DS 5220 Project Presentation

John Drohan,
Nahush Bhat,
Harshkumar Modi

Financial Time-Series Analysis

Motivation

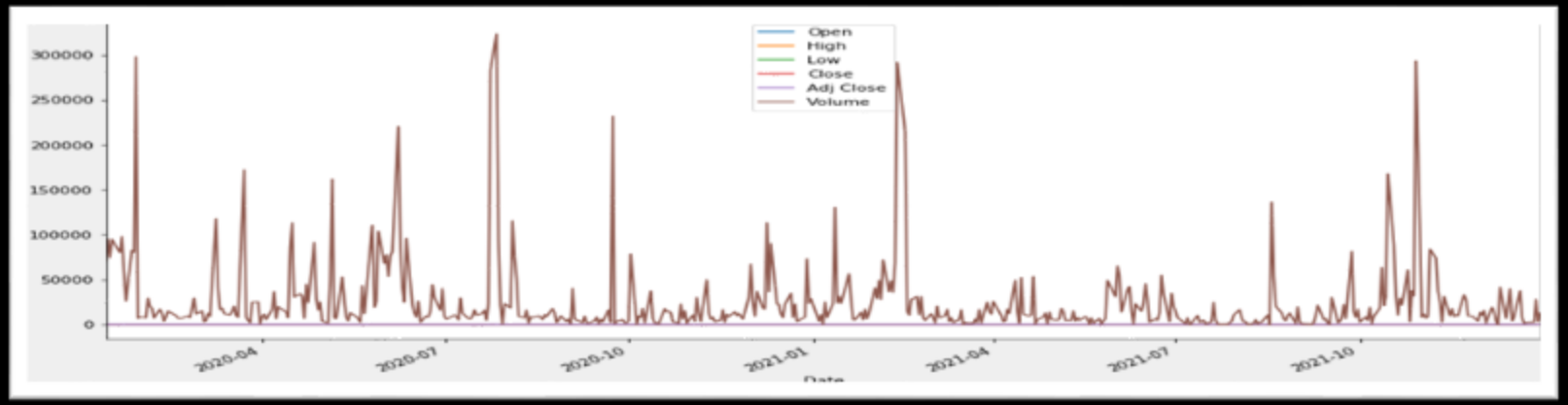
We want to investigate the efficacy of Supervised learning methods for Feature Selection and Feature Engineering. The evaluation will focus on financial time-series data.

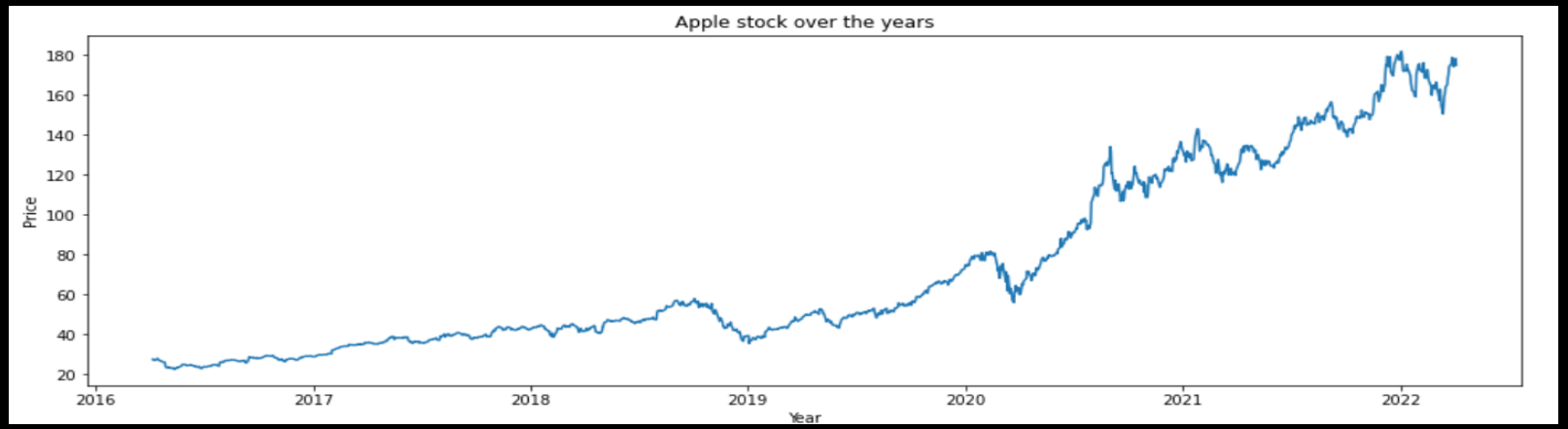
Dataset

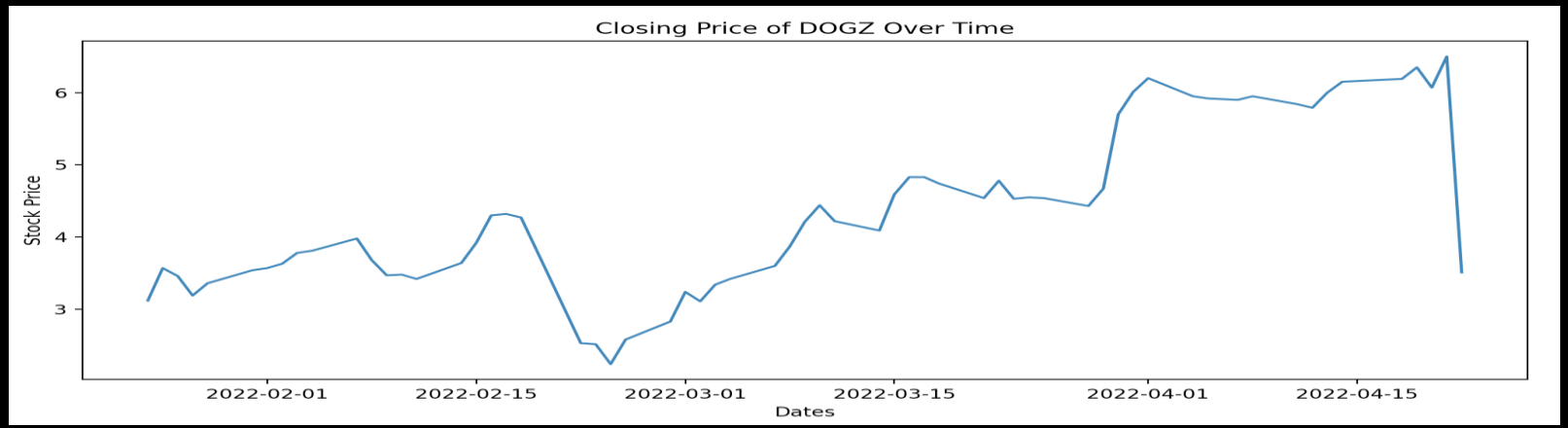
- Data is scraped from Yahoo! Finance using the *yfinance* library
- Dataset Period of 1 year prior to today's date.
- As part of EDA, each member of the team identified 10 stocks from NASDAQ, based on the following criteria:
    1. Seasonality (Nahush)
    2. Strong Trend (Harshkumar)
    3. Volatility (John)

Dataset

1. Seasonality
2. Strong Trend
3. Volatility







Dataset

- Stock market time-series data comes with standard set of features:

Date	High	Low	Open	Close	Volume	Adj Close

- We leveraged the *Technical Analysis* library to add an additional 94 features to each stock. This library has features that market experts frequently use. These are generated using the above existing features.

Project Scope

Study

Study the effectiveness of LASSO Regularization for feature selection

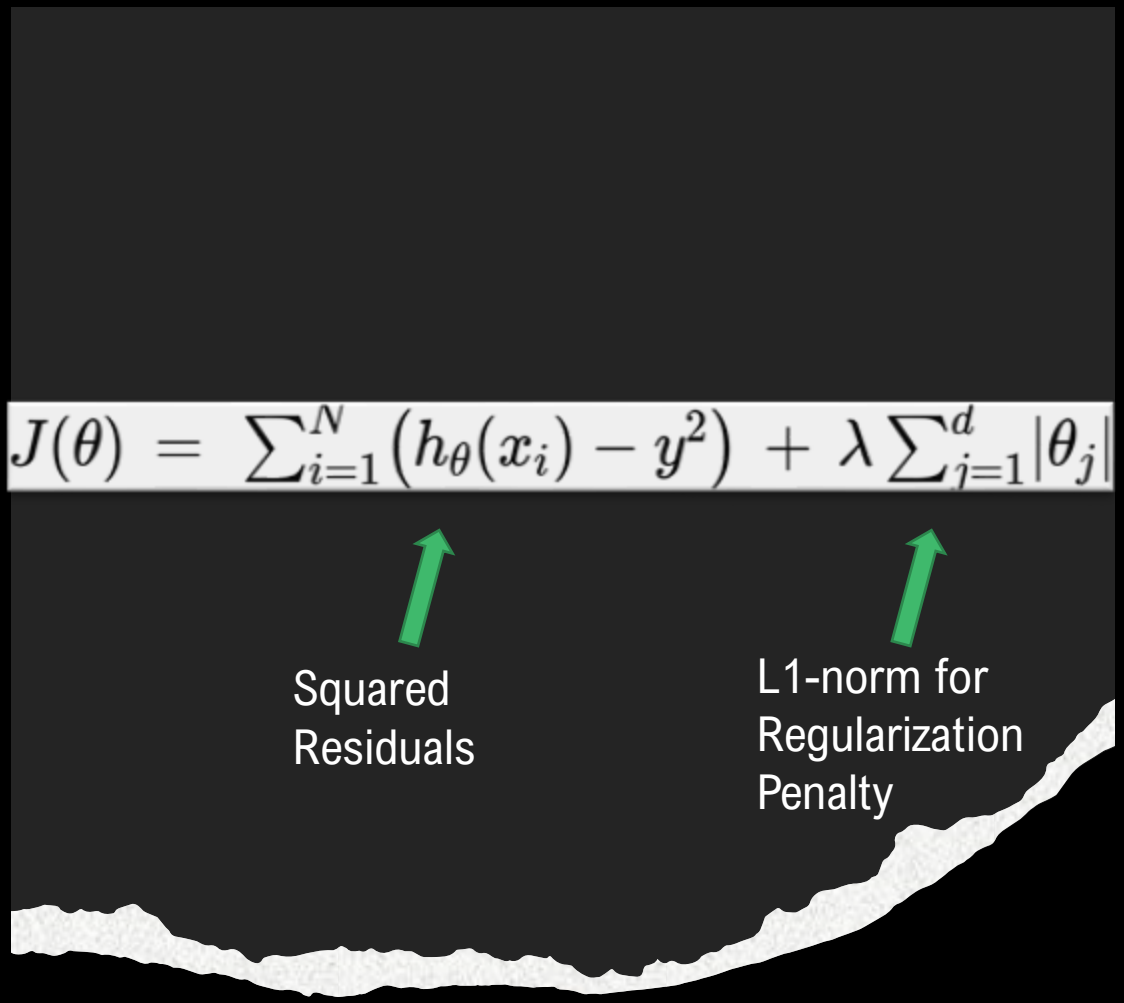
Compare

Compare its results against

- KBest – Wrapper (Supervised)
- PCA – Unsupervised

Use

Use LSTM (Long Short-Term Memory) neural network to compare the prediction performance of the features generated by the 3 methods



Methodology

LASSO Regularization

- lasso (least absolute shrinkage and selection operator) is a regression analysis method.
- It improves prediction error by shrinking the sum of the squares of the regression coefficients to be less than a fixed value, thus reduce overfitting.
- It also performs covariate selection which forces certain coefficients to zero, excluding them from impacting prediction.

Methodology

LASSO Regularization

Metric	Performance
Mean Squared Error	0.026808555
R-MSE	0.163733183
R2 Score	0.155015702
Mean Absolute % Error	0.611694930

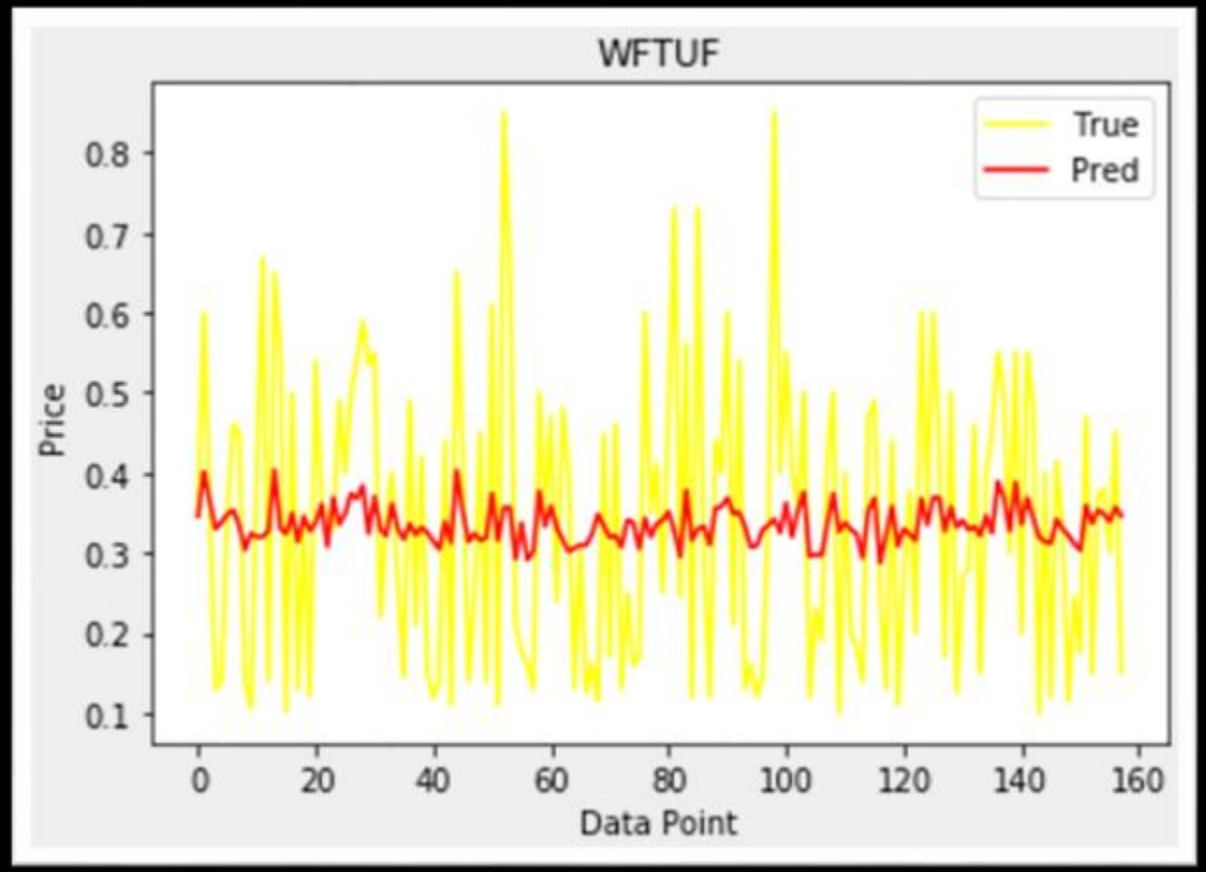


Fig: Prediction Model using LASSO on seasonal stock

Methodology

LASSO Regularization

Feature	Assigned Weights
macd	0.01255006329
bb_bbh	0.01242458302
SMA20	0.01023300562
ROC10	0.00863088891
bb_bbm	0.0055034444
bb_bbl	0.00396051131
Dlog10	0.00168713342
macd_diff	0.00112172765
EMA10	0.0006648942
RSI10	0.00046194829

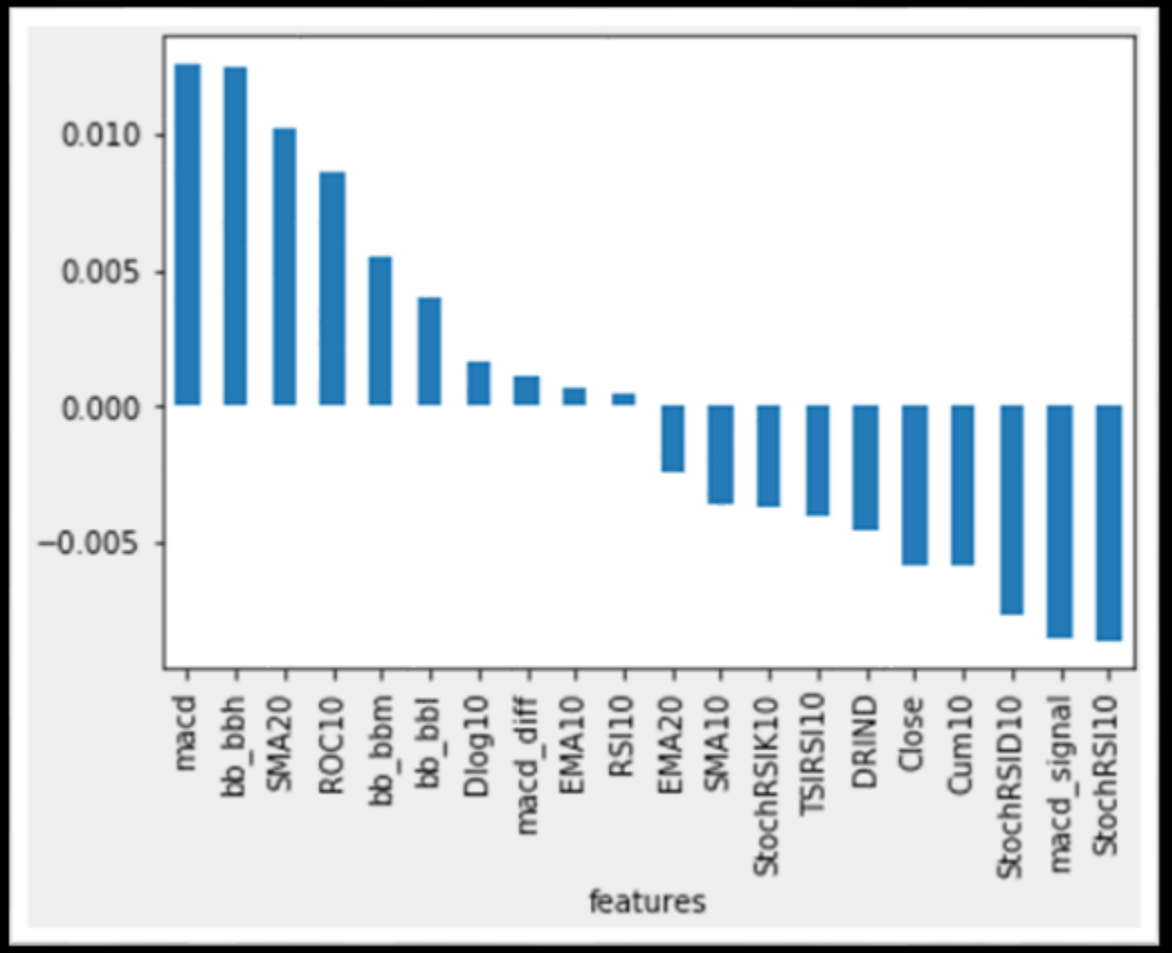
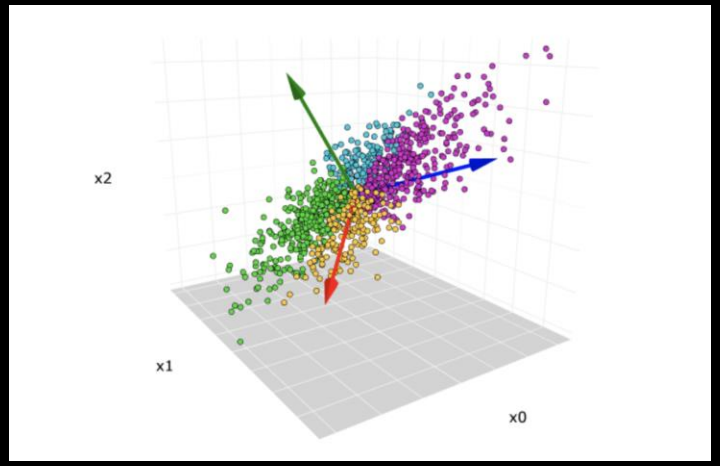


Fig: Co-efficient feature Weights selected by LASSO

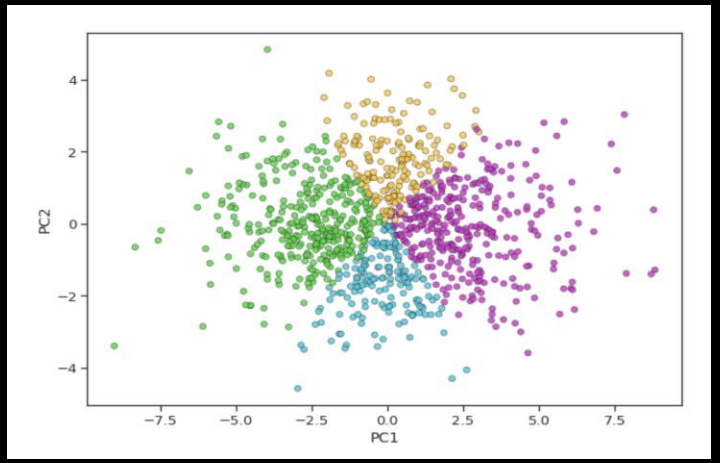
Methodology

- Principle Component Analysis
  - Dimensionality Reduction that maximizes information retention
  - Information explained through variance

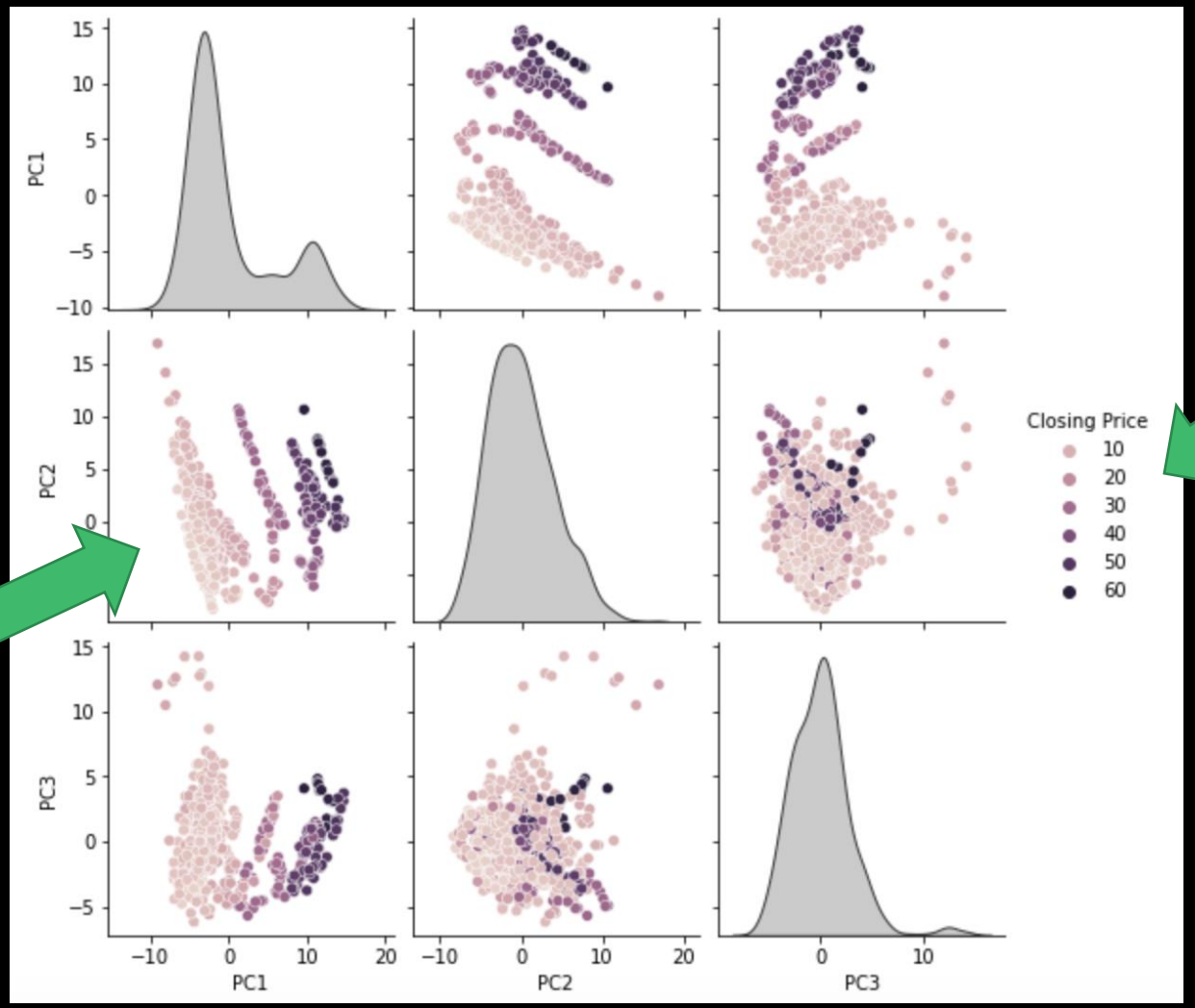
- Steps in PCA Reduction
  - Standardize Data
  - Create Covariance Matrices
  - Use Eigenvectors and values to find components
  - Select Optimal Component Value



Application of PCA
3D to 2D



Seaborn Pair-Plot to Compare top 3 Principal Components

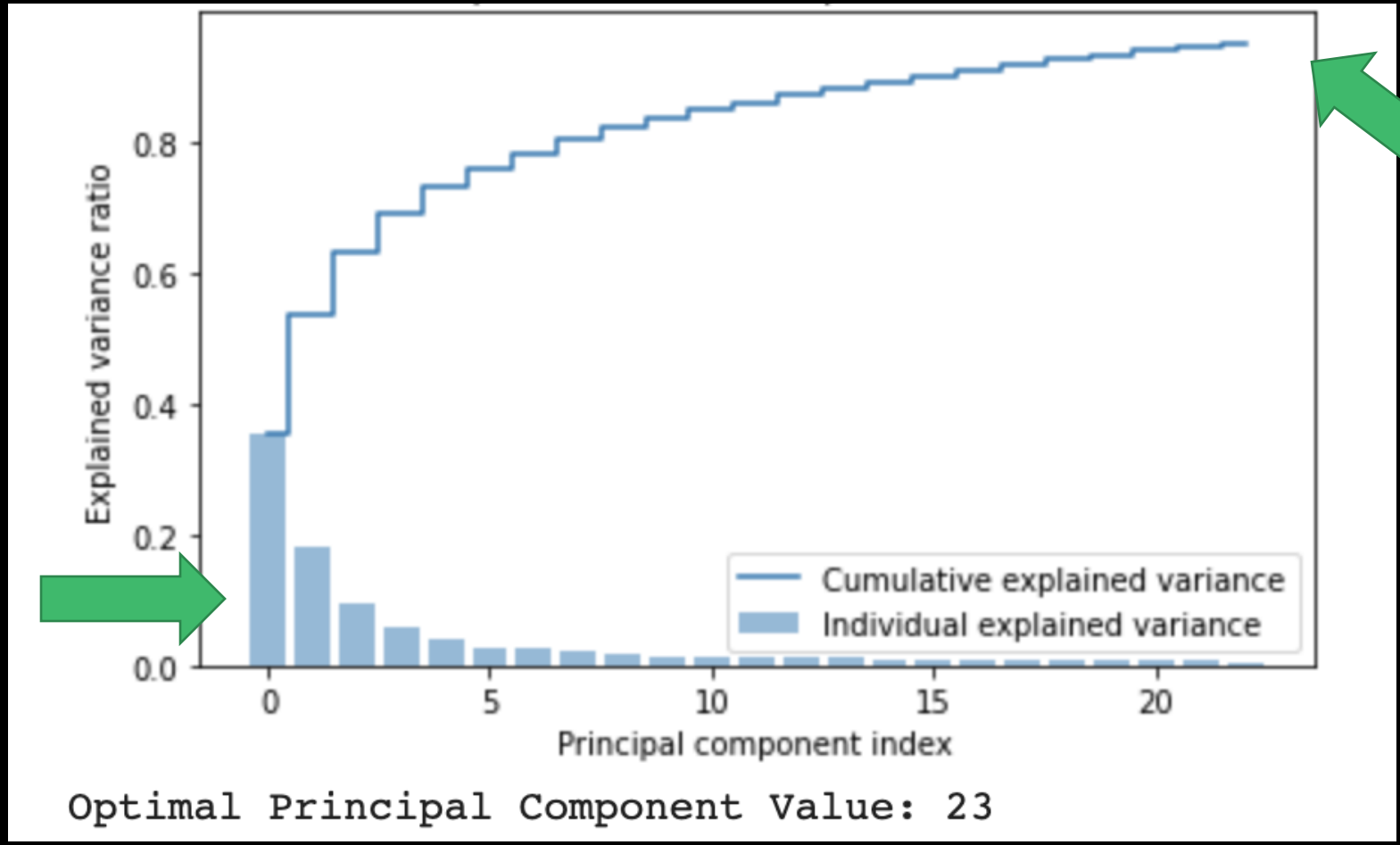


Closing Price of Stock shaded by Value in USD

Even just the first two PC can meaningfully separate Data

Top 3 Components explained over 60% of the Variance

Cumulative Summation of Explained Variance

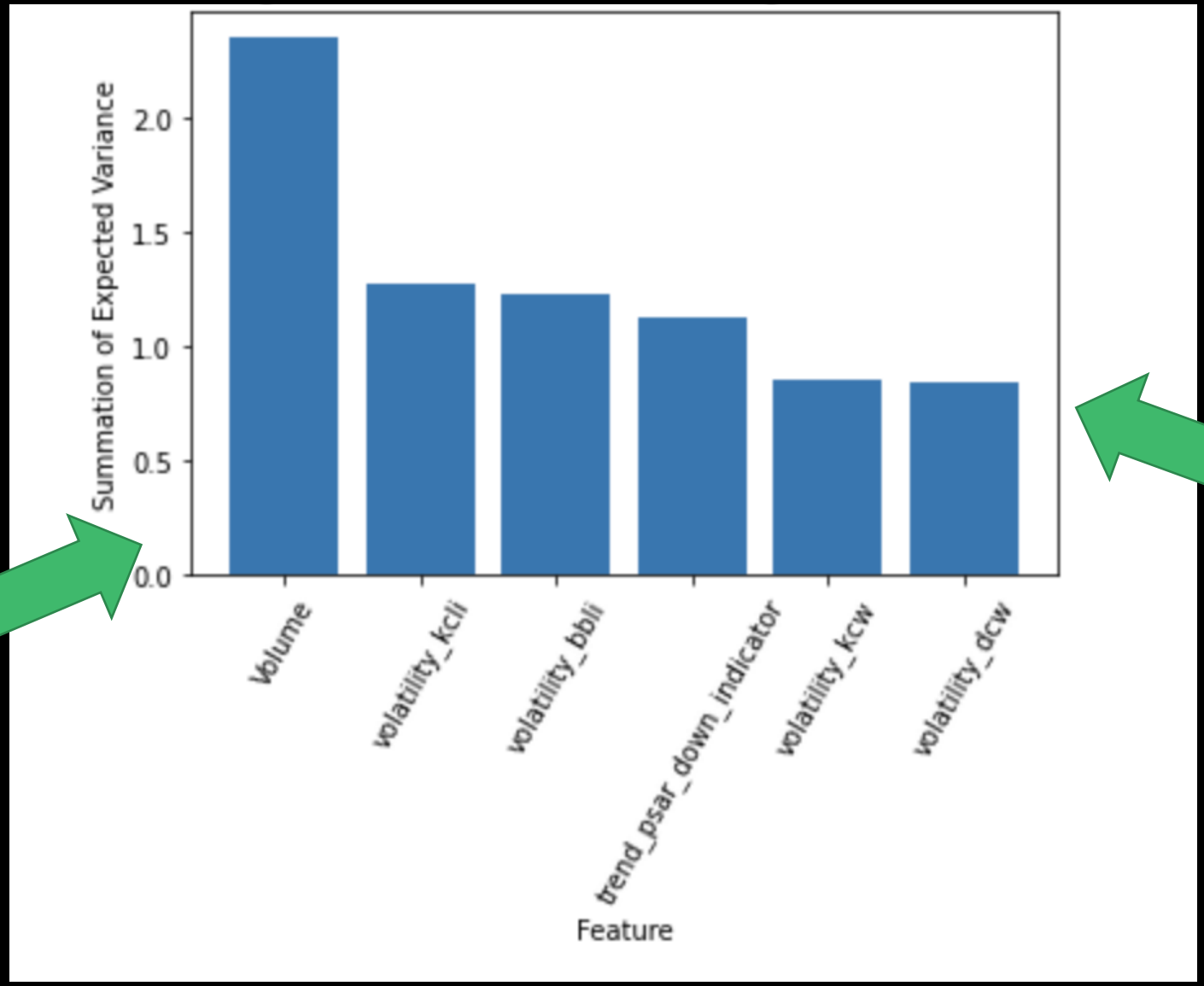


Initial Components are most significant

Final Variance explained summation is about 0.95

Minimum number of Components required for exp_var >= 0.95

Summation of Expected Variance for Top Features



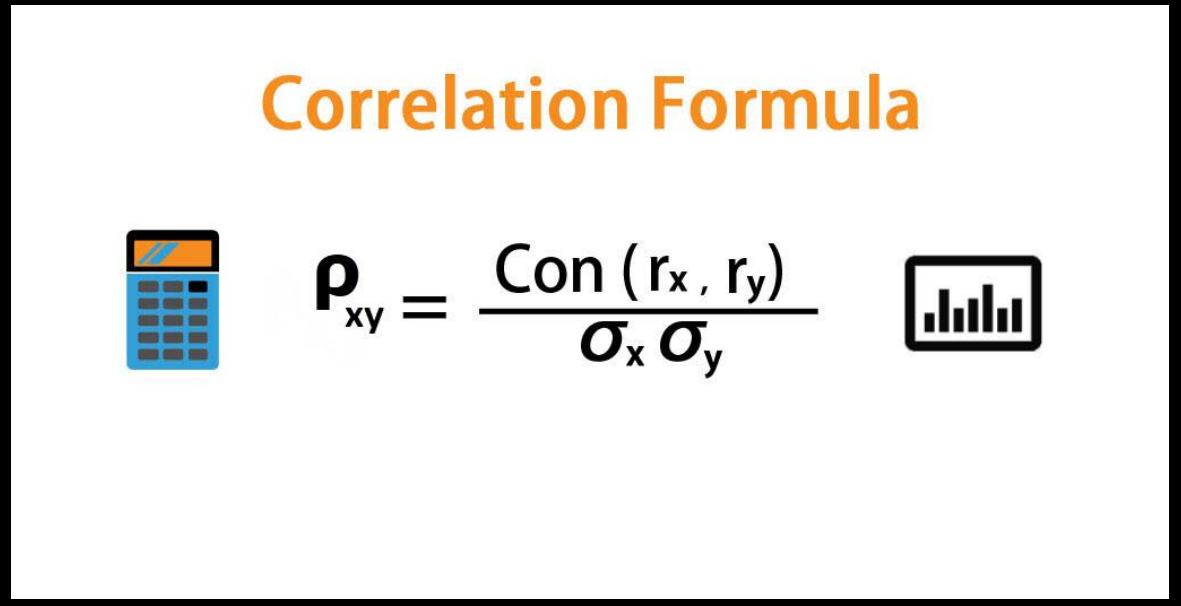
Sharp decrease in Variance after just the first feature

Total variance contributed by 6 most important features

Methodology

K-Best

- KBest is a feature selection method that filters out the best contributing features of the dataset to the target variable



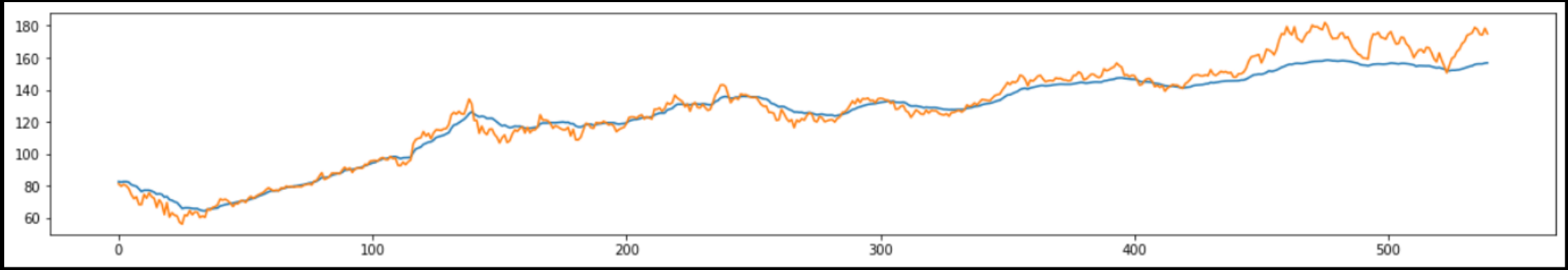
K-Best

- KBest uses correlation method to test every feature with the target variable.
- It then stores the correlation values and sorts the data having best values as features
- The p-value here helps to determine if the null hypothesis is followed or rejected. If the p-value is greater than 0.05, then the feature is not contributing to the target variable.

	Features	Correlation	P-Value
88	others_cr	1.000000	0.0
3	Adj Close	0.999954	0.0
1	High	0.999861	0.0
2	Low	0.999841	0.0
0	Open	0.999709	0.0
41	trend_ema_fast	0.998988	0.0
52	trend_ichimoku_conv	0.998931	0.0
24	volatility_kcl	0.998790	0.0
22	volatility_kcc	0.998753	0.0
85	momentum_kama	0.998669	0.0
39	trend_sma_fast	0.998574	0.0
23	volatility_kch	0.998562	0.0
54	trend_ichimoku_a	0.998385	0.0
12	volume_vwap	0.998383	0.0
42	trend_ema_slow	0.997807	0.0
31	volatility_dcm	0.997685	0.0
15	volatility_bbm	0.997564	0.0
53	trend_ichimoku_base	0.997108	0.0
16	volatility_bbh	0.997028	0.0
30	volatility_dch	0.996947	0.0
40	trend_sma_slow	0.996844	0.0
29	volatility_dcl	0.996662	0.0
17	volatility_bbl	0.996091	0.0
55	trend_ichimoku_b	0.995287	0.0
61	trend_visual_ichimoku_a	0.989416	0.0

Steps in KBest

- We find the pearson correlation value of each feature.
- Next we sort it in descending order.
- From different K values, we chose the best K value that fits and gives accurate model.




```
In [80]:  1  r2_score(y_actual_inv_transformed, y_pred_inv_transformed)
Out[80]: 0.9436808227269993
```

Next Steps

- Implement data balancing methods to control for strong statistical characteristics found in financial time-series data
- Perform feature selection for all types of stocks (seasonal, trending and volatile) and measure the differences in model performance
- Comparing results of all 30 stocks on model performance of LSTM
- Implementing k-fold cross validation to improve regularization results

Team Member Contributions

- Nahush Bhat – Proposal, PPT, Report, Code (Pre-Process, EDA, Lasso)
- John Drohan – Proposal, PPT, Report, Code (Pre-Process, EDA, PCA)
- Harshkumar – Proposal, PPT, Report, Code (Pre-Process, EDA, Kbest)