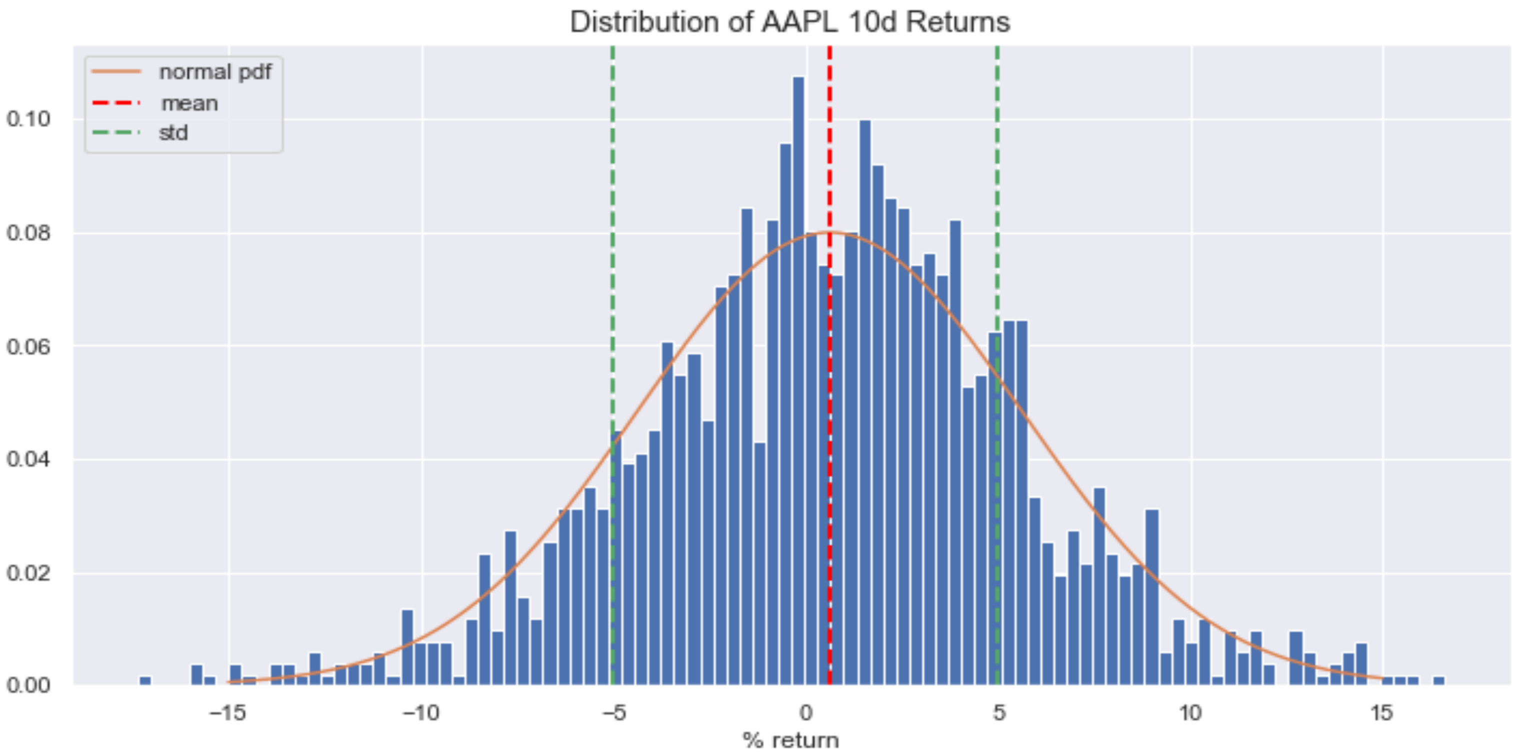

Distribution of AAPL Daily Returns
normal pdf
mean
std
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−10
−5
0
5
% return

EDA *(cont.)*



AAPL 10d Return
AAPL
15
10
5
0
−5
−10
−15
% return
2013
2014
2015
2016
2017
2018
2019
date

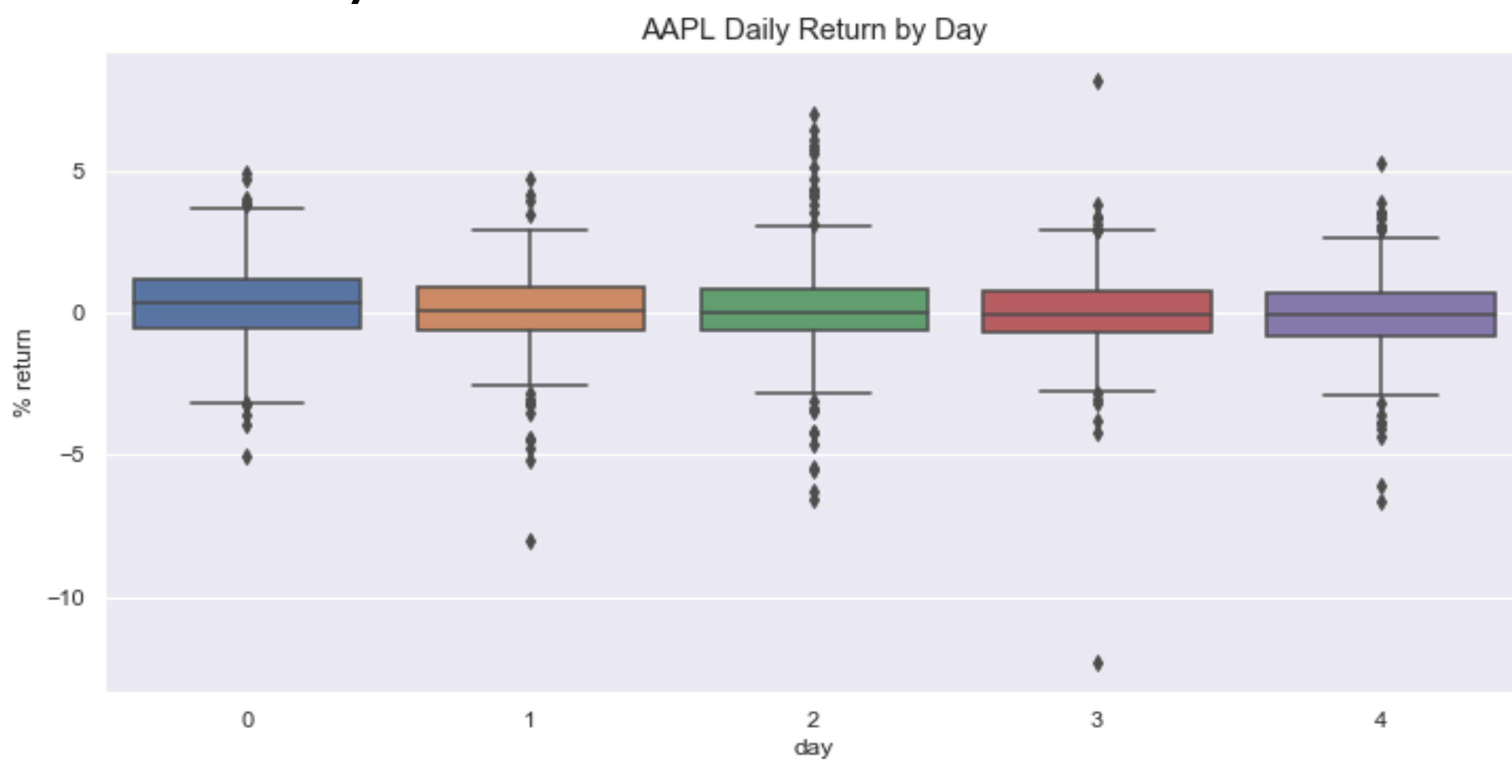
EDA *(cont.)*



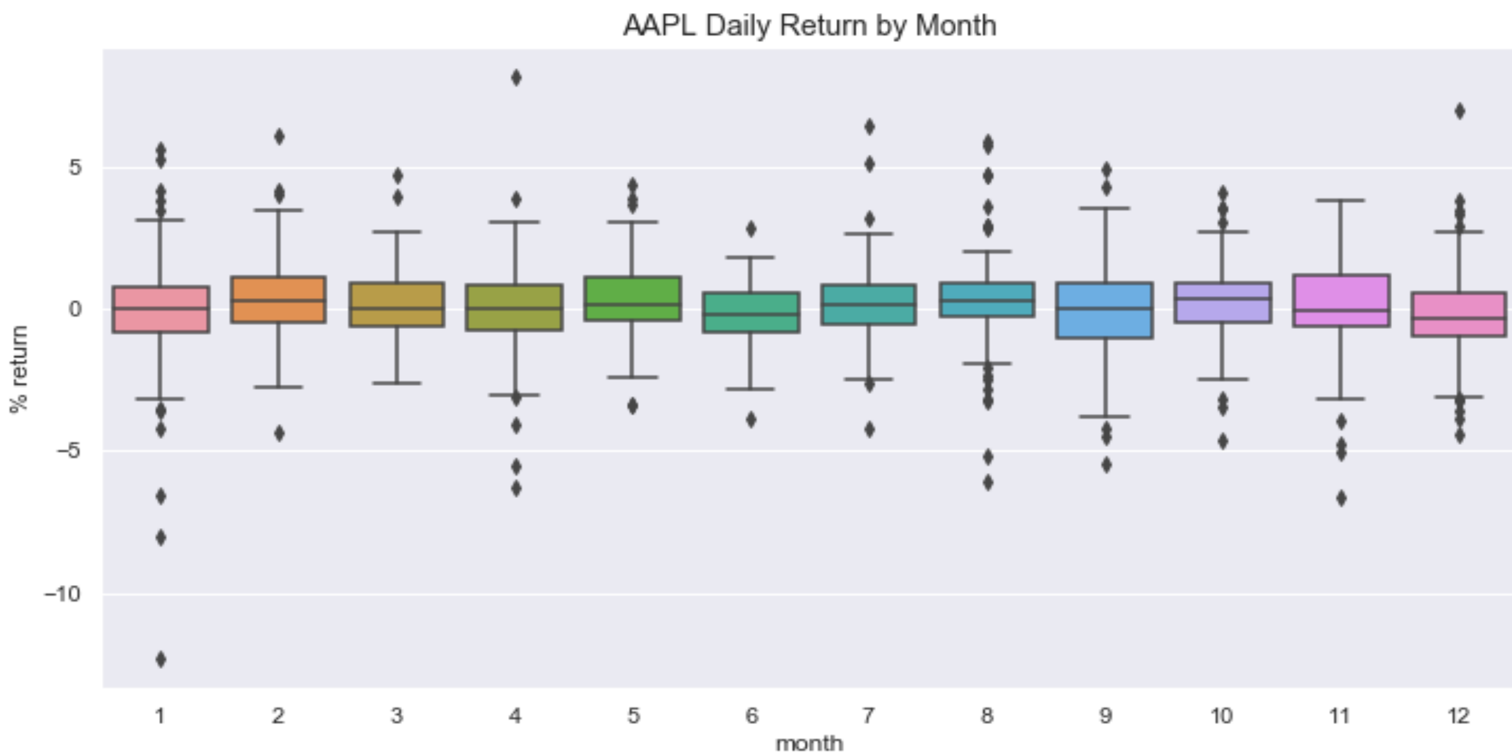
Distribution of AAPL 10d Returns
normal pdf
mean
std
0.10
0.08
0.06
0.04
0.02
0.00
−15
−10
−5
0
5
10
15
% return

EDA *(cont.)*

Does the stock return differ based on the **day of the week** or the **month of the year**?



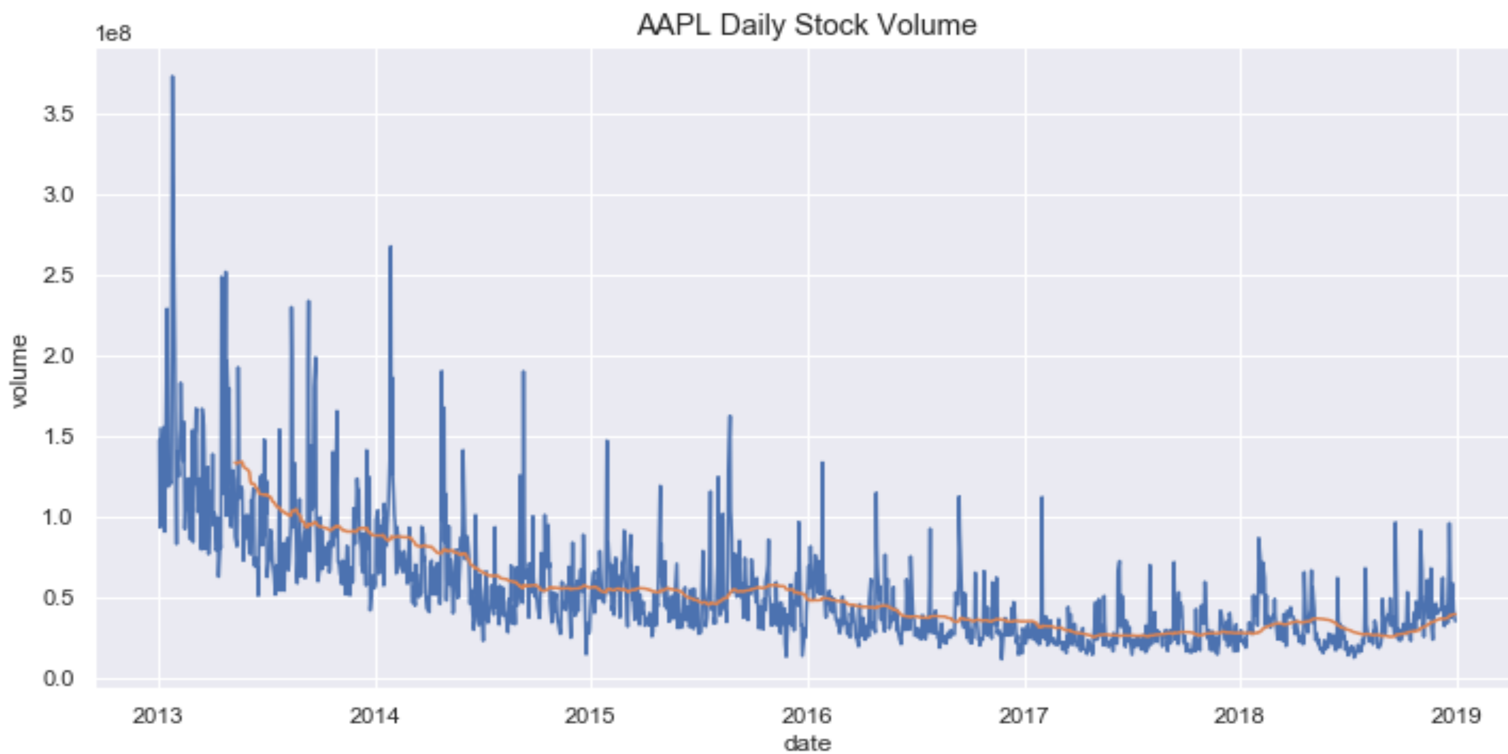
EDA *(cont.)*



AAPL Daily Return by Month
% return
5
0
−5
−10
1
2
3
4
5
6
7
8
9
10
11
12
month

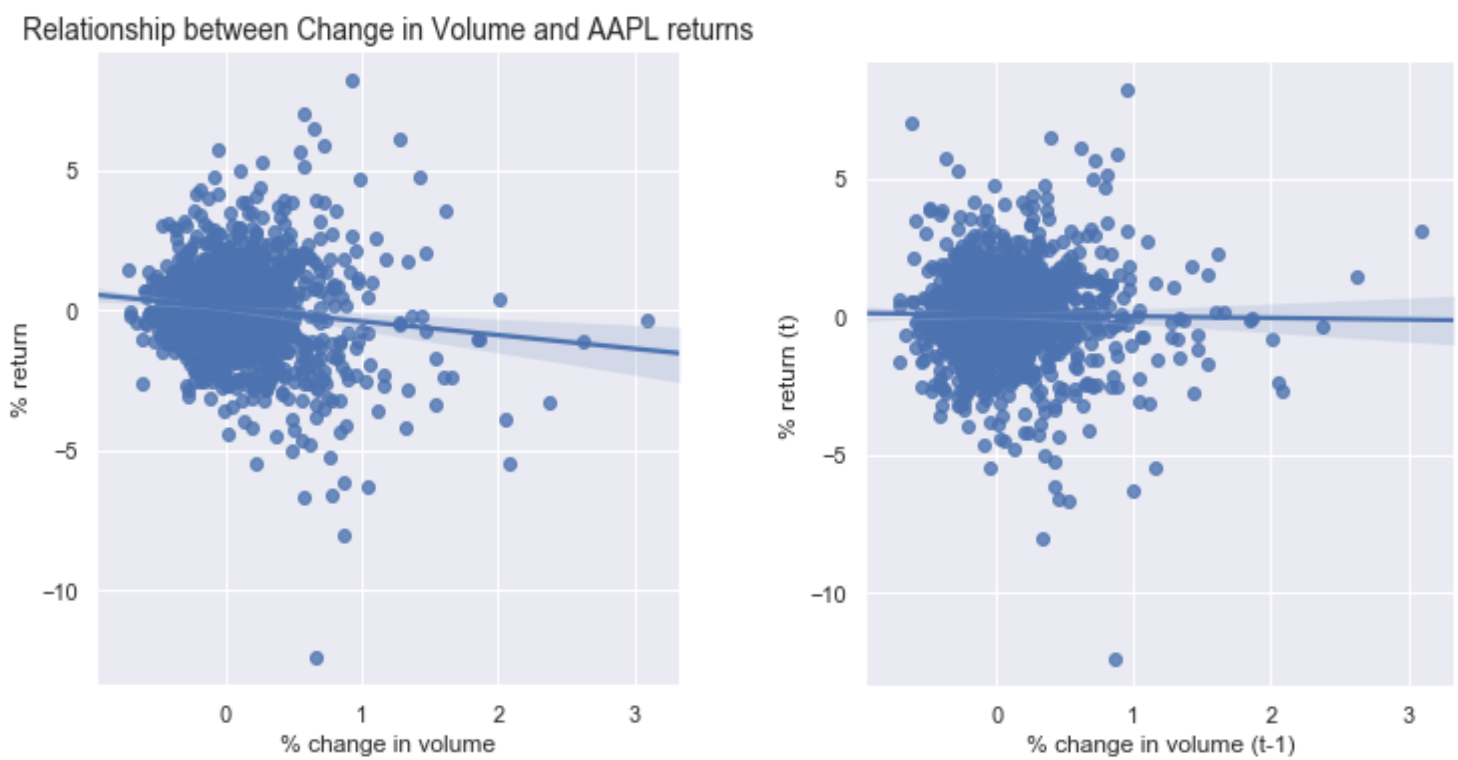
EDA *(cont.)*

How does the daily stock **volume change over time**?



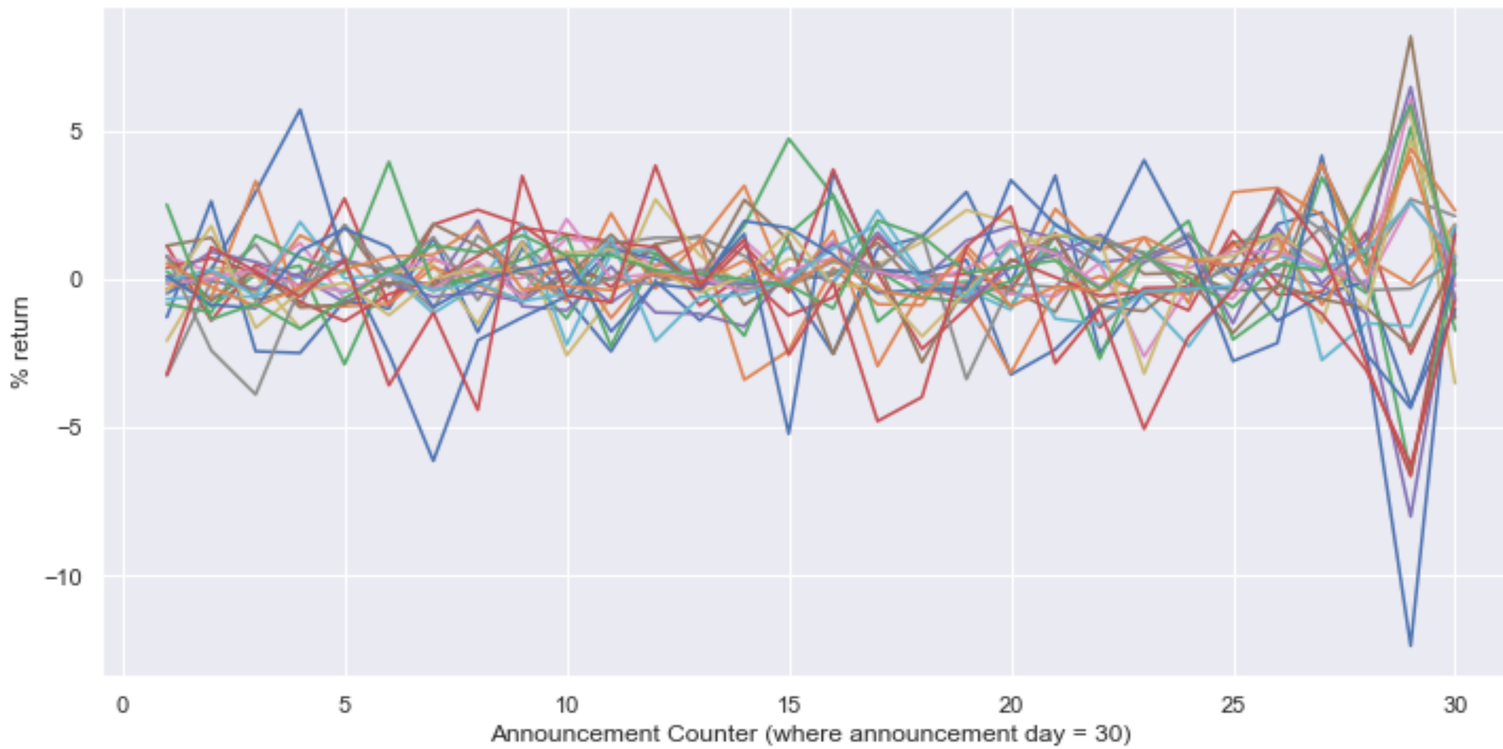
EDA *(cont.)*

Is there any **relationship** between the **change in volume** and **returns**?



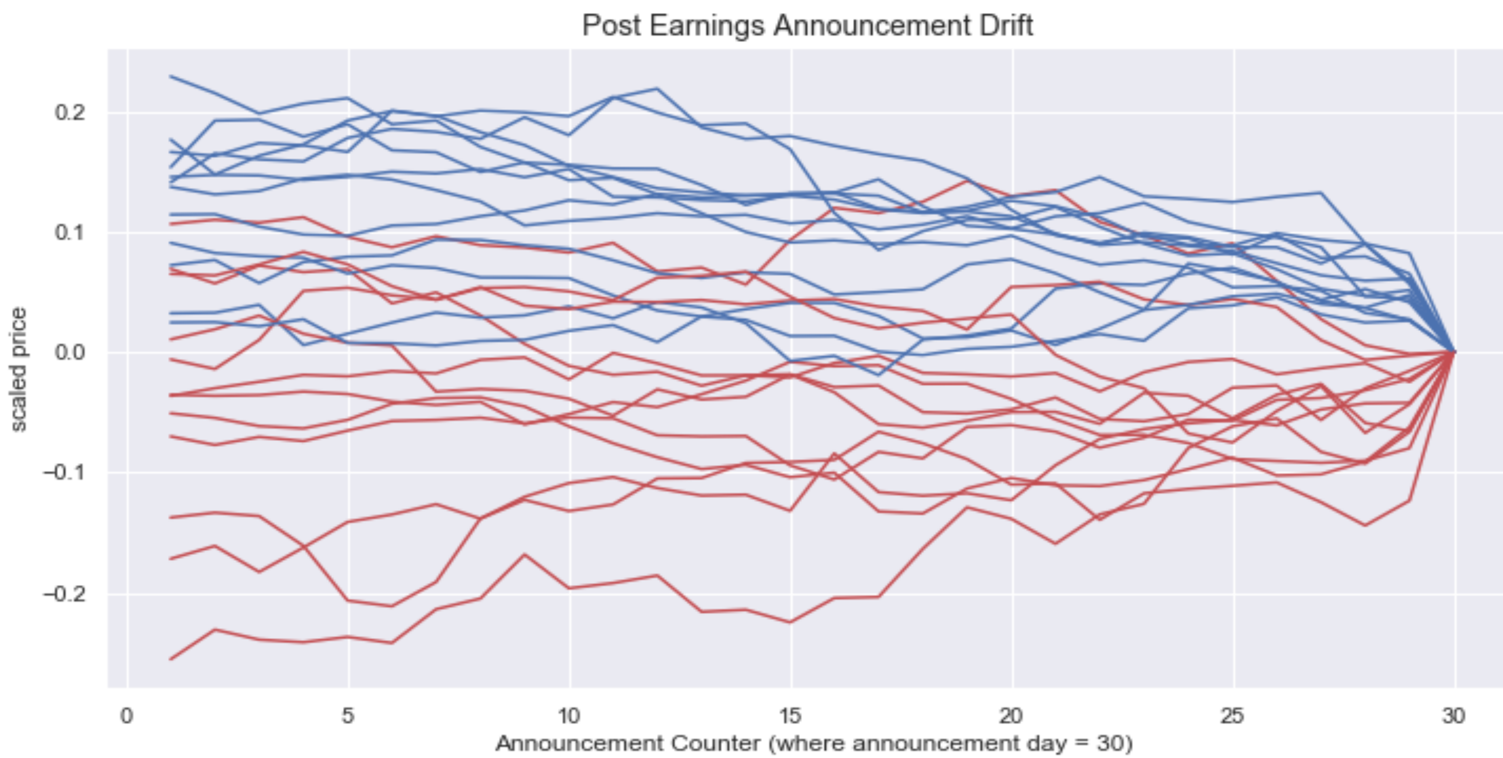
EDA *(cont.)*

What effect does the **earnings announcement** have on **stock returns**?



EDA *(cont.)*

Is there **post earnings announcement drift** (PEAD)?



Prediction Models

- **Historical Price based models:**
  - Simple moving average, Auto ARIMA, FB Prophet, XGBoost
- **Multiple variable based models:**
  - XGBoost
- Evaluate prediction accuracy using **RMSE**

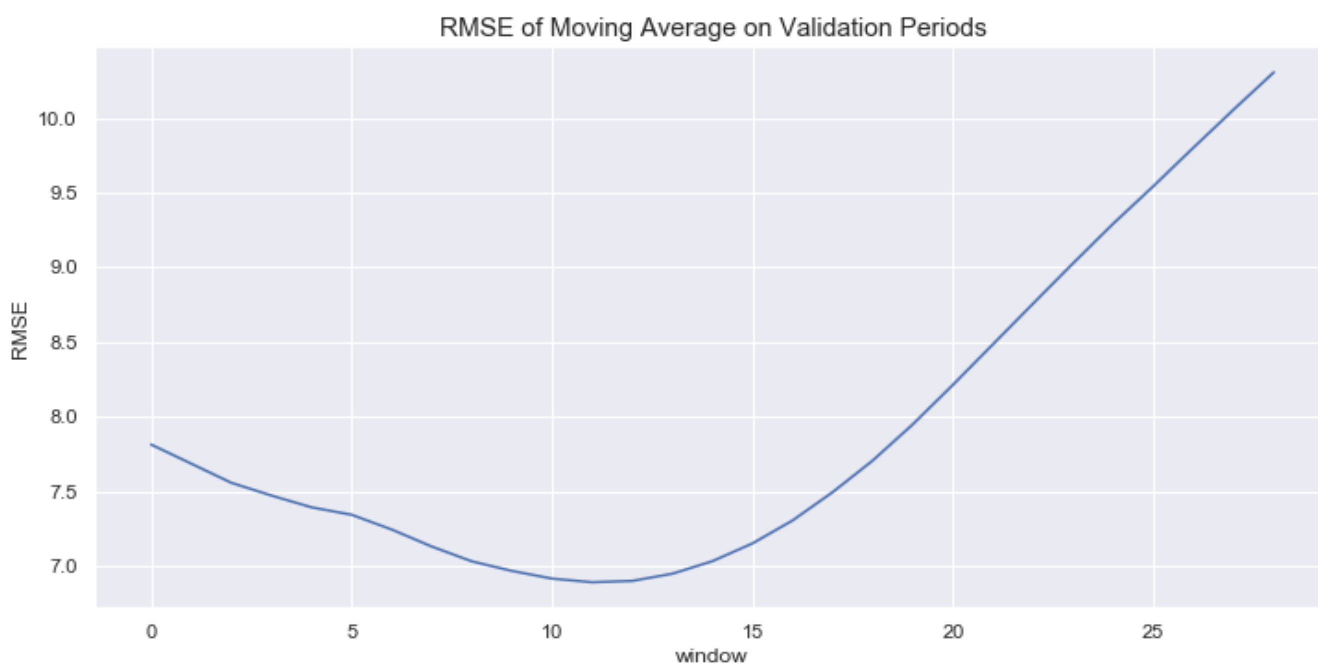
Train, Validation, Test Split

- **Test Set:** last 10 trading days in sample
- **Training Set:** sample not including test data
- **Validation Set:**
  - split training data into 10 evenly spaced folds
  - set aside the last 10 trading period of each fold as the validation sets
  - Not applicable to Auto ARIMA, FB Prophet models

Train, Validation, Test Split *(cont.)*

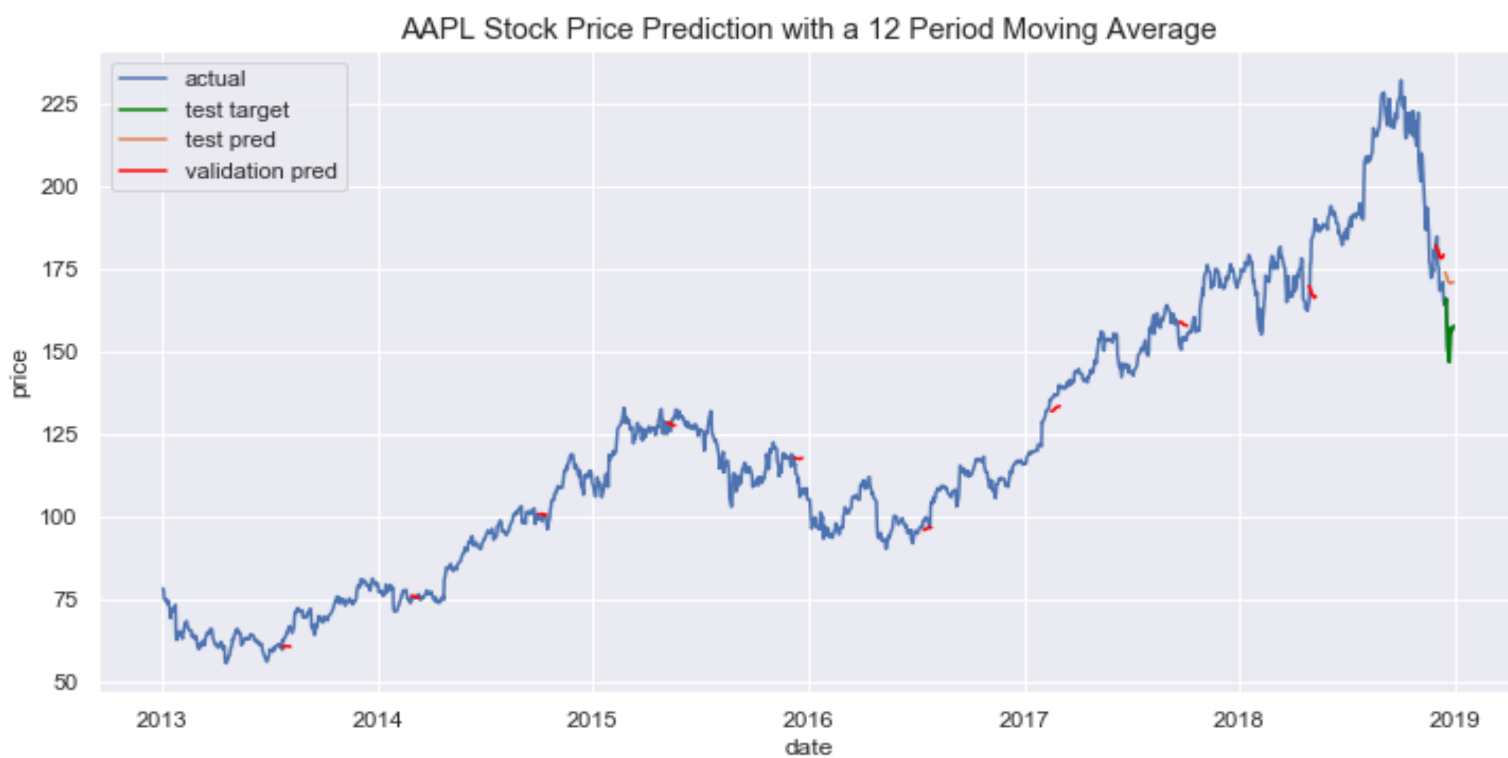
Model 1: Simple Moving Average

- Predicted Price_t = Avg($\text{Price}_{(t-10,\ t-1)}$)
- **Rolling window of 12** has the lowest validation RMSE



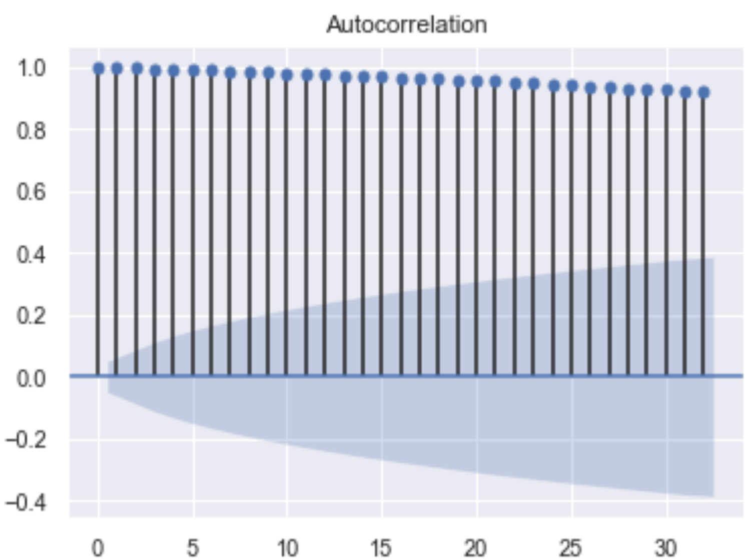
Simple Moving Average *(cont.)*

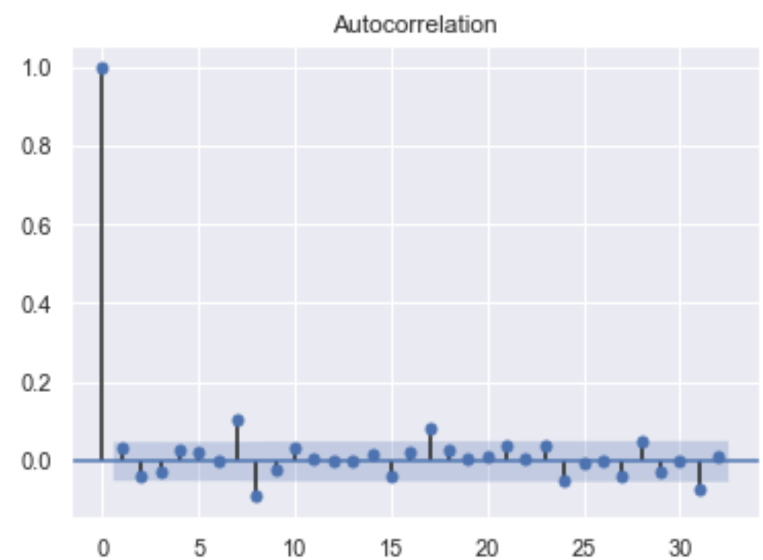
- Using rolling window=12, **RMSE** of the test set is **15.26**



Model 2: Auto ARIMA

- Using **auto_arima** model from the **pmdarima** package
- Use ndiffs, nsdiffs to first approximate the **d** and **D parameters** in the model (**1, 0 respectively**)





Auto ARIMA *(cont.)*

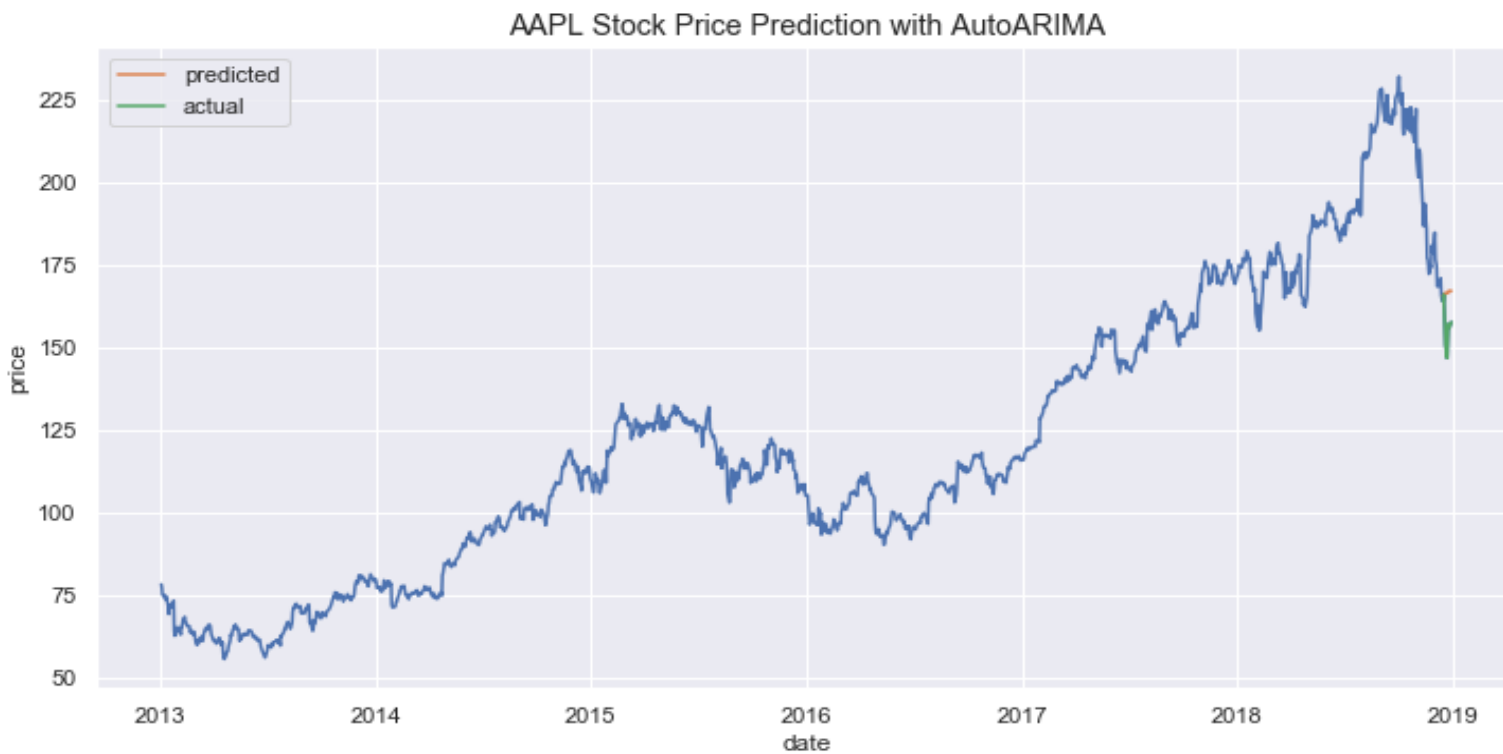
- Auto ARIMA model uses **only historical closing price** from the training data **to make predictions**
- Searches over possible models and **returns the best ARIMA model based on AIC/BIC**

```
from pmdarima import auto_arima

model = auto_arima(train_data.PRC, start_p=1, start_q=1,max_p=5, max_q=3, m=12,start_P=0,
                  seasonal=True,d=1, D=0, trace=True,error_action='ignore',suppress_warnings=True
)
```

Auto ARIMA *(cont.)*

- **RMSE** of the test set is **10.86**



Model 3: FB Prophet

- **fbprophet** model from **Prophet** package
- **Additive regression model** where non-linear trends are fit with **yearly**, **weekly**, and **daily seasonality**, plus **holiday effects**
- fbprophet model uses **only historical closing price** from the training data **to make predictions**

FB Prophet *(cont.)*

- **RMSE** of the test set is **51.94**

Model 4: XGBoost w/ Price Features

- Extreme Gradient Boosted Decision Tree Algorithm
- **XGBRegressor model** from **xgboost** package
- Steps:
  1. **Create features**
  2. **Tune parameters** using training/validation set
  3. **Fit model** and **predict test values**
  4. **Plot feature importance**

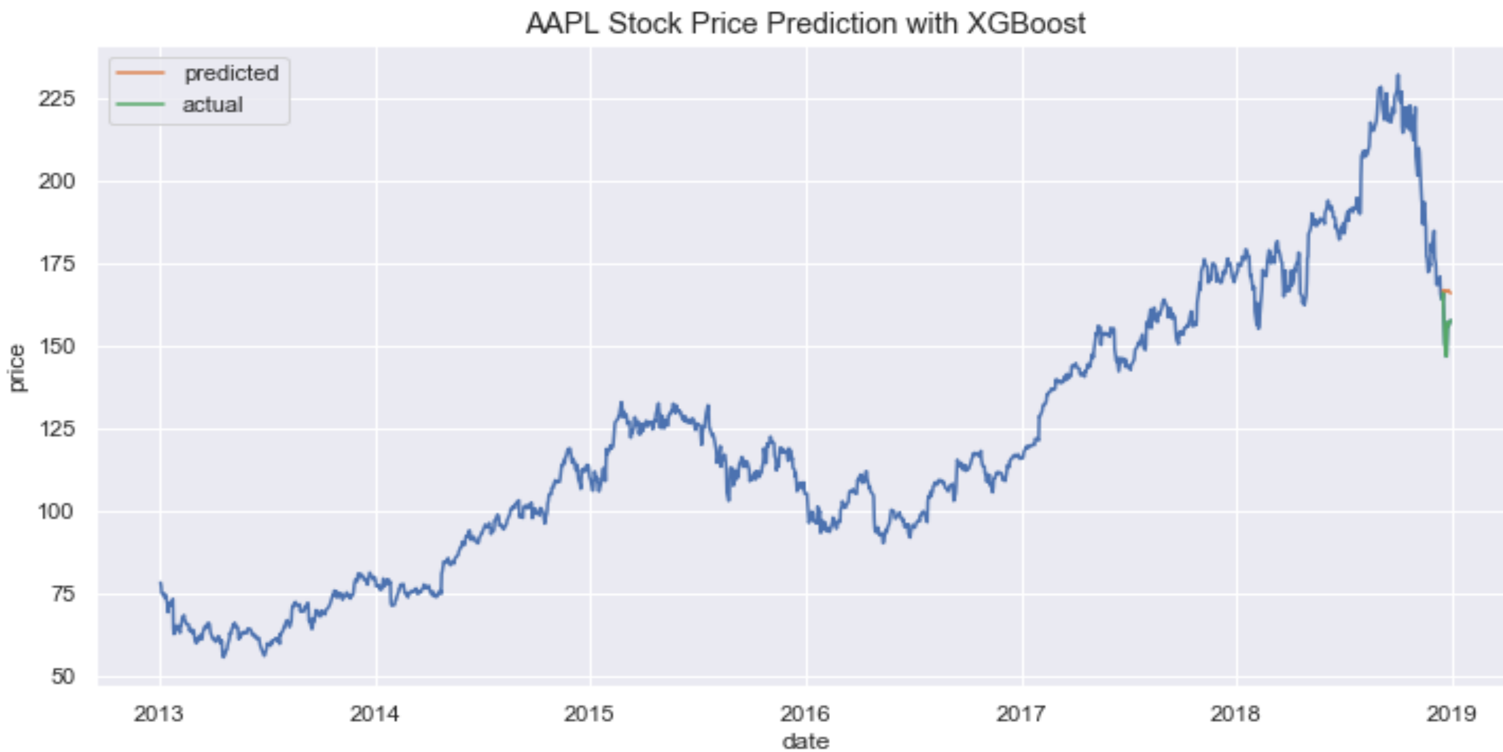
XGBoost w/ Price Features *(cont.)*

1. Create features using the date and closing price
   - **Date features**
     - month, 'qtr’, year', 'day of the week’, day of the month, day of the year, day number, start/end of the week
   - **Lag features**
     - Moving mean, median, min, max
       - *Short term lags: 5,10,15*
       - *Long term lag: 50 (only for max and min*)
     - Apart from mean, features flatly extended to test data
   - **Price encodes**
     - Mean of week, month, year

XGBoost w/ Price Features *(cont.)*

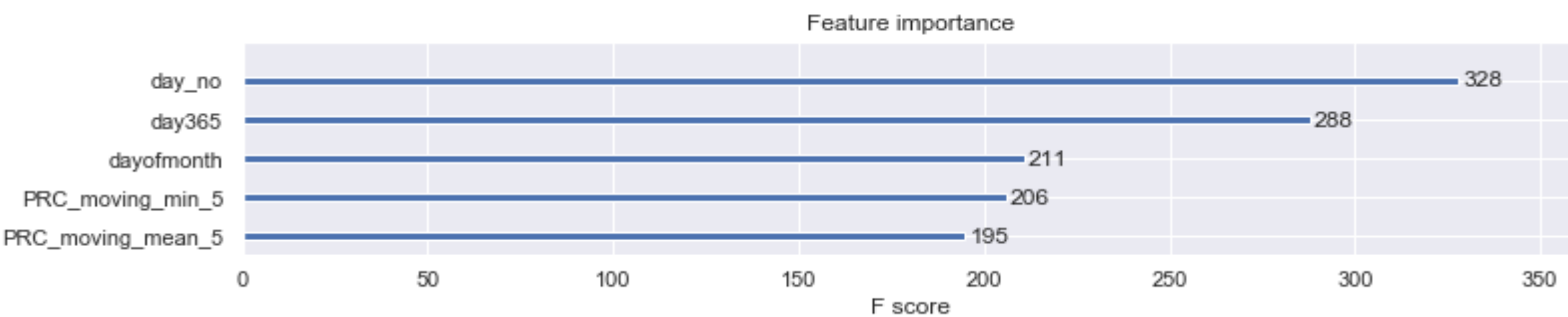
2. **Sequentially tune parameters** to find the best model based on validation error

3. **RMSE** of the test set is **10.74**



XGBoost w/ Price Features *(cont.)*

4. **Top 5 most important features:**
   1. Day number
   2. Day of the year
   3. Day of the month
   4. Short term moving min
   5. Short term moving mean

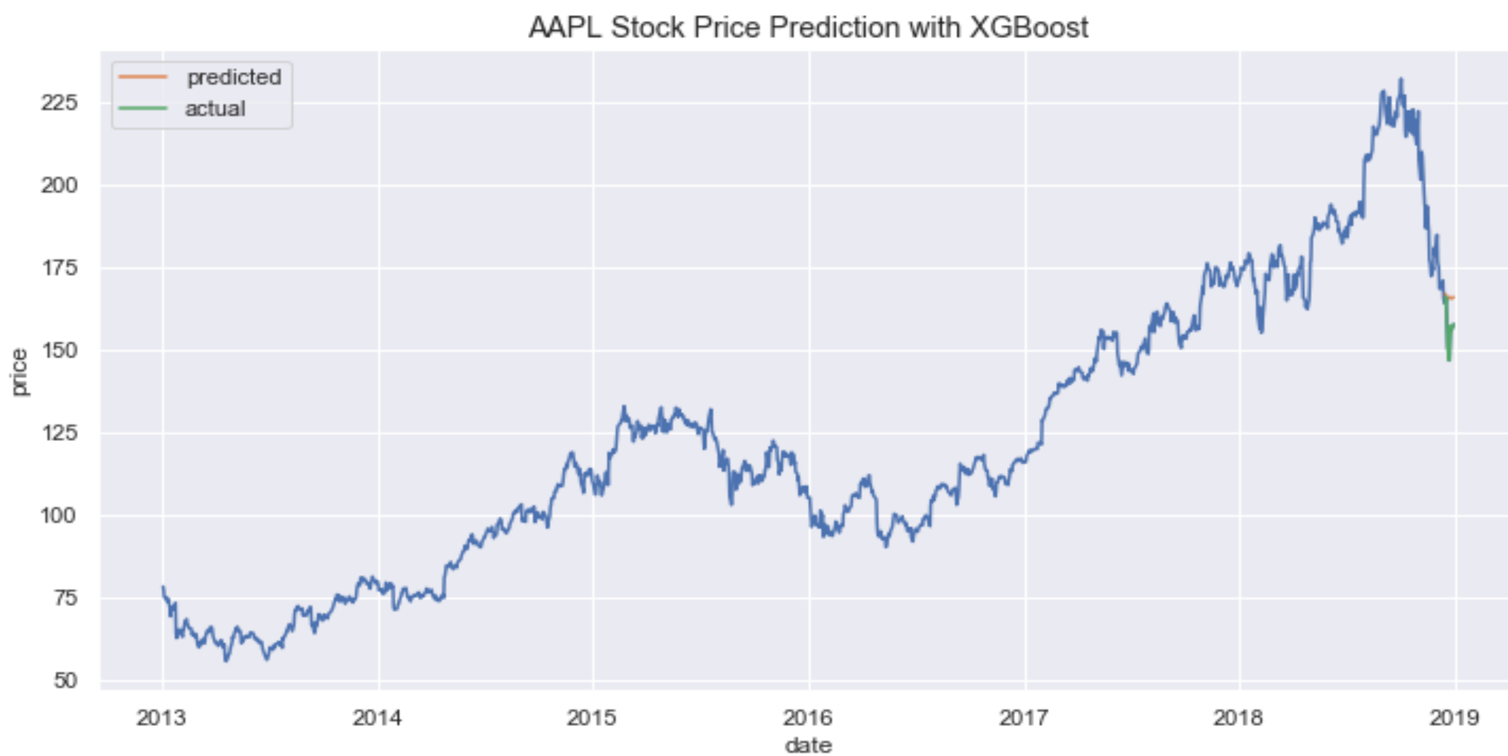


Model 5: XGBoost extended

- Same steps as previous XGBoost model, adding **additional lag features**
- In addition to lag variables on closing price, create lag features for:
  - **Volume, number of trades, S&P500 daily return, bid-ask spreads, difference between open and close prices**

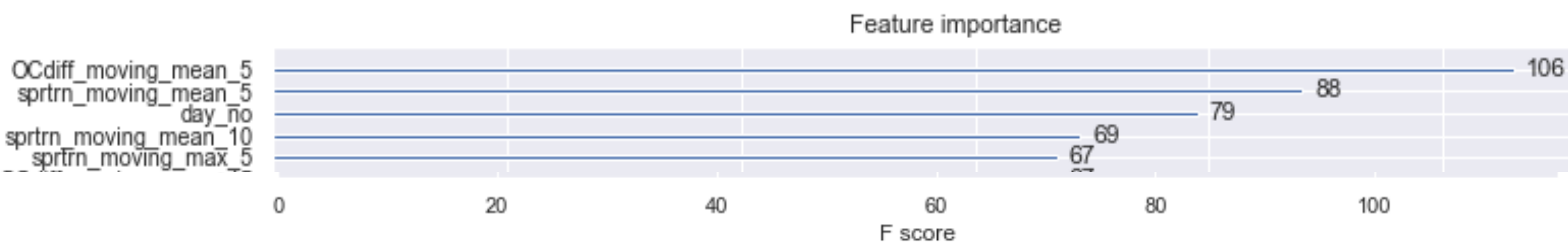
XGBoost extended *(cont.)*

- **RMSE** of the test set is **10.07**



XGBoost extended *(cont.)*

- **Top 5 most important features:**
  1. Short term mean difference between open and close prices
  2. Short term mean (5 day) S&P500 daily return
  3. Day number
  4. Short term mean (10 day) S&P500 daily return
  5. Short term max S&P500 daily return



Final Results

Comparison of different models:

	Model	RMSE
1	Simple Moving Avg	15.26
2	AutoARIMA	10.86
3	FB Prophet	52.03
4	XGBoost (price based)	10.74
5	XGBoost (multiple features)	10.07