Stock Market Trend Analysis

Final Year Project End-term presentation

Supervised by:

Dr. Vivek Chaturvedi

Sudhakar Mishra|1SG17CS091

Objective

To develop a method for prediction of stock price direction, that

- Creates and selects the best features
- Creates different models
- Produces a descriptive result
- Shall be robust
- Could be used for Commercial purpose

Motivation

- Lots of literature
- Not very much commercialization of research
- Mathematical confidence is better than random guess
- Risks are high
- Challenging data
- New things to learn about stock market
- Understanding statistical approach
- Use of coding expertise, data structure, machine learning

Tools used

- Python
- Numpy
- Pandas
- Matplotlib
- Ta-lib
- Sk-learn
- Keras
- StatsModel
- Tkinter

Key-steps

- Analyzing different stock market data
- Reviewing current literature
- Finding flaws with current literature
- Providing Solution to those
- Creating features
- Feature selection
- Defining new better models
- Understanding the time-series characteristic of data
- Developing model solely based on that
- Comparison of models
- Result analysis

Background

- Raw data contains Open, Low, High, Close, Volume for each working day.
- Two types of prediction : Open/Close, Close/Close(more preferred?).
- Single day prediction/Multiple day prediction(more easy?)

How do most of the traders currently work?

- Use their business understanding.
- Try to analyze impact of international market to local market.
- Closely watch the govt. Decisions, upcoming festivals, occasional events, companies policy etc.
- Often indulge in improper means to gain information like insider trading.
- They also use some statistical tool(??) to analyze the trend

Statistical tool

- Technical indicators used by traders
- Mathematical formulas in terms of OHLCV.
- Indicate different states of stocks
- Few of them are:-Relative Strength Indicator, Stochastic %K, STochastic %D, Slow %D, Momentum, Rate of Change, Williams %R, A/D oscillator, Disparity, Price Oscillator(OSCP), Commodity Channel Index, Triple Exponential Moving Average etc.
- A lot more of them, each holding some particular significance.
- Can be calculated over variable number of days to fit our prediction need(short-term/long-term).

Relative Strength Indicator(RSI)

- RSI = 100 – [100 / (1 + (Average of Upward Price Change / Average of Downward Price Change))] where the Upward and Downward price change may be calculated over variable number of days(preferrably in multiple of week days).
- Lies between 0-100, >70 indicating overbought(?) and <30 indicating oversold(?)
- Traders use this directly to buy and sell stocks during extreme condition.
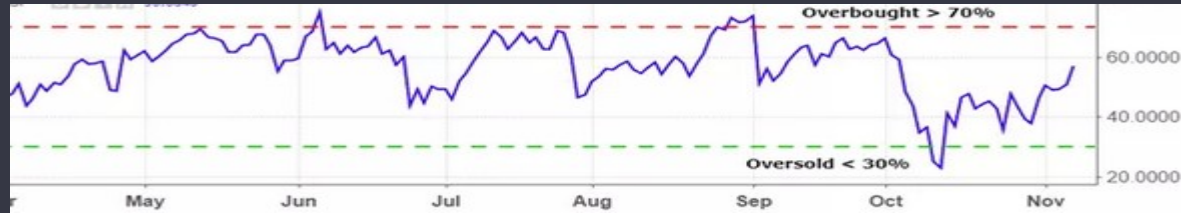
Moving Average Convergence Divergence(MACD)

- MACD is calculated by subtracting the 26-period EMA from the 12-period EMA.
- 9-period EMA is considered as signal line
- Above the signal line represents the buying condition while below it represents selling condition..
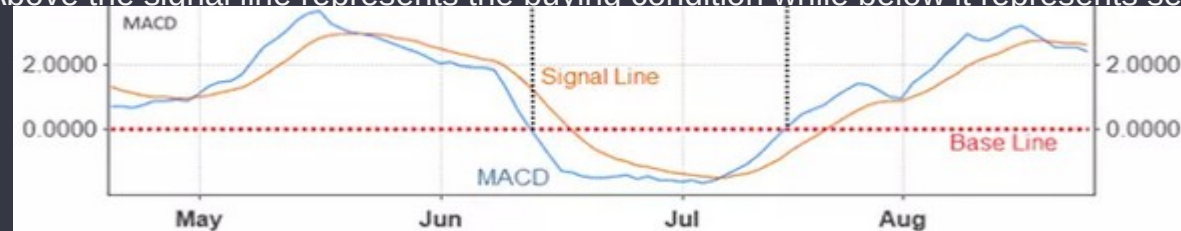
Why should we use these indicators?

- Almost all researchers suggest this
- Bad performance over OHLCV
- Improved performance over these indicators
- Each indicators have their own importance and in models like random forest, it is like taking a vote from different features.
- Most important, it cancels the short term noise(daily noise)
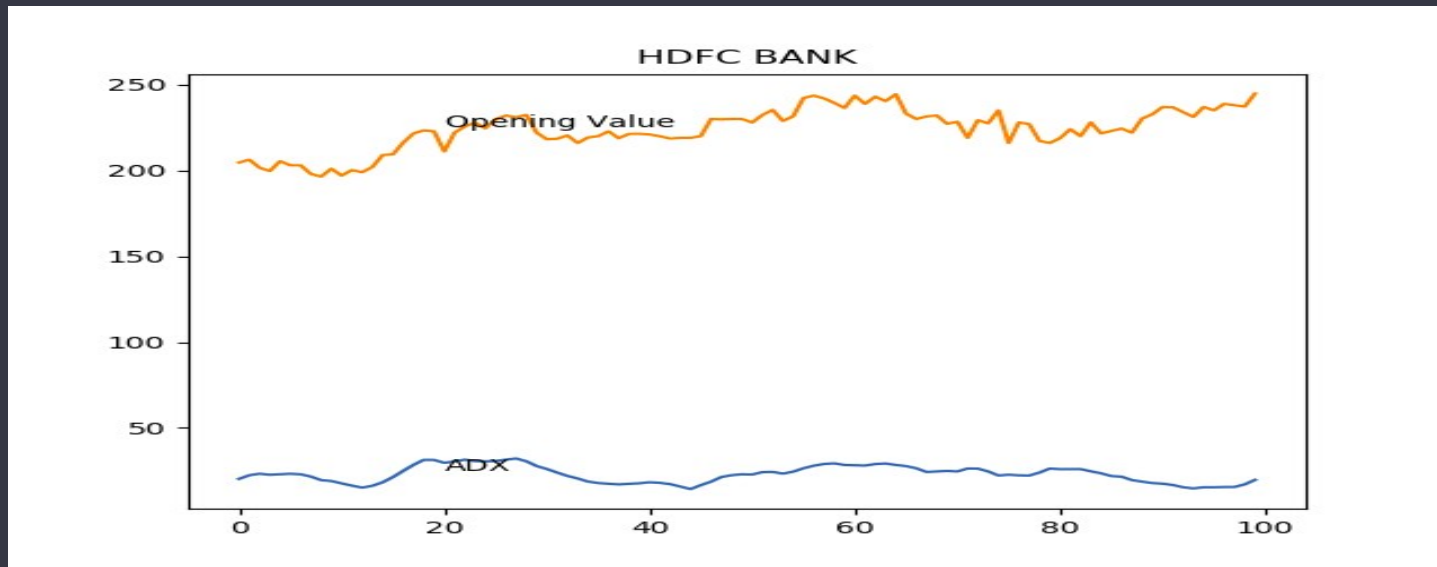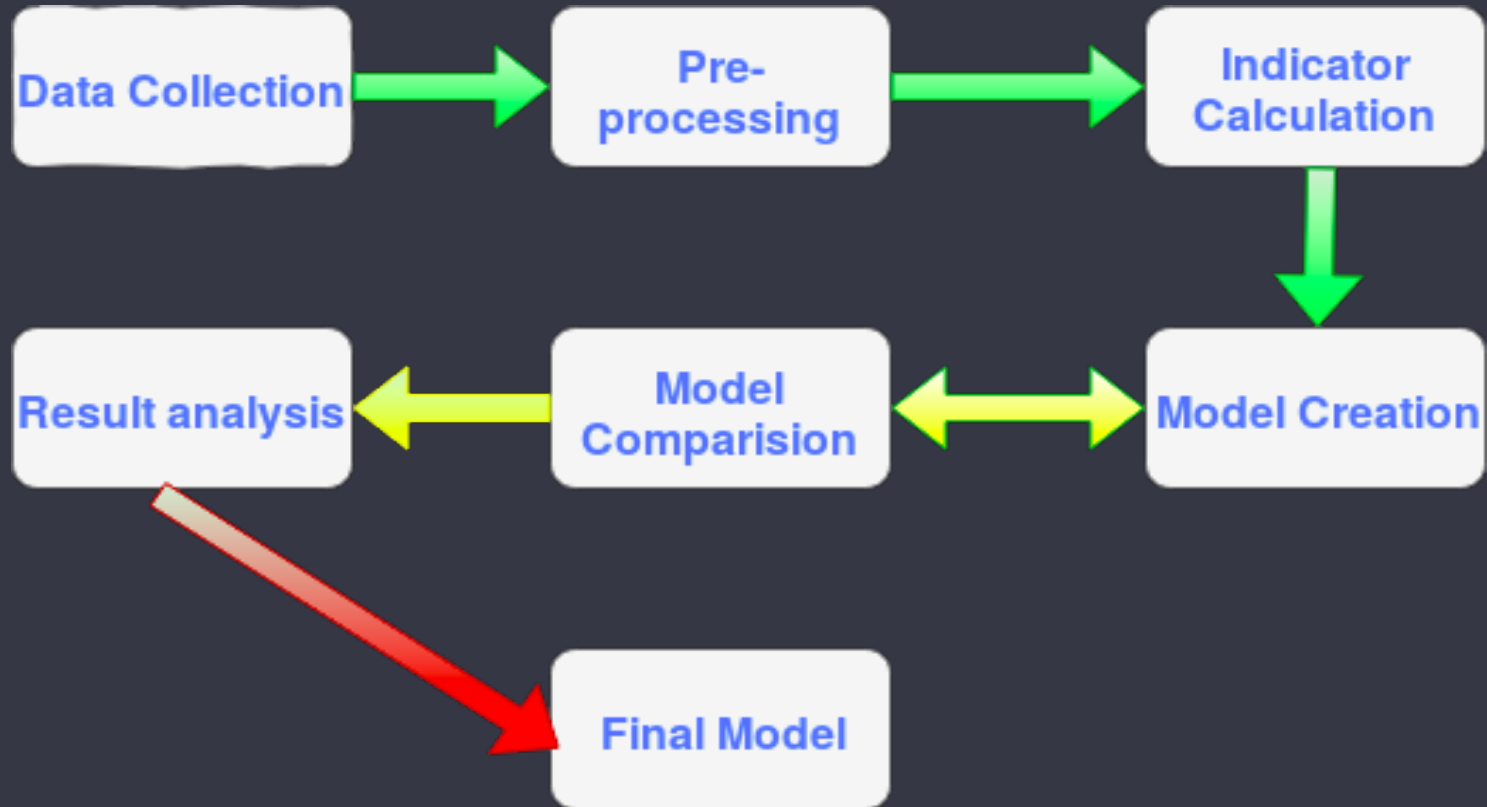
Literature Survey

- [1] Xienji Di. Stock Trend Prediction with Technical Indicators using SVM. SCPD Apple. 2011.

- [2] Jia-Yann Leu Jung-Hua Wang. Stock Market Trend Prediction Using ARIMA-based Neural Networks. International Conference on neural network. 2016.

- [3] Fangyan Dai Kai Chen Yi Zhou. A LSTM-based method for stock returns prediction: A case study of China stock market. IEEE International Conference on Big Data.2015.

- [4] M. Thenmozhi Manish Kumar. FORECASTING STOCK INDEX MOVEMENT: A COMPARISON OF SUPPORT VECTOR MACHINES AND RANDOM FOREST. Indian Institute of Capital Markets 9th Capital Markets Conference Paper. 2006.

- [5] M. Thenmozhi Manish Kumar. Stock Index Return Forecasting and Trading Strategy Using Hybrid ARIMA-Neural Network Model.

- [6] Luckyson Snehanshu Sudeepa. Predicting the direction of stock market prices using random forest. Applied Mathematical FInance. 2016.

Literature Survey(contd.)

Each paper gives a different idea. Some of them are:-

- Use of technical indicators.
- Varying hyperparameter to fit a model particularly in a dataset.
- Predicting over a longer interval rather than day-to-day prediction.
- Retraining model over and over again with a rolling window.
- Exponential smoothing of data.
- Creating ensemble of different models.
- Feature Selection.
- Time series analysis of data
- Making data stationary
- Use of ARIMA model

Workflow

```
Data Collection  →  Pre-processing  →  Indicator Calculation
                                              ↓
Result analysis  ←  Model Comparision  ↔  Model Creation

Result analysis  →  Final Model
```

Data Collection

Indian Dataset(after 2005).

- BSE SENSEX
- Nifty 50
- HDFC BANK
- ICICI BANK
- AXIS BANK
- YES BANK
- HSBC
- TATA MOTORS
- APPLE(USA)

All of the datas are available on in.investing.com free for any commercial and non-commercial use.

Any new data should must be saved in data/raw_data folder in csv format.
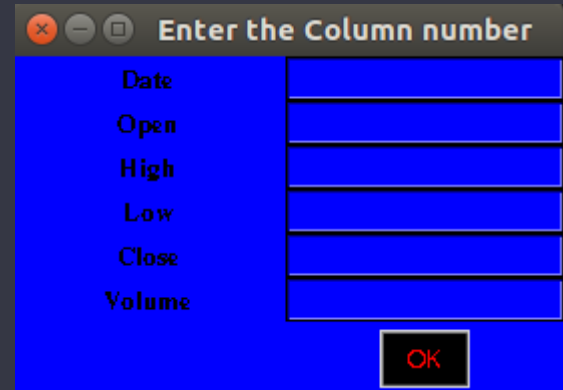
Data pre-processing

All files have Date, Open, High, Low, Close, Volume columns.

Enter the correct column number for Date | Day | Open | High | Low | Close | Volume.

The NA values are replaced to mean.

The pre-processed data is then saved to data/proc_data folder

Indicator Calculation

- Calculated using Ta-lib.
- Two types of indicator file yoc and ycc are generated.
- YOC files contain indicators for prediction of Open/Close type.
- YCC files contain indicators for prediction of Close/Close type.
- Total around 110 indicators generated using IndicatorCalculator class.
- Created new indicator, *Lag(a normalized comparison of today's closing vs n days earlier closing).*(Relevance?)
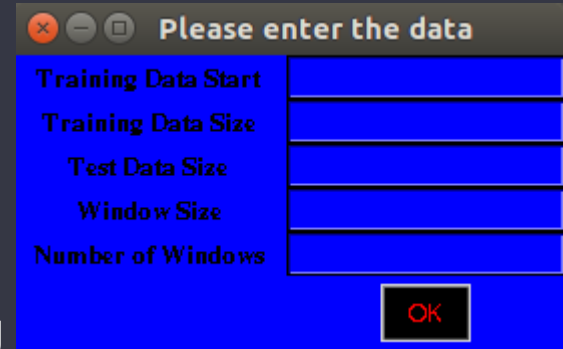- The calculated files are saved in indicators/ folder
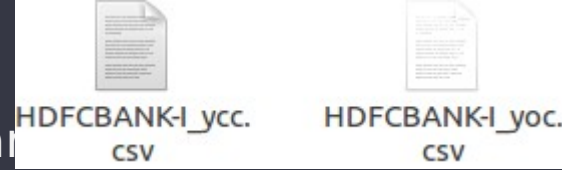
Feature creation and Selection

- All technical indicators(110) used as feature
- Left to Recursive Feature Elimination(RFE) for feature selection
- RFE eliminates features recursively based on weight of features
- Total 27 features are being used
- Lag1, Lag2, Lag5 is being selected repeatedly
- From YOC/YCC the particular 27 are selected for use

Model Creation

- LASSO and RIDGE
- Linear Discriminant Analysis
- Naive Bayes
- K Nearest Neighbour
- Support Vector Machine(RBF kernel) with GridSearchCV applied over C(penalty constant) and Gamma(Kernel coefficient)
- Random Forest with GridSearchCV(?) applied over number of estimators, minimum sample splits.
- Ensemble model (Random Forest + SVM)
- Long Short Term Memory

The data is passed to models by splitting in two parts training and testing

Model Comparison and Result analysis

- For comparison I used Nifty 50 data set.

Model	Accuracy
Lasso	55.3%
Ridge	54.2%
Naive Bayes	52.0%
K Nearest Neighbour	53.9%
Linear Discriminant Analysis	56.7%
Support Vector Machine	60.8%
Random Forest	63.7%
LSTM	63.1%
Ensemble Model(RF+SVM)	62.4%

Model Comparison and Result analysis(contd.)

- Except for Random forest and Support Vector Machine there is not much to do with these models.

- But with Random Forest and SVM we have several hyper-parameters to tune.

- Results vary as we vary the parameters.

- Hyperparameters control the tendency of overfit or underfit

Support Vector Machine

- Linear SVM just have penalty constant to vary but with kernelized SVM we also have kernel coefficient.
- Penalty constant determines, how much loss could we allow to classify a data point correctly.
- Kernel coefficient more or less decides the influence radius of data
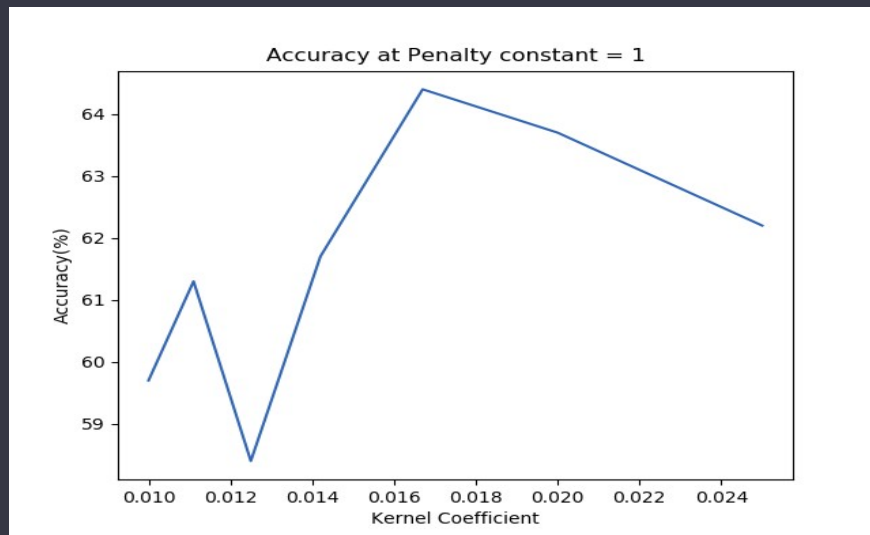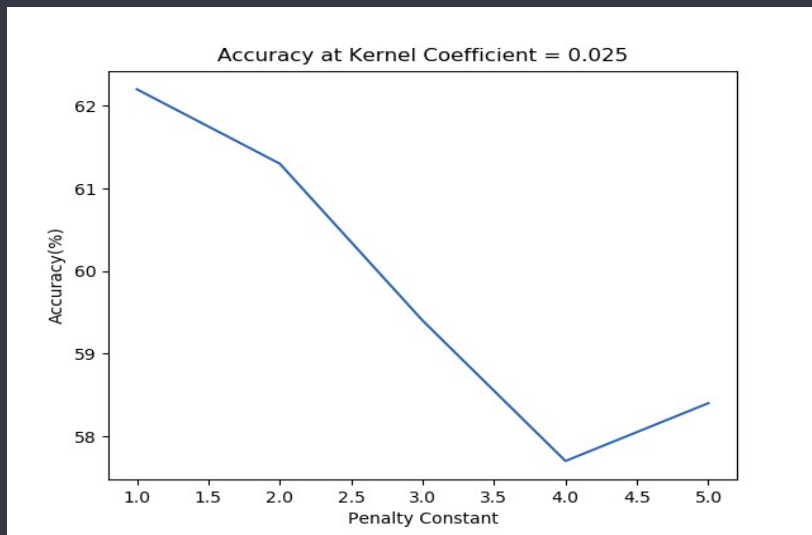
Random Forest

- There are lot of parameters to vary in Random Forest like number of estimators, minimum sample split, max depth, max features, max leaf nodes etc.
- Mainly number of estimators and minimum sample split affect the result, however extreme value of others may also affect.


Accuracy at minimum sample split = 6
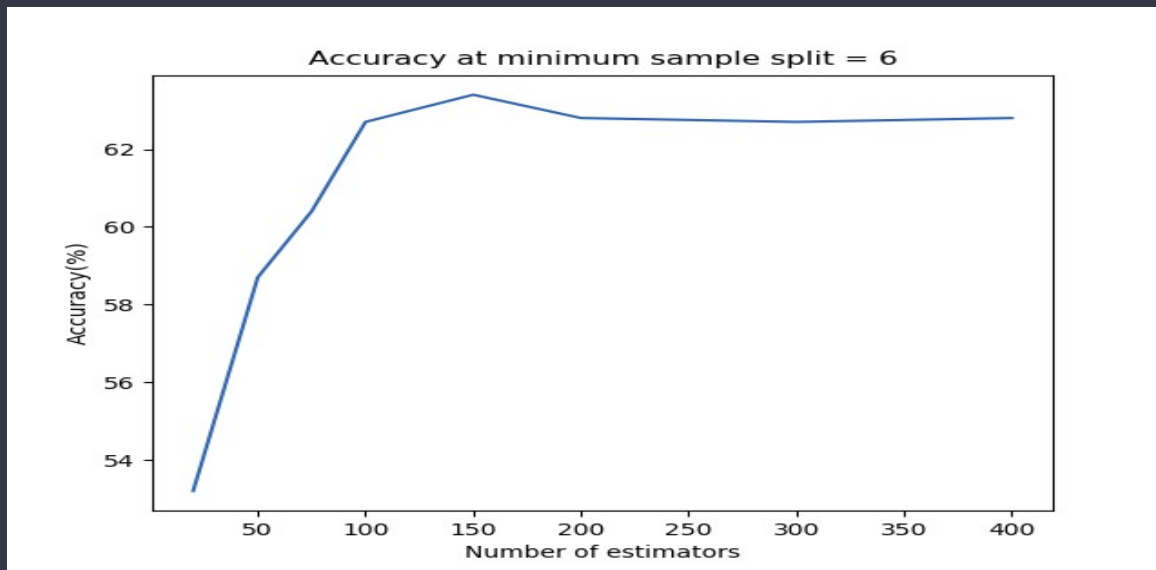
Ensemble Model

- Tried several like (RF+SVM, RF+LR, RF+SVM+LR, SVM+LR etc.)
- Accuracy less than the best of all models used .
- Hard voting/Soft Voting
- Accuracy slightly better with Soft voting
- Out of 100, only 78 classified, 62 accurate.
- Better return with hard voting

LSTM

- Two layer of LSTM (27 nodes ->300 nodes).
- Output layer, 1 node of Dense
- Option to store or forget information.
- Not required to train with indicators
- Easy to catch festival trends, seasonal market

Time-series property

- Ever growing data
- Variable mean and variance
- Non-stationary by Dickey-Fuller test
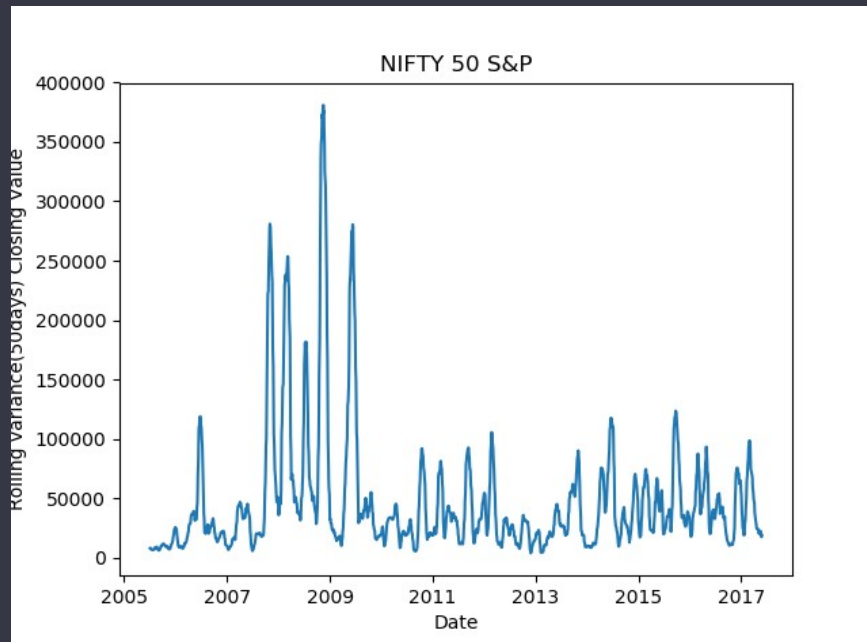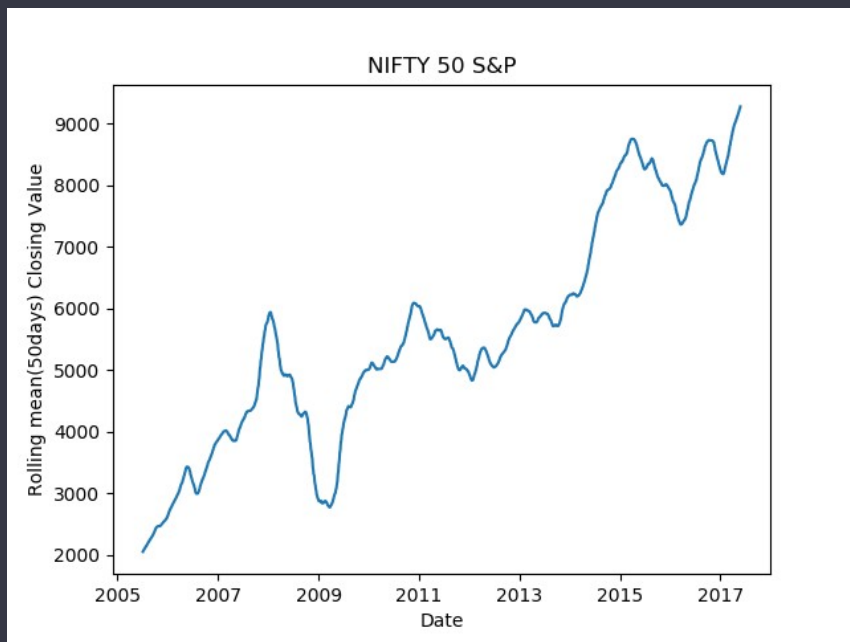
Stationary Data

- Zero mean and constant variance
- First/Second order difference to make data stationary
- Nifty data becomes stationary after first difference
- Dickey-Fuller test suggests same


NIFTY 50 S&P 1 Difference
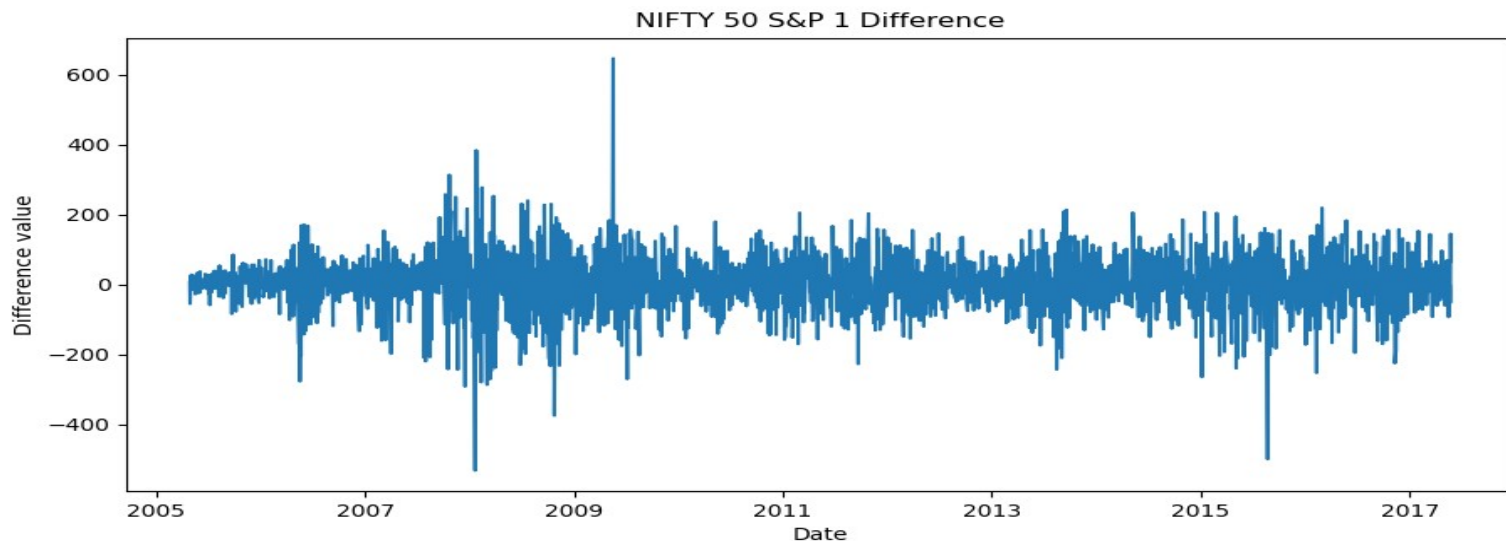
ARIMA model

- Three parameters
- Auto-regressive(p) :- Dependency on its own prior value
- Integrated(d) :- Difference order to make data stationary
- Moving Average(q) :- Dependency of observation on residual error
- Literature suggests trial and hit method to select (p,d,q)
- Not correct way
- Accuracy may be result of some other influencing factors
- Mathematical ways to select (p,d,q)
- (d) can be selected by doing Dickey Fuller test of data.
- (p) by Autocorrelation Function
- (q) by Partial Autocorrelation Function
- With proper (p,d,q), accuracy obtained around 67%

ACF and PACF

- ACF :- a bar chart of the coefficients of correlation between a time series data and priors of itself
- PACF :- Partial correlation between same
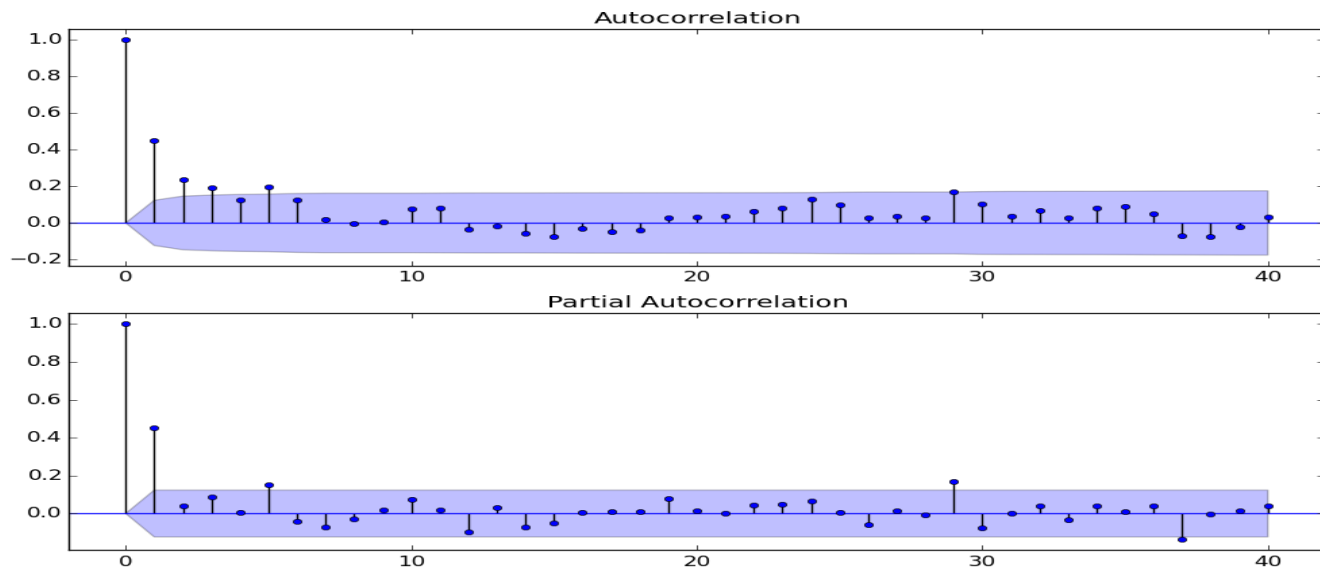
Flaws in research papers

- ill representation of accuracy(next slide)
- Noise-less data (Ex. Swiss stocks, Japan's stock) or time period selection for validation of model where data was stagnant. Same can't be generalized.
- Biased data, even 100% positive/negative prediction might give great results.
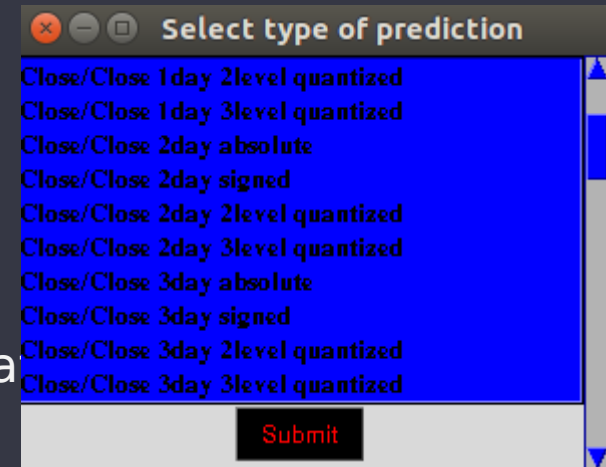- Impure data
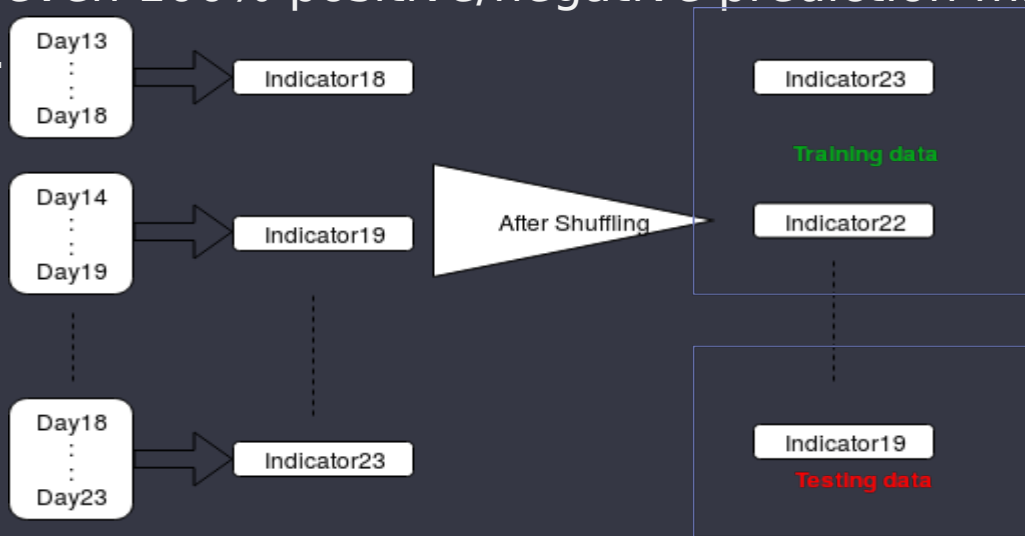
Does High Accuracy means benefits?

- Even the papers somehow show high accuracy, it may not certainly lead to benefits.
- Returns on each day may be different.
- Higher returns on wrong prediction and lower on correct.(maybe)
- Better to have false negatives than false positives.(?)
- False negative means lowering price(prediction), so you buy nothing you lose nothing
- False Positive means increasing price(prediction), so you buy but you have to sell at lower price
- Even a greater accuracy with high false positives may result into bad returns

Overcoming flaws

- Analyse results well
- Check for overfitting
- Check if result is biased(higher positives/nega
- Check for false positives and true positives
- Quantize the results for increasing true positive confidence(1/2/3 level)
- C

```
LDA_dist_actual_total        : 58.939 %,        41.061 %
LDA_dist_pred_train          : 74.833 %,        25.167 %
LDA_dist_pred_test           : 51.667 %,        48.333 %
LDA_accuracy_train_[T,+,-]   : 68.167 %,        70.156 %,        62.252 %
LDA_accuracy_test__[T,+,-]   : 60.000 %,        38.710 %,        82.759 %
```

```
Lasso_dist_actual_total        : 55.690 %,        44.310 %
Lasso_dist_pred_train          : 87.145 %,        12.855 %
Lasso_dist_pred_test           : 41.667 %,        58.333 %
Lasso_accuracy_train_[T,+,-]   : 58.765 %,        58.812 %,        58.442 %
Lasso_accuracy_test__[T,+,-]   : 58.333 %,        56.000 %,        60.000 %
```

Whats next?

- Using Garch Model try to predict in a less volatile time period
- Easing the process to predict quantized direction
- Creating a interface for real-time prediction
- Making tool capable of generating PDF report based on data performance
- Creating a business plan for investing of money considering the lowered risk

Github repo: *https://github.com/vikasboss/stock-market-trend-prediction*

Questions/Suggestions

Thanks!