Clustering US Stock market data, finding if investing in any month is a Profitable strategy or not?

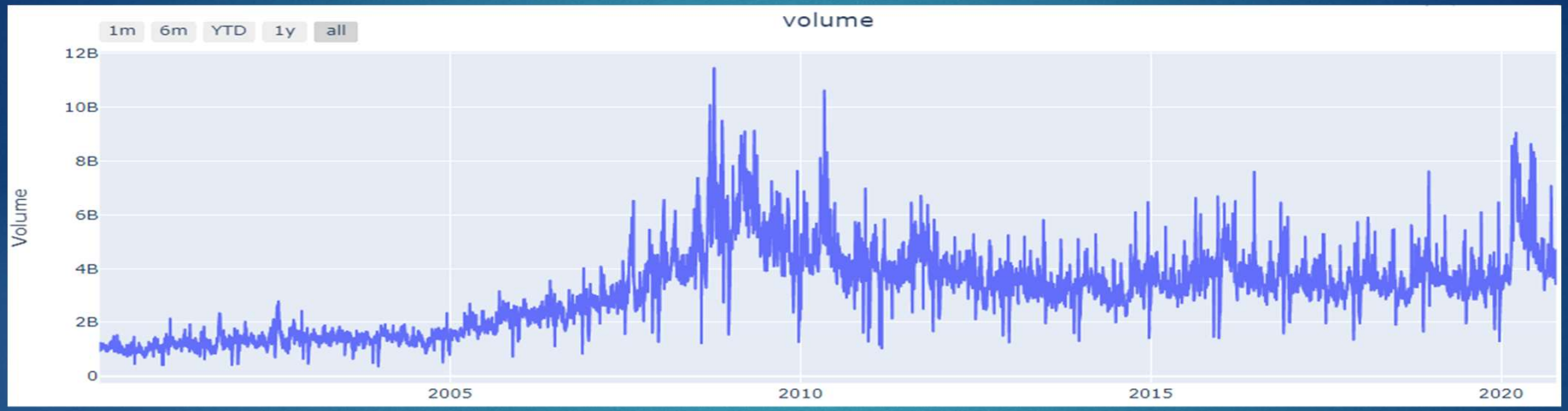
BY,

SIDHARTH.S

PROBLEM STATEMENTS

1. Examine & Identify the optimal number of clusters for daily volume data of U.S. stock market (2000-2020).
2. How can you cluster the Parameters (1, 2, 3) which are fractional values of Opening price, Closing price, Volume of Stock traded on daily basis?
3. Compute the monthly returns, Use decision tree to classify if investing in any month can be a profitable strategy.
4. What are the error metrics of your model?

1. Examining the Daily-Volume data

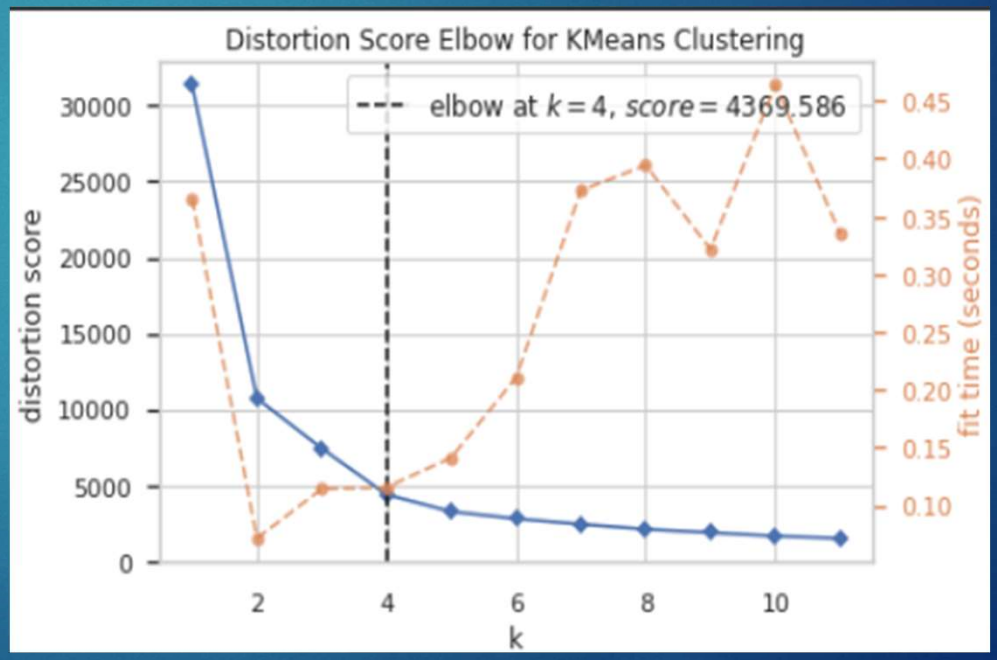


- Volume of stocks traded was very high between 2008 to 2011 and then again in 2020 (these were periods of **uncertainty** – Depression & Covid pandemic).
- Trading Volume range remained stable in the remaining period. It was in a higher range between 2011-2020 in comparison to 2000-2007.

1. Identifying the Optimal number of Clusters for the daily-Volume data

There are 2 important ways to identify them in Machine Learning:

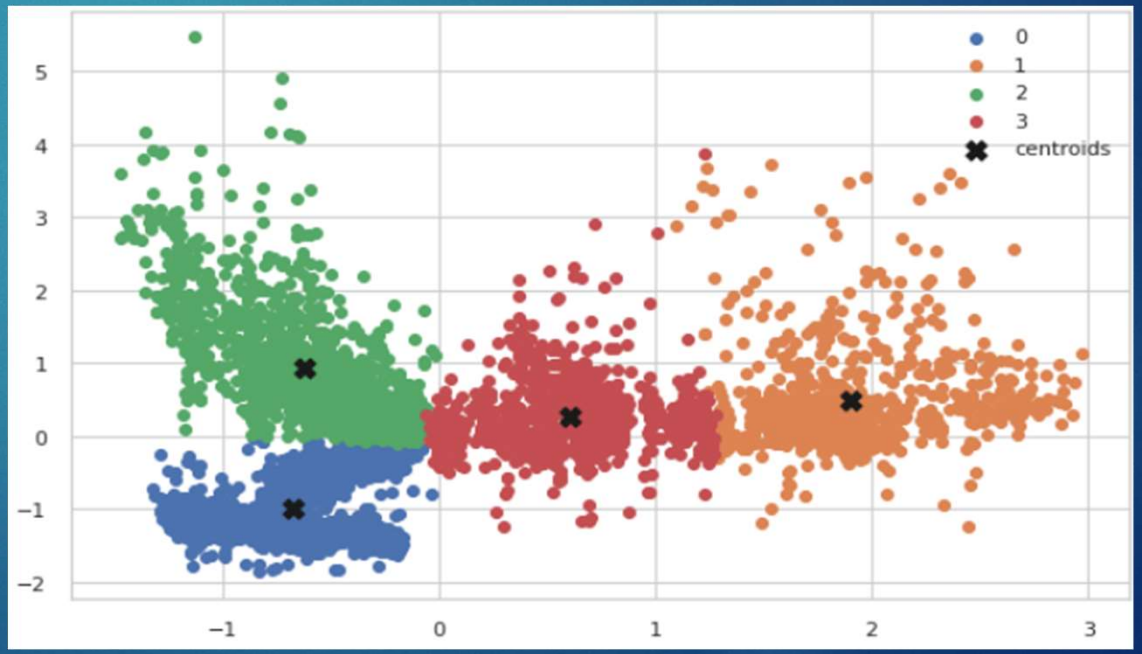
- Using K-means clustering **Elbow plot**:
  - For each value of K, we are calculating **WCSS** (Within-Cluster Sum of Square).
  - Optimal no : of cluster is where WCSS takes a sharp turn.
  - For our data, optimal cluster is identified to be **4**.



1. Identifying the Optimal number of Clusters for the daily-Volume data

- 2nd way is using **Silhouette score**:
  - We got a score of **0.51** which means that:
    - Clusters are not well separated from each other.
    - But there is Cohesion between data-points belonging to same cluster.
  - But overall the score is a good one. (range: -1 to +1)

Volume data (optimal clusters - 4)



2. Calculating the Parameters of **Business Value**

- **Fractional differencing** for Opening price, Closing price, Volume of stocks traded is calculated using the given formula.
- **Business value**:
  - Fractional differencing makes the series stationary (i.e. probability distribution does not change when shifted in time).
  - Hence, forecasting becomes more reliable.

Date	parameter-1	parameter-2	parameter-3
2000-01-03	-0.009549	0.009549	0.082850
2000-01-04	-0.009549	0.037979	0.082850
2000-01-05	-0.028796	-0.001848	-0.007033
2000-01-06	0.040267	-0.000958	-0.069553
2000-01-07	-0.000967	-0.027116	0.115405

The null values for the 1st row is backward filled.

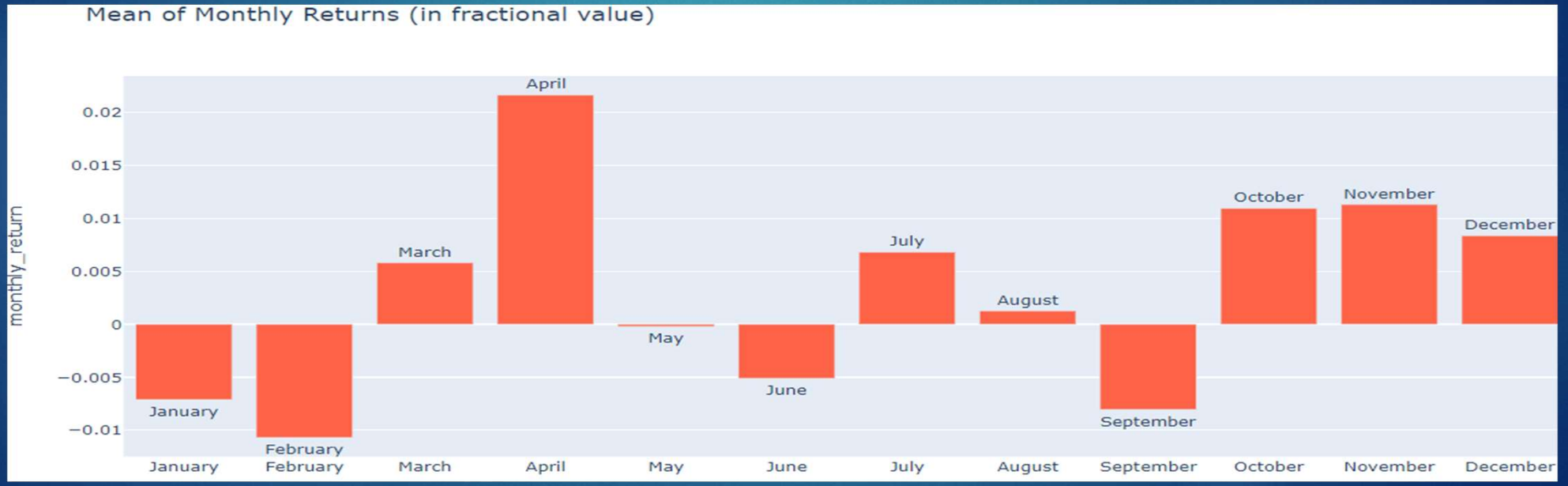
2. Clustering the Parameters (1, 2, 3)

- Using Scikit-learn ML package, **K-means** clustering algorithm can be applied to cluster the 3-parameters.
- The optimum no : of clusters is identified to be **5**.
- Since the data-points of these 3 parameters are **very close** to each other, clusters are not well separated from each other.

3-parameters clustered into 5 groups

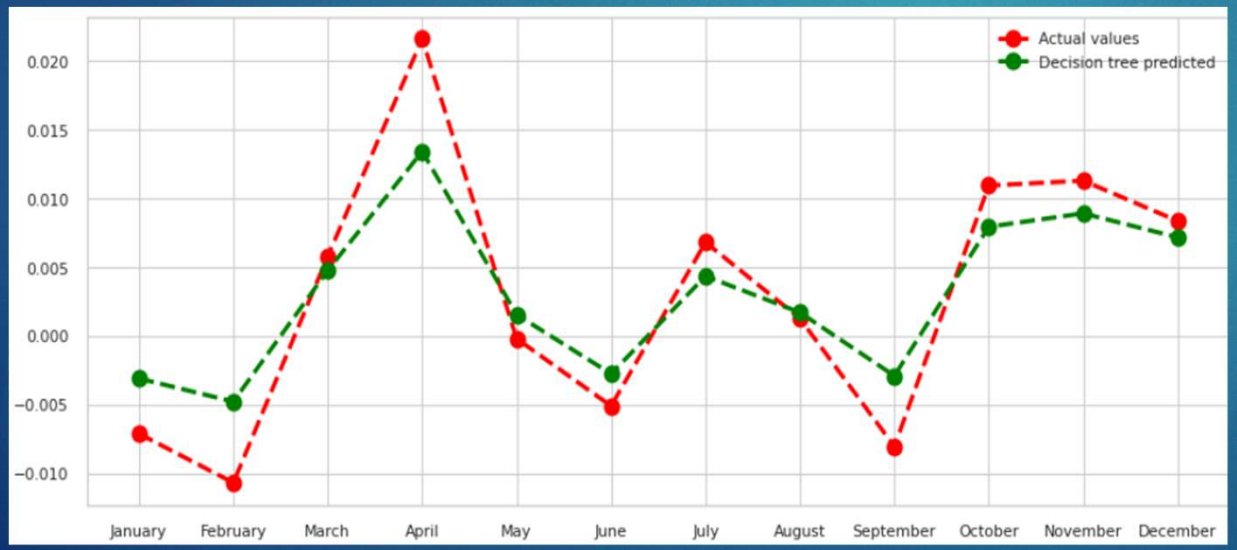
3. Computing the Monthly-Returns

- Extracting the Dates using USFederalHolidayCalendar, CustomBusinessMonthBegin & End (pandas modules), and then monthly_returns is calculated.
- **Grouping** the Months by taking **Mean of monthly_returns**.



3. Comparing the Actual-Returns with returns-predicted by Decision-tree model

- Taking only **50%** of the data for **training** the DT - Regressor model.
- **Predicting** the trained model on the **entire**(100%) dataset (for the sake of comparison).



- The values are **at peak** during **April**, least during February & September.
- From the graph, we can see that the Decision tree model trained using only 50% data is able to **capture the trends correctly**.

3. Converting the Business-statement into Data-science problem

- Converting the **Business statement** (investing in any month can be a profitable strategy or not) into **Data Science problem** (Binary-classification using DECISION TREE classifier algorithm).
  - If the predicted monthly-return is positive, then class is assigned the value **1**.
  - If the predicted monthly-return is negative, then class is assigned the value **0**.
- The model was trained on 80% data and tested on 20% data.
- **Accuracy of 70%** is achieved by this model (Decision Tree Classifier).

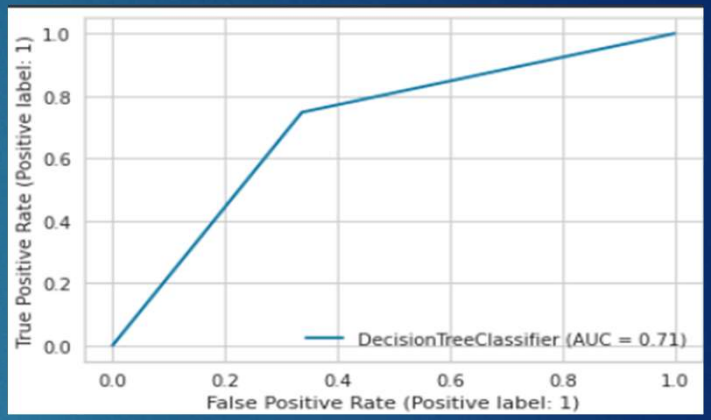
4. Error-Metrics of the Model

```
confusion matrix:
 [[427 217]
 [234 692]]
```

```
classification_report:
              precision    recall  f1-score   support

           0       0.65      0.66      0.65       644
           1       0.76      0.75      0.75       926

    accuracy                           0.71      1570
   macro avg       0.70      0.71      0.70      1570
weighted avg       0.71      0.71      0.71      1570
```



Confusion matrix:

- Data-points that are correctly predicted (1119/1570)
- Data-points that are wrongly predicted (451/1570)

Classification report:

- Precision & Recall for class-0 & class-1 : both are in good range.
- Class-1 is better predicted.
- Overall, Accuracy of 71% is achieved by this model.

AUC_ROC score :

- of 70% tells us that the model performs well.

Improvements that can be made

- **Hyper-parameter** tuning:
  - Using max_depth (limiting tree size)
  - Specifying minimum no : of samples in the leaf node.
- Using **Ensemble** techniques:
  - Bagging (Random Forest algorithm)
  - Boosting (XGBoost)
- Thus the model can be **refined continuously** to get better results.

THANKYOU