Pandemics vs stock market

Austin Roberto Isaac

Pandemics vs stock market

Authors: Roberto, Austin, Issac

Intro
- Analyzed past pandemics to compare to current covid-19 pandemic
- Predicted buy/sell stocks during pandemics

Methods
- Machine learning models used: Linear Regression, KNN, SVC and Random Forest
- ARIMA model
- Moving Averages
- Data Visualization

Result
- Uncertain prediction due to many factors
- Ensemble learning increases probability of prediction being correct
- Ex. JnJ during SARS model predicted

Team Sigmoid Frontsquat

With the help of **data** from **stock** numbers during past pandemics **machine learning** can **help** you make a data driven decision to **buy, hold** or **sell** during **Covid-19**

Model Results
Apple

Nike

Johnson & Jonson

Ford

Alaska Airlines

Chevron

American Express

Data sets and collecting data

- Downloaded manually from Yahoo Finance only 100 days
- Kaggle
- stock market data vs. Covid-19 data, problems integrating the two

Extracting S&P 500 ticker symbols

- We also used BeautifulSoup to scrape the web for the ticker symbols for all 500 companies on the S&P 500 as a pickle object

```
def save_sp500_tickers():
    response = requests.get('https://en.wikipedia.org/wiki/List_of_S%26P_500_companies')
    soup = bs.BeautifulSoup(response.text)
    table = soup.find('table',{'class':'wikitable sortable'})
    tickers = []
    for row in table.findAll('tr')[1:]:
        ticker = row.findAll('td')[0].text.replace('\n','')
        tickers.append(ticker)
    with open('sp500tickers.pickle','wb') as f:
        pickle.dump(tickers,f)
    # print(tickers)
    return tickers

save_sp500_tickers()
```

Pandas datareader

- Used ticker symbols to get financial data from Yahoo Finance
- Requested data starting from January 1, 2000 - today
- API call to pandas_datareader

```
for ticker in tickers:
    print(ticker)
    if not os.path.exists('stock_dfs/{}.csv'.format(ticker)):
        try:
            df = web.DataReader(ticker,'yahoo',start,end)
            df.to_csv('stock_dfs/{}.csv'.format(ticker))
        except KeyError:
            pass
    else:
        print('Already have {}'.format(ticker))
```

Integrating COVID-19 data with financial data?

- At first we thought we could regress stock price on COVID-19 confirmed cases or deaths
- We scrapped this idea very quickly and decided to just try and investigate data during strategically picked dates
- We used the CDC's fact sheets and time lines to find information on other pandemics
- SARS
    - The start/end dates we used to subset our larger dataset October 2003 - January 2004
    - 8,098 cases 774 deaths
- H1N1 - SWINE FLU!
    - Lasted from April 2009 until August 2010
    - 60 million cases 12,469 deaths
    - Notably around same time of financial crisis
- Ebola
    - Lasted from December 2013 until January 2016
    - 28,816 cases and 11,310 deaths

Data preprocessing

- Subsetting our data by company
  - We have data from 500 companies dating back 20 years
  - We chose a handful of companies that we are interested in and represent the economy
  - Apple, American Express, Nike, Chevron, Johnson & Johnson, Ford, Alaska Airlines
- Subsetting our data by pandemic
  - We also subset of data by pandemic in order to look at performance during each pandemic
- Adjusted close
  - Adjusted closing price accounts for stock splits and is the best measure
- Put all stocks we are interested in into one data frame

Percent change as a metric for buy/sell/hold

- This function creates new columns for the percent change over the course of a week
- Fills the last few days with zeros because its undefined
- This is the main metric for our machine learning classification

```
def process_data_for_labels(pandemic,ticker):
    # how many days
    days = 7
    df = pd.read_csv('stock_dfs_{}/{}.csv'.format(pandemic,pandemic), index_col=0)
    # df = pd.read_csv('sars_data.csv', index_col=0)

    tickers = df.columns.values.tolist()
    df.fillna(0,inplace=True)

    for i in range(1, days+1):
        df['{}_{}d'.format(ticker,i)] = (df[ticker].shift(-i) - df[ticker]) / df[ticker]
    df.fillna(0,inplace=True)
    return tickers,df
```

Extracting feature sets for buy/sell/hold

```python
def buy_sell_hold(*args):
    cols = [c for c in args]
    requirement = .02
    for col in cols:
        if col > requirement:
            return 1
        if col < -requirement:
            return -1

    return 0
```

```python
def extract_feature_sets(pandemic,ticker):
    tickers, df = process_data_for_labels(pandemic,ticker)
    hm_days=7
    df['{}_target'.format(ticker)] = list(map( buy_sell_hold, df['{}_1d'.format(ticker)],
        df['{}_2d'.format(ticker)],
        df['{}_3d'.format(ticker)],
        df['{}_4d'.format(ticker)],
        df['{}_5d'.format(ticker)],
        df['{}_6d'.format(ticker)],
        df['{}_7d'.format(ticker)]
        ))
    vals = df['{}_target'.format(ticker)].values.tolist()
    str_vals = [str(i) for i in vals]
    print('Data spread: ',Counter(str_vals))
    df.fillna(0,inplace=True)
    df = df.replace([np.inf, -np.inf], np.nan)
    df.dropna(inplace=True)
    df_vals = df[[ticker for ticker in tickers ]].pct_change()
    df_vals = df_vals.replace([np.inf, -np.inf],0)
    df_vals.fillna(0, inplace=True)
    X=df_vals.values
    y=df['{}_target'.format(ticker)].values
    return X,y
```

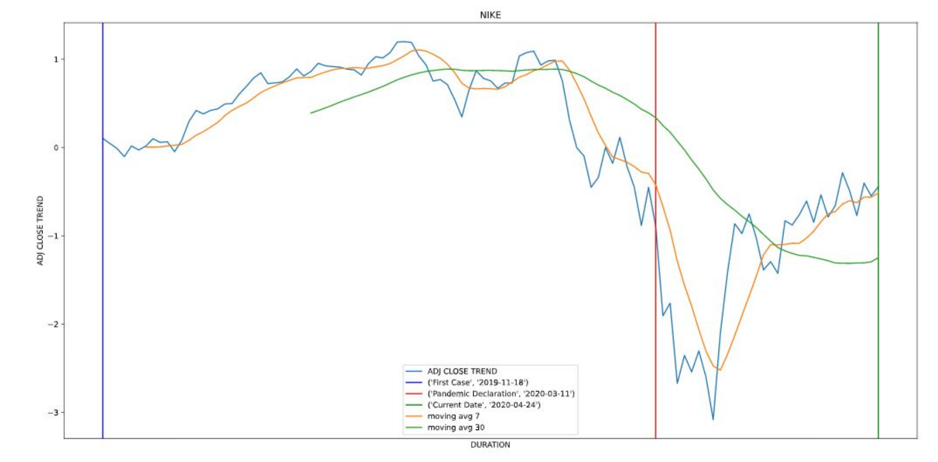
Machine learning models

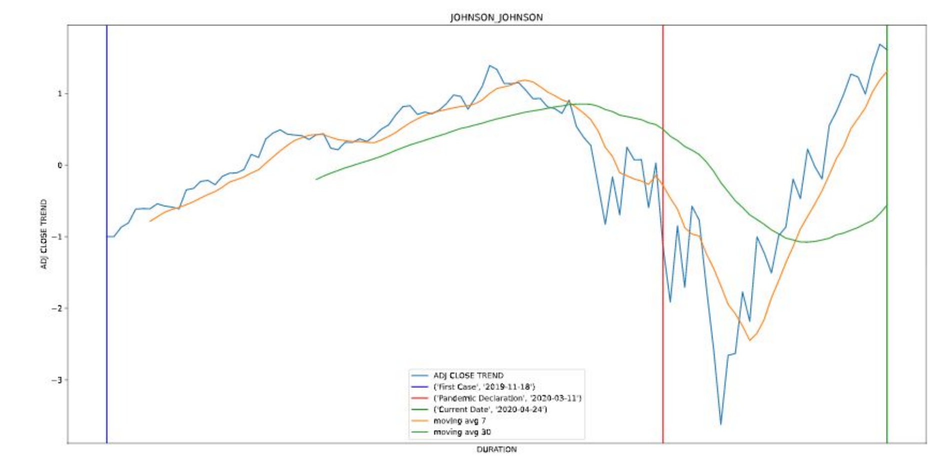
- K nearest neighbors
- Linear Support Vector Classifier
- Random forest classifier
- Ensemble classifier with all three (majority vote)

```
########### actual ml
def do_ml(pandemic,ticker):
    X,y=extract_feature_sets(pandemic,ticker)

    X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=.25,random_state=42)


    # K nearest neighbors
    clf = neighbors.KNeighborsClassifier()
    # voting classifies
    clf = VotingClassifier([
        ('lsvc', svm.LinearSVC()),
        ('knn', neighbors.KNeighborsClassifier()),
        ('rfor', RandomForestClassifier())
    ])



    clf.fit(X_train,y_train)
    confidence = clf.score(X_test,y_test)
    print('Accuracy:',confidence)
    predictions = clf.predict(X_test)
    print('Predicted spread:',Counter(predictions))

    return confidence
```

Data Visualization

- Blue Indicates the first reported case, Red indicates when the pandemic was declared, and green is the current date.
- Used moving averages: Yellow is short term moving average, Green is long term moving average.
- Used the subsetted data for each pandemic, plots created by matplotlib





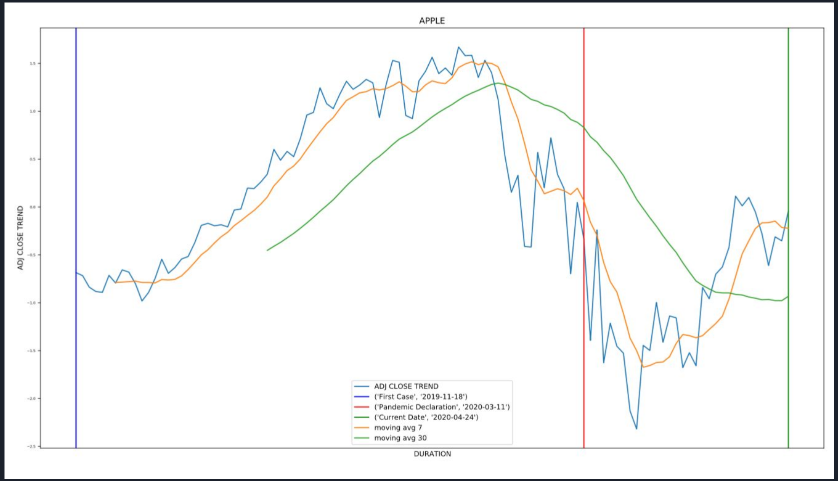
Results - Apple = BUY

	buy-train	sell-train	hold-train	accuracy	buy-test	sell-test	hold-test
SARS	160	146	10	0.53	52	19	0
H1N1	183	122	30	0.45	65	19	0
Ebola	122	211	102	0.41	64	69	1
COVID-19	62	39	8	0.4642	25	2	1

```
AAPL
sars
Data spread:  Counter({'1': 118, '-1': 105, '0': 8})
Accuracy: 0.5
Predicted spread: Counter({1: 41, -1: 17})


swine
Data spread:  Counter({'1': 183, '-1': 122, '0': 30})
Accuracy: 0.42857142857142855
Predicted spread: Counter({1: 70, -1: 14})


ebola
Data spread:  Counter({'1': 221, '-1': 211, '0': 102})
Accuracy: 0.3805970149253731
Predicted spread: Counter({-1: 70, 1: 64})


corona
Data spread:  Counter({'1': 62, '-1': 39, '0': 8})
Accuracy: 0.5357142857142857
Predicted spread: Counter({1: 25, -1: 2, 0: 1})
```

Apple moving average and ARIMA predictions

- Coefficient = .0009
- 4-27 prediction is $283.21 actual value is $283.17
- 4-28 prediction is $283.45 actual value is $278.58
- 4-29 prediction is $283.70 actual value is $287.73
- RMSE = 3.65



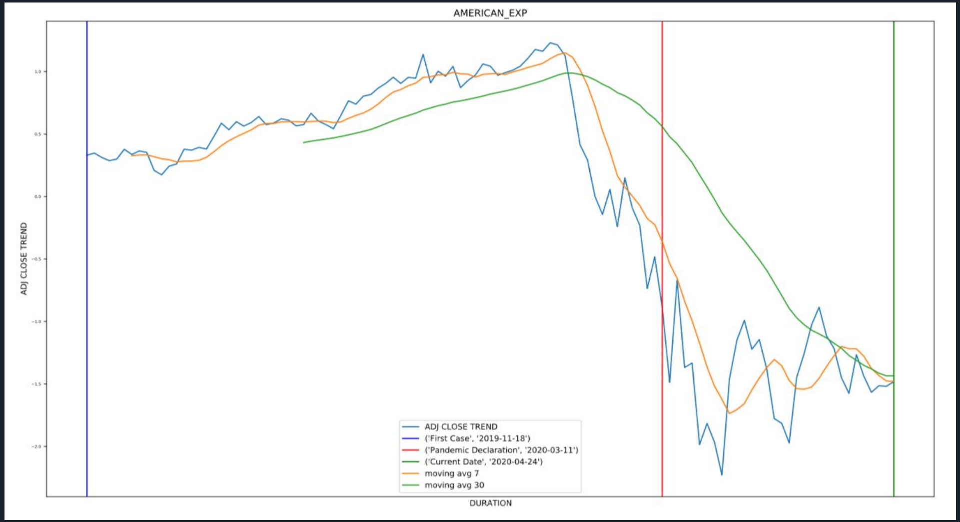
American Express = BUY

AXP							
	buy-train	sell-train	hold-train	accuracy	buy-test	sell-test	hold-test
SARS	147	131	38	0.4684	45	33	1
H1N1	191	122	22	0.5952	71	13	0
Ebola	156	205	173	0.3731	7	111	16
COVID-19	46	40	23	0.4643	19	7	2

```
AXP
sars
Data spread:  Counter({'1': 111, '-1': 102, '0': 18})
Accuracy: 0.4827586206896552
Predicted spread: Counter({-1: 38, 1: 20})


swine
Data spread:  Counter({'1': 191, '-1': 122, '0': 22})
Accuracy: 0.5833333333333334
Predicted spread: Counter({1: 72, -1: 12})


ebola
Data spread:  Counter({'-1': 205, '0': 173, '1': 156})
Accuracy: 0.35074626865671643
Predicted spread: Counter({-1: 117, 0: 11, 1: 6})


corona
Data spread:  Counter({'1': 46, '-1': 40, '0': 23})
Accuracy: 0.4642857142857143
Predicted spread: Counter({1: 21, -1: 6, 0: 1})
```

American Express moving average and ARIMA

- Coefficient = .0002
- 4-27 prediction is $83.18 actual value is $85.06
- 4-28 prediction is $83.19 actual value is $88.19
- 4-29 prediction is $83.21 actual value is $96.12
- RMSE = 8.06



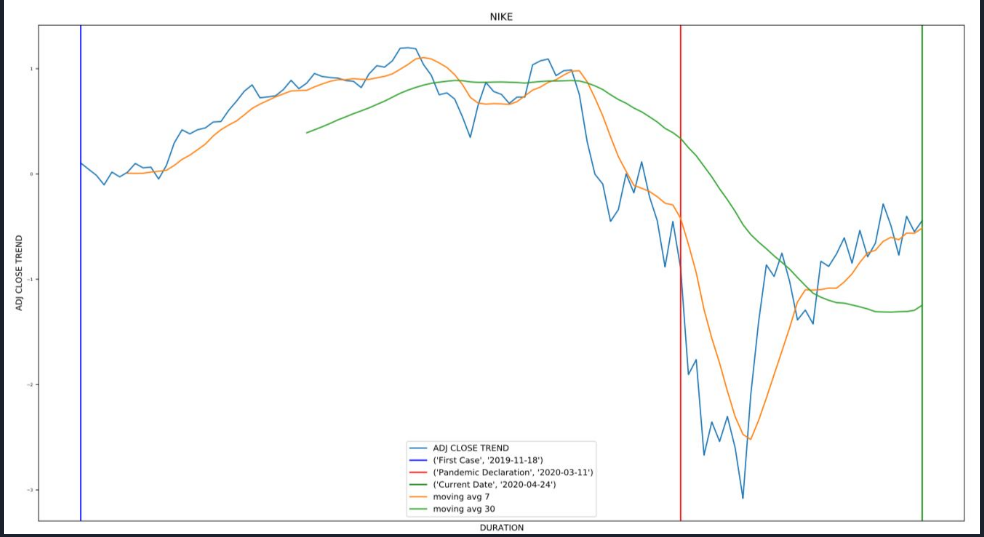
Nike = BUY

NIKE							
	buy-train	sell-train	hold-train	accuracy	buy-test	sell-test	hold-test
SARS	160	121	53	0.5063	52	26	1
H1N1	161	121	53	0.4167	68	14	2
Ebola	208	172	53	0.3657	74	52	8
COVID-19	51	39	19	0.3214	21	6	0

```
NKE
sars
Data spread:  Counter({'1': 119, '-1': 88, '0': 24})
Accuracy: 0.3793103448275862
Predicted spread: Counter({1: 46, -1: 12})


swine
Data spread:  Counter({'1': 161, '-1': 121, '0': 53})
Accuracy: 0.4880952380952381
Predicted spread: Counter({1: 66, -1: 17, 0: 1})


ebola
Data spread:  Counter({'1': 208, '-1': 172, '0': 154})
Accuracy: 0.3656716417910448
Predicted spread: Counter({1: 76, -1: 51, 0: 7})


corona
Data spread:  Counter({'1': 51, '-1': 39, '0': 19})
Accuracy: 0.35714285714285715
Predicted spread: Counter({1: 19, -1: 8, 0: 1})
```

Nike moving average and ARIMA

- Coefficient = .0008
- 4-27 prediction is $88.44 actual value is $89.37
- 4-28 prediction is $88.50 actual value is $88.8
- 4-29 prediction is $88.57 actual value is $88.07
- RMSE = 8.06



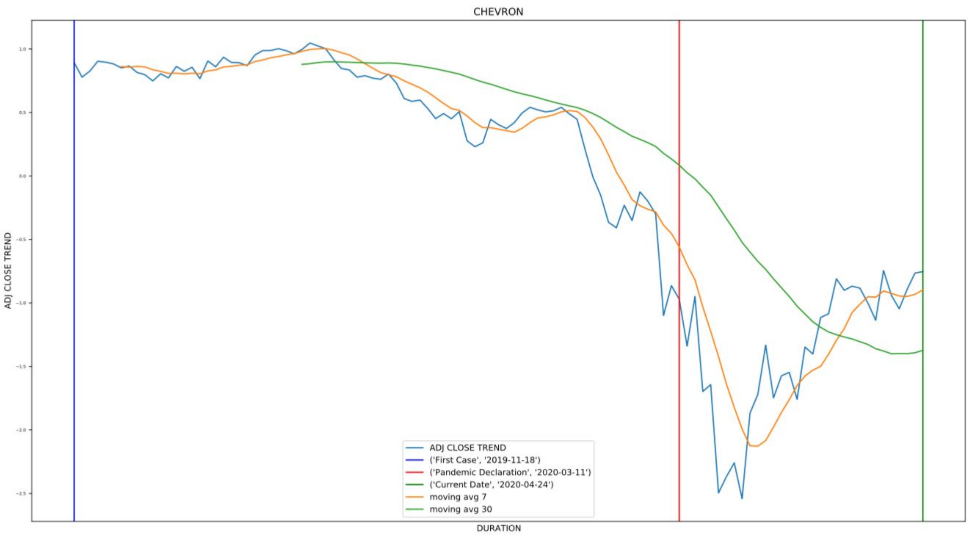
Chevron

CVX							
	buy-train	sell-train	hold-train	accuracy	buy-test	sell-test	hold-test
SARS	128	76	112	0.4431	32	16	31
H1N1	149	127	59	0.3571	56	25	3
Ebola	163	217	154	0.3806	5	114	15
COVID-19	30	54	25	0.5714	4	22	2

```
CVX
sars
Data spread:  Counter({'1': 82, '0': 79, '-1': 70})
Accuracy: 0.46551724137931033
Predicted spread: Counter({-1: 24, 1: 20, 0: 14})


swine
Data spread:  Counter({'1': 149, '-1': 127, '0': 59})
Accuracy: 0.34523809523809523
Predicted spread: Counter({1: 60, -1: 24})


ebola
Data spread:  Counter({'-1': 217, '1': 163, '0': 154})
Accuracy: 0.3656716417910448
Predicted spread: Counter({-1: 113, 0: 18, 1: 3})


corona
Data spread:  Counter({'-1': 54, '1': 30, '0': 25})
Accuracy: 0.4642857142857143
Predicted spread: Counter({-1: 21, 1: 4, 0: 3})
```

Chevron moving average and ARIMA

- Coefficient = .0003
- 4-27 prediction is $87.04 actual value is $89.37
- 4-28 prediction is $87.06 actual value is $88.8
- 4-29 prediction is $87.09 actual value is $88.07
- RMSE = 4.89



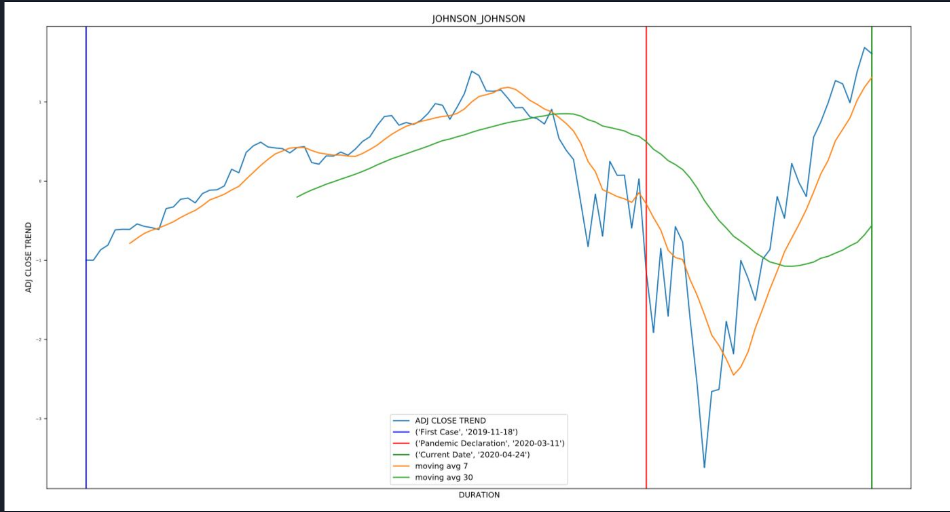
Johnson & Johnson = BUY

JNJ							
	buy-train	sell-train	hold-train	accuracy	buy-test	sell-test	hold-test
SARS	110	143	63	0.4404	11	64	4
H1N1	97	64	174	0.4881	4	6	74
Ebola	153	144	237	0.4254	4	31	99
COVID-19	52	29	28	0.3571	23	0	5

```
JNJ
sars
Data spread:  Counter({'-1': 116, '1': 76, '0': 39})
Accuracy: 0.46551724137931033
Predicted spread: Counter({-1: 53, 1: 5})


swine
Data spread:  Counter({'0': 174, '1': 97, '-1': 64})
Accuracy: 0.5
Predicted spread: Counter({0: 72, -1: 8, 1: 4})


ebola
Data spread:  Counter({'0': 237, '1': 153, '-1': 144})
Accuracy: 0.39552238805970147
Predicted spread: Counter({0: 97, -1: 30, 1: 7})


corona
Data spread:  Counter({'1': 52, '-1': 29, '0': 28})
Accuracy: 0.4642857142857143
Predicted spread: Counter({1: 21, 0: 6, -1: 1})
```

JNJ moving average and ARIMA

- Coefficient = .0003
- 4-27 prediction is $154.91 actual value is $154.29
- 4-28 prediction is $154.96 actual value is $151.39
- 4-29 prediction is $155.02 actual value is $150.24
- RMSE = 3.46



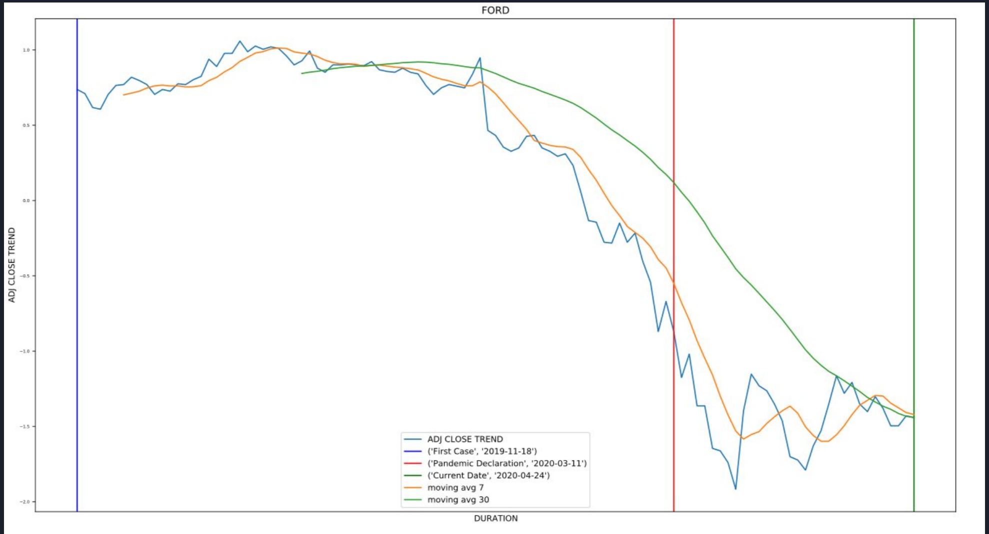
Ford = SELL

F							
	buy-train	**sell-train**	**hold-train**	**accuracy**	**buy-test**	**sell-test**	**hold-test**
SARS	164	131	21	0.481	57	22	0
H1N1	185	139	11	0.5119	61	23	0
Ebola	191	212	131	0.3582	10	110	14
COVID-19	34	56	19	0.3929	2	3	23

```
F
sars
Data spread:  Counter({'1': 112, '-1': 106, '0': 13})
Accuracy: 0.5172413793103449
Predicted spread: Counter({1: 41, -1: 17})


swine
Data spread:  Counter({'1': 185, '-1': 139, '0': 11})
Accuracy: 0.5238095238095238
Predicted spread: Counter({1: 64, -1: 20})


ebola
Data spread:  Counter({'-1': 212, '1': 191, '0': 131})
Accuracy: 0.40298507462686567
Predicted spread: Counter({-1: 106, 0: 16, 1: 12})


corona
Data spread:  Counter({'-1': 56, '1': 34, '0': 19})
Accuracy: 0.4642857142857143
Predicted spread: Counter({-1: 23, 0: 4, 1: 1})
```

Ford moving average and ARIMA

- Coefficient = -.0002
- 4-27 prediction is $4.868 actual value is $5.17
- 4-28 prediction is $4.867 actual value is $5.38
- 4-29 prediction is $4.866 actual value is $5.26
- RMSE = .41



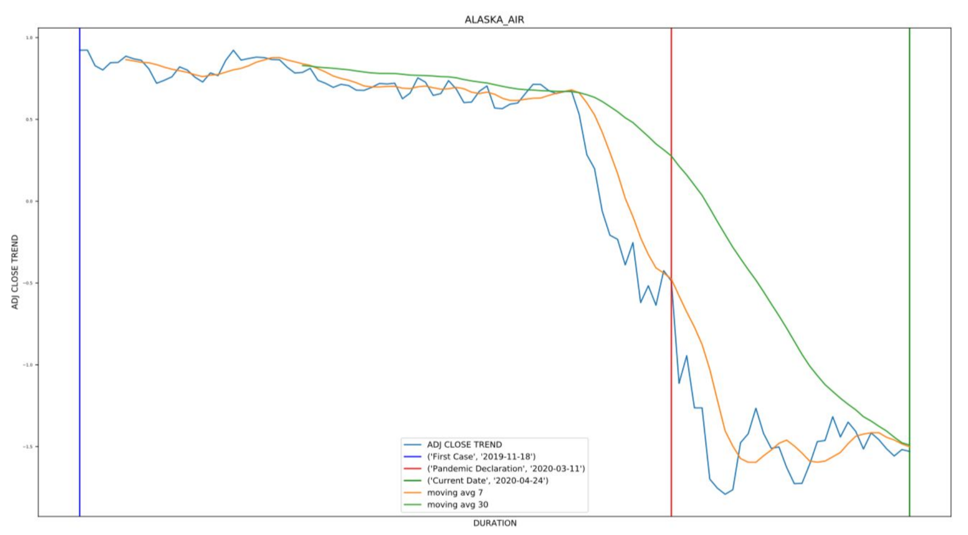
Alaska Airlines

ALK							
	buy-train	sell-train	hold-train	accuracy	buy-test	sell-test	hold-test
SARS	153	151	12	0.4684	43	36	0
H1N1	211	116	8	0.5238	65	19	0
Ebola	294	204	36	0.5298	104	30	0
COVID-19	35	62	12	0.4643	2	25	1

```
ALK
sars
Data spread:  Counter({'1': 122, '-1': 106, '0': 3})
Accuracy: 0.41379310344827586
Predicted spread: Counter({-1: 33, 1: 25})


swine
Data spread:  Counter({'1': 211, '-1': 116, '0': 8})
Accuracy: 0.5357142857142857
Predicted spread: Counter({1: 68, -1: 16})


ebola
Data spread:  Counter({'1': 294, '-1': 204, '0': 36})
Accuracy: 0.5522388059701493
Predicted spread: Counter({1: 97, -1: 37})


corona
Data spread:  Counter({'-1': 62, '1': 35, '0': 12})
Accuracy: 0.4642857142857143
Predicted spread: Counter({-1: 25, 1: 2, 0: 1})
```

Alaska Airlines moving average and ARIMA

- Coefficient = .0003
- 4-27 prediction is $4.868 actual value is $27.85
- 4-28 prediction is $4.867 actual value is $31.58
- 4-29 prediction is $4.866 actual value is $34.00
- RMSE = 4.01



Conclusion

- We are not financial advisors!
- We trained the classifier on all the data from January 2000 Until April 20
- Predictions we’re to buy all except Johnson and Johnson

REFERENCES

[1] Angelos Delivorias. Nicole Scholz. 2020. Economic impact of epidemics and pandemics. (2020). Retrieved April 30, 2020

[2] Michael Grogan ARIMA and Python: Stock Price Forecasting using statsmodels. Retrieved April 30, 2020 from https://youtu.be/o7Ux5jKEbcw

[3] Harrison Kinsley. ARIMA and Python: Stock Price Forecasting using statsmodels. Retrieved April 30, 2020 from https://www.youtube.com/watch?v=2BrpKpWwT2A&list=PLQVvvaa0QuDcOdF96TBtRtuQksErCEBYZ

[4] Ishan Shah. How Pandemics Impact Financial Markets. (April 2020). Retrieved https://towardsdatascience.com/pandemics-impact-financial-markets-9a4feb6951f5

[5] Carson Kai-Sang Leung, Richard Kyle MacKinnon, and Yang Wang. 2014. A machine learning approach for stock price prediction. In Proceedings of the 18th International Database Engineering & Applications Symposium(IDEAS ’14). Association for Computing Machinery, New York, NY, USA, 274–277.

[6] Pratik Patil, Ching-Seh Mike Wu, Katerina Potika, and Marjan Orang. 2020. Stock Market Prediction Using Ensemble of Graph Theory, Machine Learning and Deep Learning Models. In Proceedings of the 3rd International Conference on Software Engineering and Information Management (ICSIM ’20). Association for Computing Machinery, New York, NY, USA, 85–92. DOI:https://doi.org/10.1145/3378936.3378972

[7] Biao Huang, Qiao Ding, Guozi Sun, and Huakang Li. 2018. Stock Prediction based on Bayesian-LSTM. In Proceedings of the 2018 10th International Conference on Machine Learning and Computing (ICMLC 2018). Association for Computing Machinery, New York, NY, USA, 128–133.