Grow me the Money!

Padma Ram

Jigar Lotia

Roy Booker

Problem Statement

The stock investing market is quite vast and there are multitudes of different approaches that people use for their stock investing which could be based on Research, Rumours, Advice from family / friends, Professional Advice or in a modern day, Advice from Robots or Machines.

With the advice from the machine, there again are multiples of different models which are in play to predict the price movement on the stock but which one is suitable or performs better across the market? We don't know and we are seeking to find out.

Executive Summary

To analyse the S&P 500 trading data for the past 20 years using various machine learning models and provide a visual comparison and evaluation on how each model performs.

Concept

Description

This comparison aims to assess the Machine Learning Models against baseline performance strategy

Baseline performance:

Bollinger band strategy is built and included to compare statistical analysis to the machine learning models.

Machine Learning Models:

1.AdaBoost - which is a type of Ensemble Supervised Learning.
2.DecisionTreeClassifier - which is a type of Decision Tree Model
3.LogisticRegression - which is a type of Linear Regression Model.

To see which model is most effective for a short term investment (less the 5 years) and which performs best for a long term investment(5 years or more)

Approach

Technologies Used

- Os
- Pandas
- Backtesting
- backtesting.lib
- Finta
- Numpy
- lib
- Pathlib
- Pandas.tseries.offsets / dateOffset
- Matplotlib.pyplot
- Matplotlib.dates
- plotly.express
- Sklearn.linear_model, LogisticRegression
- sklearn.preprocessing, StandardScaler
- sklearn.metrics, classification_report
- sklearn, tree

Tasks

Padma Ram Took on the data Data Preparation, Bollinger Band analysis, Logistic regression analysis, Library and creation of our visualisation comparison

Jigar Lotia Prepared the Logistic regression model with a different processor analysis, the creation of our scope, parameter definition and Flow diagram.

Roy Booker created the Decision tree model analysis and the adaboost model analysis

Data Preparation

```
[48]: # Define a window size of 12
short_window = 12

# Create a simple moving average (SMA) using the short_window and assign this to a new columns called sma_fast
trading_df["sma_fast"] = trading_df["Price"].rolling(window=short_window).mean()

[49]: # Define a window size of 100
long_window = 100

# Create a simple moving average (SMA) using the long_window and assign this to a new columns called sma_slow
trading_df["sma_slow"] = trading_df["Price"].rolling(window=long_window).mean()
```

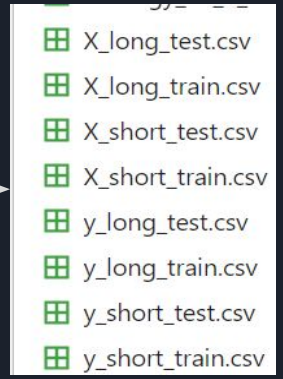
Create train and test data files

```
X_long_train.to_csv("../data/X_long_train.csv")
y_long_train.to_csv("../data/y_long_train.csv")

X_long_test.to_csv("../data/X_long_test.csv")
y_long_test.to_csv("../data/y_long_test.csv")

X_short_train.to_csv("../data/X_short_train.csv")
y_short_train.to_csv("../data/y_short_train.csv")

X_short_test.to_csv("../data/X_short_test.csv")
y_short_test.to_csv("../data/y_short_test.csv")
```



Data preparation notebook uses the historical price of S&P 500 Index for past 20 years. This data is sourced from Investing.com and is stored as S&P500_Data.csv.

The notebook performs the following pre-processing using the Pandas DataFrame :

- Parse Dates in the data file with parse_dates attribute
- Calculate daily percentage change using the pct_change function
- Dropping nulls using dropna function
- Preparing Features X (sma 12, sma 100) and Target y(signal)
- Slicing the data for training and testing based on below short term and long term

  Short term Training : 2015:2016
  Short term Testing : 2017:2022
  Long term Training : 2002:2006
  Long term Testing : 2007:2022

Model Evaluation

The baseline performance is built using the Bollinger long position trading strategy. The Performance metrics library **lib.py** is implemented to use in each machine learning model notebooks to evaluate the model performances such as Sharpe, Sortino, Cumulative, Maximum returns and annual volatility.

lib.py

```python
# Import
import pandas as pd
import numpy as np

def performance_metrics(df, strategy_name):
    metrics = [
    'Annualized Return',
    'Cumulative Returns',
    'Annual Volatility',
    'Sharpe Ratio',
    'Sortino Ratio',
    'Max Actual Return',
    'Max Strategy Return',
    'Max SReturn Lag'
    ]
```

```python
predictions_df = pd.DataFrame(index=X_test.index)
predictions_df["predicted_signal"] = lr_testing_signal_predictions
predictions_df["actual_returns"] = trading_df["actual_returns"]
# Calculate the  strategy returns
predictions_df["strategy_returns"] = predictions_df["actual_returns"] * predictions_df["pre
# Calculate the cumulative returns
predictions_df["cumulative_Returns"] = (
    1 + predictions_df["strategy_returns"]
).cumprod() - 1

predictions_df.head()

predictions_df.to_csv("../data/Strategy_LR_S_Term_Returns.csv")
```

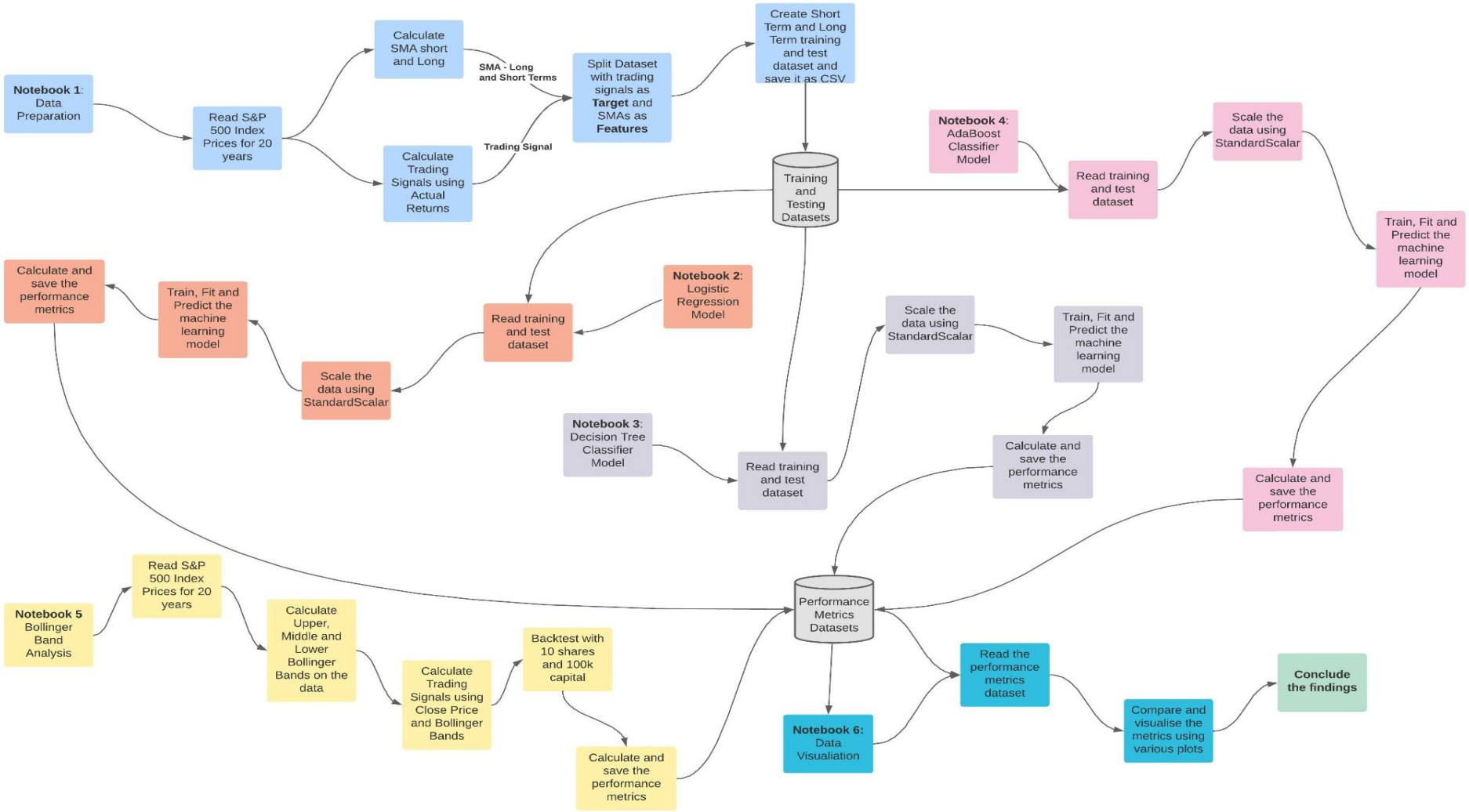
```python
evaluation_df = lib.performance_metrics(predictions_df, "Strategy_LR_S_Term")
evaluation_df.to_csv("../data/Strategy_LR_S_Term_Metrics.csv")
evaluation_df
```

	Strategy_LR_S_Term
Annualized Return	0.111104
Cumulative Returns	0.573461
Annual Volatility	3.95183
Sharpe Ratio	0.55113
Sortino Ratio	0.758395
Max Actual Return	1714.96054
Max Strategy Return	1720.540289
SS Lag	-0.207265
Max SReturn Lag	-5.579749

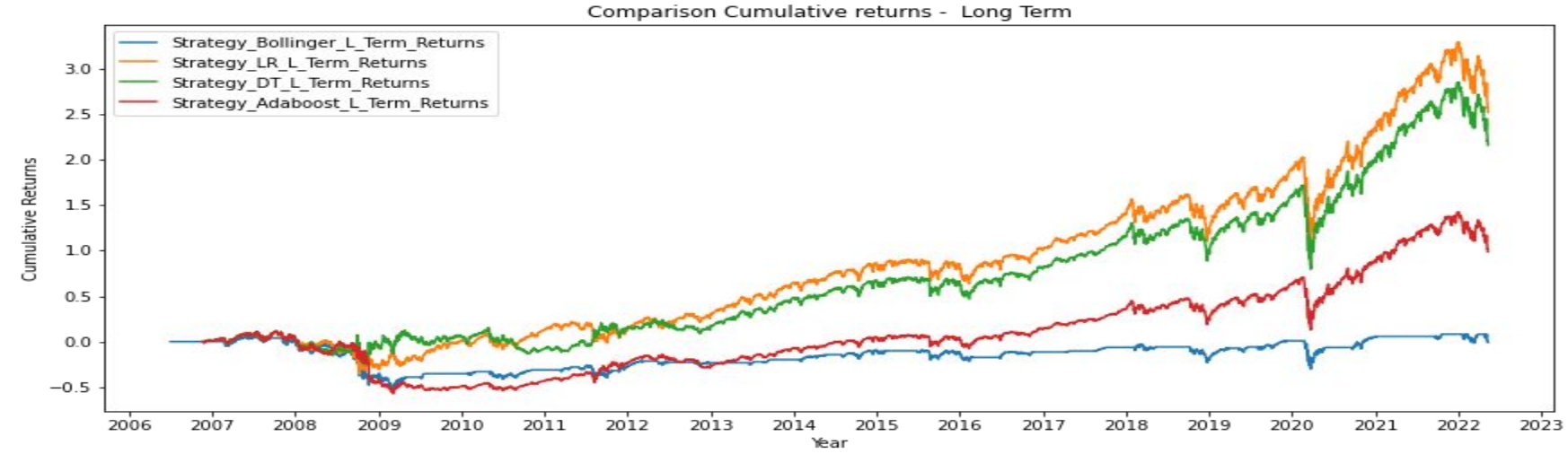
/ ... / Project-2-group-4 / data /

Name

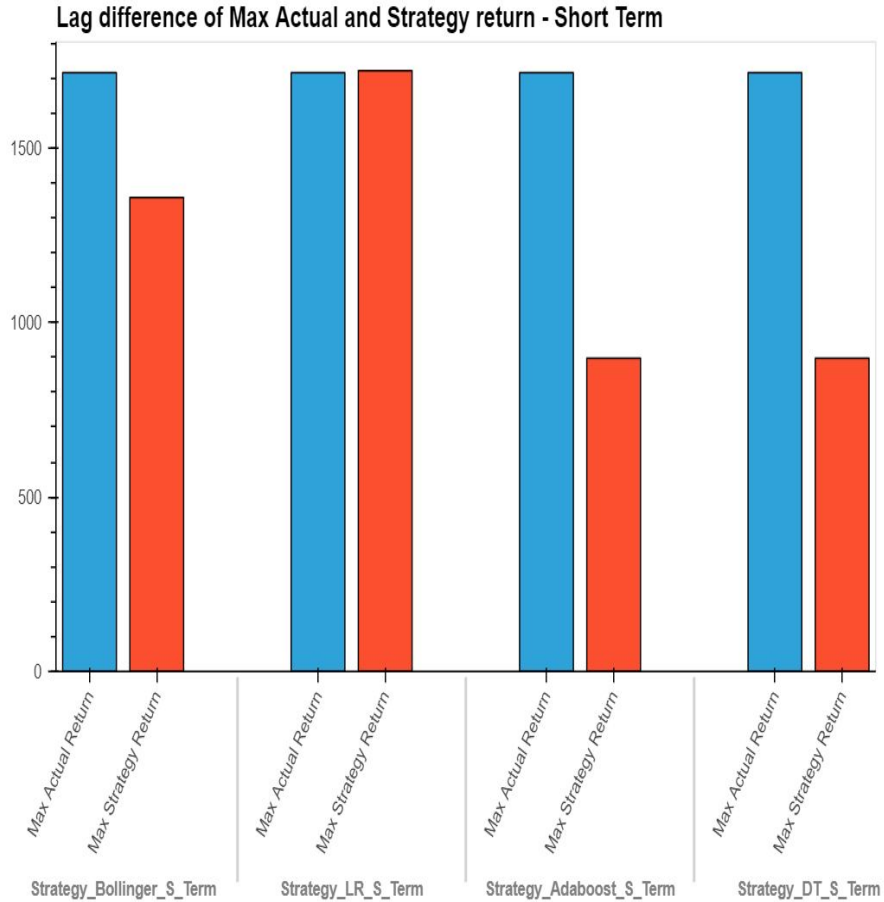
- SP500_Data.csv
- Strategy_Adaboost_L_Term_Metrics.csv
- Strategy_Adaboost_L_Term_Returns.csv
- Strategy_Adaboost_S_Term_Metrics.csv
- Strategy_Adaboost_S_Term_Returns.csv
- Strategy_Bollinger_L_Term_Metrics.csv
- Strategy_Bollinger_L_Term_Returns.csv
- Strategy_Bollinger_S_Term_Metrics.csv
- Strategy_Bollinger_S_Term_Returns.csv
- Strategy_DT_L_Term_Metrics.csv
- Strategy_DT_L_Term_Returns.csv
- Strategy_DT_S_Term_Metrics.csv
- Strategy_DT_S_Term_Returns.csv
- Strategy_LR_L_Term_Metrics.csv
- Strategy_LR_L_Term_Returns.csv
- Strategy_LR_S_Term_Metrics.csv
- Strategy_LR_S_Term_Returns.csv

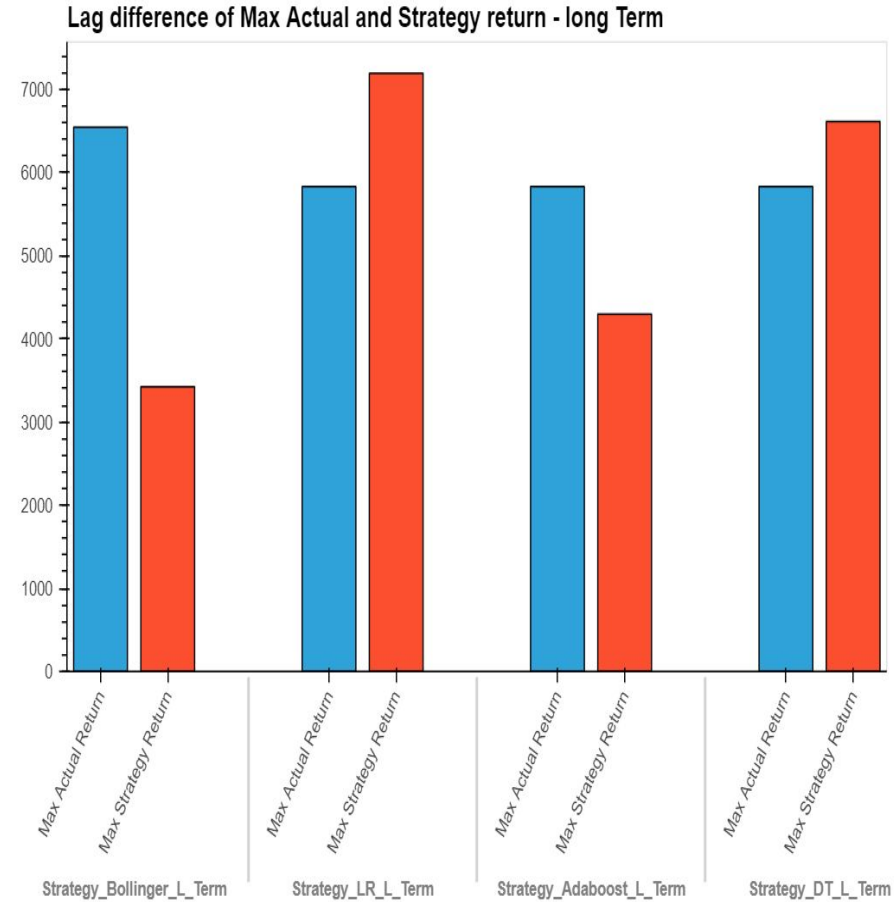

Notebook 1: Data Preparation
Read S&P 500 Index Prices for 20 years
Calculate SMA short and Long
Calculate Trading Signals using Actual Returns
SMA - Long and Short Terms
Trading Signal
Split Dataset with trading signals as Target and SMAs as Features
Create Short Term and Long Term training and test dataset and save it as CSV
Training and Testing Datasets
Notebook 4: AdaBoost Classifier Model
Read training and test dataset
Scale the data using StandardScalar
Train, Fit and Predict the machine learning model
Calculate and save the performance metrics
Train, Fit and Predict the machine learning model
Scale the data using StandardScalar
Read training and test dataset
Notebook 2: Logistic Regression Model
Notebook 3: Decision Tree Classifier Model
Read training and test dataset
Scale the data using StandardScalar
Train, Fit and Predict the machine learning model
Calculate and save the performance metrics
Calculate and save the performance metrics
Notebook 5 Bollinger Band Analysis
Read S&P 500 Index Prices for 20 years
Calculate Upper, Middle and Lower Bollinger Bands on the data
Calculate Trading Signals using Close Price and Bollinger Bands
Backtest with 10 shares and 100k capital
Calculate and save the performance metrics
Performance Metrics Datasets
Notebook 6: Data Visualiation
Read the performance metrics dataset
Compare and visualise the metrics using various plots
Conclude the findings

Visualisation Comparison

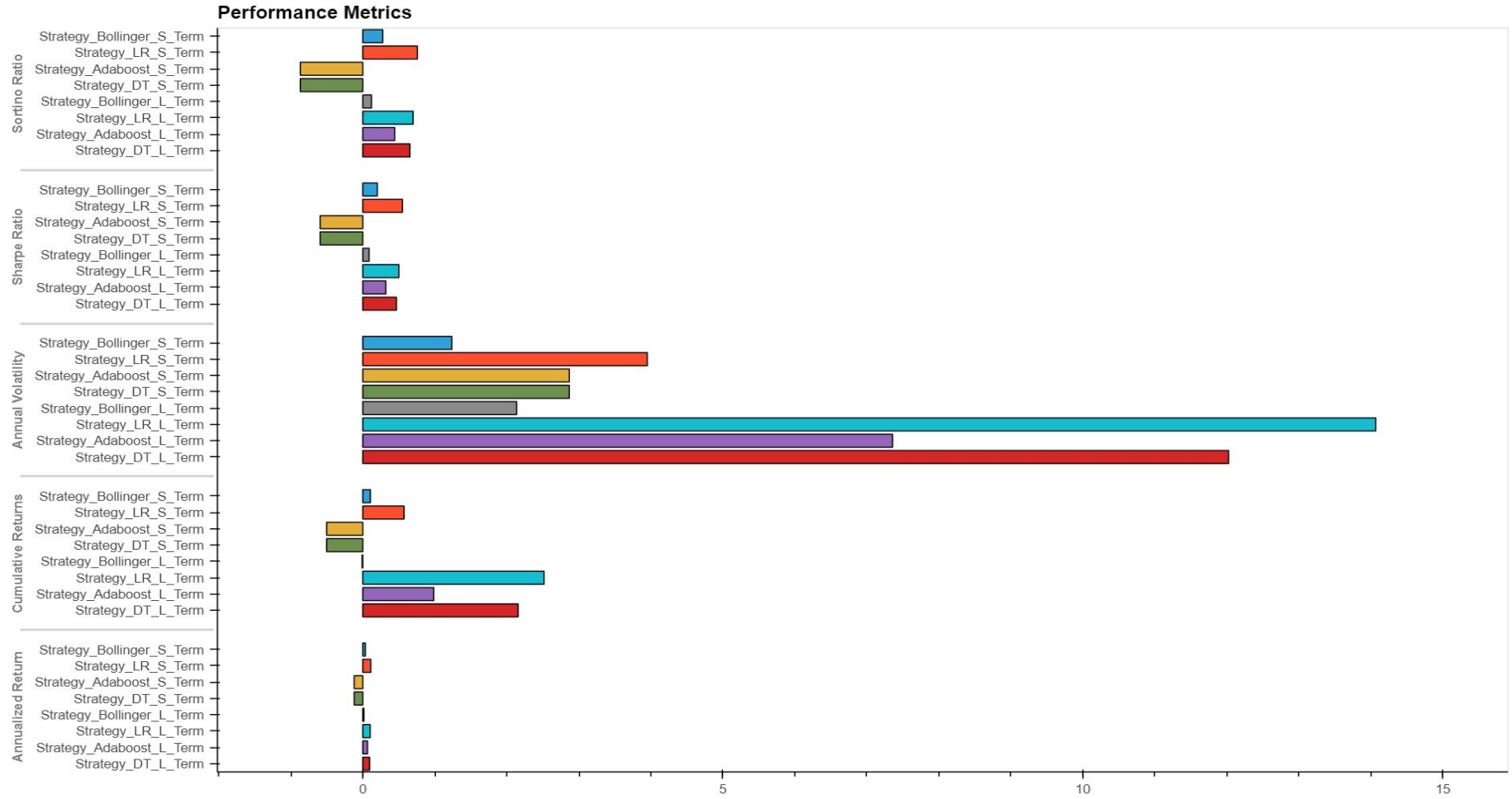


Visualisation Comparison





Visualisation Comparison



Next Steps

The most efficient model can then be used by the Amazon Lex based RoboAdvisor to advise users on how much returns they can expect on their initial investment.

RoboAdvisor is not part of this project due to the time constraints.

Other machine learning models or processing technique may need to be trained and tested periodically to assess new technologies or models. The models we completed were considered all that we had time for during the project.

Links

Our github link if any other information is required.

https://github.com/roybooker/Project-2-group-4

The S&P 500 data was taken from this link

https://au.investing.com/indices/us-spx-500


ANY QUESTIONS?