BAN 620 Course Project

STOCK MARKET PREDICTION ANALYSIS USING LOGISTIC REGRESSION

Team Members:

Prathyush Kaparthi - bs9845
Jithesh Kota - ti3491
RamaKrishna Saseendra Danthinada - qc7205
Sai Kalyan Veeramalli - nx9090
Sai Kumar D - wp6750

Date: 14 May, 2023

Project Summary:

This report is a summary of a data analysis project completed by a team of five individuals. The goal of the project was to analyze a dataset and predict the outcome variable "Target" as either profit or loss based on a variety of predictor variables.

The report provides an overview of the project, including the steps taken to obtain, explore, and preprocess the data, as well as the data mining task and partitioning techniques used. The report also outlines the two analysis methods used - logistic regression and classification tree - and presents the results and performance metrics for each method.

Overall, this report provides a comprehensive summary of the data analysis project, including the methodology used, the results obtained, and the conclusions drawn from the analysis.

Introduction:

Dataset Link: https://www.kaggle.com/datasets/luisandresgarcia/stock-market-prediction

Description:
We have procured the dataset from kaggle.com containing data regarding stock prices of various companies with 98192 rows and 77 columns initially.

Due to the rapid growth of urbanisation and various factors relating to the fluctuating market in our everyday life we would like to step forward and predict how the future Stocks and shares will be affected by the past market and would like to leverage the outcome of this project for the future investments.

We will be using various methods and models to predict the outcome variable and also we will be using different classification methods like Classification trees and Boosted trees.

Main Chapter

1. Develop Understanding:

   Our project goal is to help investors by predicting the future outcomes whether a company will be facing a loss or profit.

   Our analysis will be based on different factors where we will be considering every minute factor for the company where and how it will be affecting the companies rise and fall of revenue and how it's going to affect the stock market.

2. ***Explore, Clean and Preprocess Data; Reduce Data Dimension***

- The data we acquired has 841 rows with 41 columns, but some of the 41 columns are redundant or irrelevant as predictors.
- Initially, the shape of dataset before dropping unwanted rows and columns is (98192, 77) i.e., 98192 rows and 77 columns

```
Out[983]: (98192, 77)
```

The columns present at the initial phase are:

```
In [5]: df.columns
Out[5]: Index(['company', 'age', 'market', 'year', 'month', 'day', 'hour', 'minute',
               'volume', 'high', 'low', 'close', 'open', 'AVERAGE_SMA_3_CLOSE',
               'EMA_3_CLOSE', 'MACD_3_CLOSE', 'AVERAGE_SMA_3_OPENHIGH',
               'EMA_3_OPENHIGH', 'MACD_3_OPENHIGH', 'AVERAGE_SMA_3_HIGHLOW',
               'EMA_3_HIGHLOW', 'MACD_3_HIGHLOW', 'AVERAGE_SMA_3_VOLUME',
               'EMA_3_VOLUME', 'MACD_3_VOLUME', 'AVERAGE_SMA_4_CLOSE', 'EMA_4_CLOSE',
               'MACD_4_CLOSE', 'AVERAGE_SMA_4_OPENHIGH', 'EMA_4_OPENHIGH',
               'MACD_4_OPENHIGH', 'AVERAGE_SMA_4_HIGHLOW', 'EMA_4_HIGHLOW',
               'MACD_4_HIGHLOW', 'AVERAGE_SMA_4_VOLUME', 'EMA_4_VOLUME',
               'MACD_4_VOLUME', 'AVERAGE_SMA_7_CLOSE', 'EMA_7_CLOSE', 'MACD_7_CLOSE',
               'AVERAGE_SMA_7_OPENHIGH', 'EMA_7_OPENHIGH', 'MACD_7_OPENHIGH',
               'AVERAGE_SMA_7_HIGHLOW', 'EMA_7_HIGHLOW', 'MACD_7_HIGHLOW',
               'AVERAGE_SMA_7_VOLUME', 'EMA_7_VOLUME', 'MACD_7_VOLUME',
               'AVERAGE_SMA_20_CLOSE', 'EMA_20_CLOSE', 'MACD_20_CLOSE',
               'AVERAGE_SMA_20_OPENHIGH', 'EMA_20_OPENHIGH', 'MACD_20_OPENHIGH',
               'AVERAGE_SMA_20_HIGHLOW', 'EMA_20_HIGHLOW', 'MACD_20_HIGHLOW',
               'AVERAGE_SMA_20_VOLUME', 'EMA_20_VOLUME', 'MACD_20_VOLUME',
               'AVERAGE_SMA_50_CLOSE', 'EMA_50_CLOSE', 'MACD_50_CLOSE',
               'AVERAGE_SMA_50_OPENHIGH', 'EMA_50_OPENHIGH', 'MACD_50_OPENHIGH',
               'AVERAGE_SMA_50_HIGHLOW', 'EMA_50_HIGHLOW', 'MACD_50_HIGHLOW',
               'AVERAGE_SMA_50_VOLUME', 'EMA_50_VOLUME', 'MACD_50_VOLUME',
               'DAYS_UNTIL_END_OF_MONTH', 'DAYS_UNTIL_END_OF_TRIMESTER', 'DINAMIC3',
               'TARGET'],
              dtype='object')
```

- Total number of null values found in the dataset are 60378 which are dropped using ‘df.dropna()’ method.

```
In [7]: df.isnull().sum().sum()
Out[7]: 60378
```

- The data layout after dropping rows with null values are:

```
In [9]: df
```

Out[9]:

	company	age	market	year	month	day	hour	minute	volume	high	...	AVERAGE_SMA_50_HIGHLOW	EMA_50_HIGHLOW	MACD_50_HIGHLO
21	CNDT	21.0	NASDAQ	2021	11	10	15	30	1276090	6168.00	...	0.2	240000.0	10000
22	CNDT	22.0	NASDAQ	2021	11	9	15	30	2271889	6.20	...	0.2	240000.0	10000
23	CNDT	23.0	NASDAQ	2021	11	8	15	30	1670784	6.34	...	0.2	240000.0	0
24	CNDT	24.0	NASDAQ	2021	11	5	14	30	4124330	6.74	...	0.2	240000.0	10000
25	CNDT	25.0	NASDAQ	2021	11	4	14	30	1129657	7.21	...	0.2	230000.0	20000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
14448	CSTR	84.0	NASDAQ	2021	8	12	15	30	32863	21.23	...	0.7	680000.0	-10000
14449	CSTR	85.0	NASDAQ	2021	8	11	15	30	28098	21.20	...	0.7	680000.0	-10000
14450	CSTR	86.0	NASDAQ	2021	8	10	15	30	33961	21.28	...	0.7	680000.0	-20000
14451	CSTR	87.0	NASDAQ	2021	8	9	15	30	25654	21.34	...	0.7	680000.0	-20000
14452	CSTR	88.0	NASDAQ	2021	8	6	15	30	25459	21.48	...	0.7	680000.0	-40000

10385 rows × 77 columns

- The columns named ‘Company’ and ‘Market’ in our dataset contain non-numeric data, we cannot directly use them as predictors in our analysis. To use these columns in our analysis, we must convert them into numeric data using a technique called dummy variable encoding.

- Data types before conversion of data:

```
df.dtypes
```

```
company                          category
age                               float64
market                           category
year                                int64
month                               int64
                                   ...
MACD_50_VOLUME                    float64
DAYS_UNTIL_END_OF_MONTH           float64
DAYS_UNTIL_END_OF_TRIMESTER       float64
DINAMIC3                          float64
TARGET                            float64
Length: 77, dtype: object
```

```
|: df.market = df.market.astype('category')

   # Display category classes and category type.
   print(' ')
   print('Category levels and changed variable type:')
   print(df.market.cat.categories)  # It can take one of three classes.
   print(df.market.dtype)  # Type is now 'category'.
```

```
Category levels and changed variable type:
Index(['NASDAQ'], dtype='object')
category
```

- Now, we use the ‘category_encoders’ library to perform ordinal encoding on ‘Company’ column. Using tolist() method, we extract all the values from the ‘Company’ column and insert data into a list. Finally, fit_transform() encodes the 'company' column of data and returns a new DataFrame with integer values representing the encoded categories.

```
In [16]: import pandas as pd
         import category_encoders as ce
         data = pd.DataFrame({
             'company' : df['company'].tolist()
         })
         # create an object of the OrdinalEncoding
         ce_ordinal = ce.OrdinalEncoder(cols=['company'])
         # fit and transform and you will get the encoded data
         ce_ordinal.fit_transform(data)
```

Out[16]:

	company
0	1
1	1
2	1
3	1
4	1
...	...
10381	58
10382	58
10383	58
10384	58

click to scroll output; double click to hide

10385 rows × 1 columns

- This code concatenates df with a new DataFrame created by performing the ordinal encoding on the 'Company' column from data. The resulting DataFrame will have all the columns from df, plus a new column for the encoded values of ‘Company’.

```
[18]: frames = [df, ce_ordinal.fit_transform(data) ]

      result = pd.concat(frames)
```

```
[19]: result
```

[19]:

	company	age	market	year	month	day	hour	minute	volume	high	...	AVERAGE_SMA_50_HIGHLOW	EMA_50_HIGHLOW	MACD_50_HIGHL
21	CNDT	21.0	NASDAQ	2021.0	11.0	10.0	15.0	30.0	1276090.0	6168.00	...	0.2	240000.0	1000
22	CNDT	22.0	NASDAQ	2021.0	11.0	9.0	15.0	30.0	2271889.0	6.20	...	0.2	240000.0	1000
23	CNDT	23.0	NASDAQ	2021.0	11.0	8.0	15.0	30.0	1670784.0	6.34	...	0.2	240000.0	
24	CNDT	24.0	NASDAQ	2021.0	11.0	5.0	14.0	30.0	4124330.0	6.74	...	0.2	240000.0	1000
25	CNDT	25.0	NASDAQ	2021.0	11.0	4.0	14.0	30.0	1129657.0	7.21	...	0.2	230000.0	2000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
10380	58	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
10381	58	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
10382	58	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
10383	58	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N
10384	58	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	N

20770 rows × 77 columns

Data types after conversion:

```
30]: df.dtypes

30]: age                              float64
     year                               int64
     month                              int64
     day                                int64
     hour                               int64
                                        ...
     DAYS_UNTIL_END_OF_MONTH          float64
     DAYS_UNTIL_END_OF_TRIMESTER      float64
     DINAMIC3                         float64
     TARGET                           float64
     company_ordinal                  float64
     Length: 76, dtype: object
```

3.

A. Determine the Data Mining Task

Data mining task are typically grouped into two categories : descriptive and predictive

Here we have predicted the binary variable “Target” for which it decides a profit or loss depending on various factors i.e; predictors that are mentioned in the data set.

In the data set we have taken there is one outcome variable **“Target”** .

This is a binary classification variable, therefore the task is to predict whether the result is either profit or loss.

Here are the predictor variables:

['age', 'year', 'month', 'day', 'hour', 'minute', 'volume', 'high',
'low', 'close', 'open', 'AVERAGE_SMA_3_CLOSE', 'EMA_3_CLOSE',
'MACD_3_CLOSE', 'AVERAGE_SMA_3_OPENHIGH', 'EMA_3_OPENHIGH',
'MACD_3_OPENHIGH', 'AVERAGE_SMA_3_HIGHLOW', 'EMA_3_HIGHLOW',
'MACD_3_HIGHLOW', 'AVERAGE_SMA_3_VOLUME', 'EMA_3_VOLUME',
'MACD_3_VOLUME', 'AVERAGE_SMA_4_CLOSE', 'EMA_4_CLOSE', 'MACD_4_CLOSE',
'AVERAGE_SMA_4_OPENHIGH', 'EMA_4_OPENHIGH', 'MACD_4_OPENHIGH',
'AVERAGE_SMA_4_HIGHLOW', 'EMA_4_HIGHLOW', 'MACD_4_HIGHLOW',
'AVERAGE_SMA_4_VOLUME', 'EMA_4_VOLUME', 'MACD_4_VOLUME',
'AVERAGE_SMA_7_CLOSE', 'EMA_7_CLOSE', 'MACD_7_CLOSE',
'AVERAGE_SMA_7_OPENHIGH', 'EMA_7_OPENHIGH', 'MACD_7_OPENHIGH',
'AVERAGE_SMA_7_HIGHLOW', 'EMA_7_HIGHLOW', 'MACD_7_HIGHLOW',
'AVERAGE_SMA_7_VOLUME', 'EMA_7_VOLUME', 'MACD_7_VOLUME',
'AVERAGE_SMA_20_CLOSE', 'EMA_20_CLOSE', 'MACD_20_CLOSE',
'AVERAGE_SMA_20_OPENHIGH', 'EMA_20_OPENHIGH', 'MACD_20_OPENHIGH',
'AVERAGE_SMA_20_HIGHLOW', 'EMA_20_HIGHLOW', 'MACD_20_HIGHLOW',
'AVERAGE_SMA_20_VOLUME', 'EMA_20_VOLUME', 'MACD_20_VOLUME',
'AVERAGE_SMA_50_CLOSE', 'EMA_50_CLOSE', 'MACD_50_CLOSE',
'AVERAGE_SMA_50_OPENHIGH', 'EMA_50_OPENHIGH', 'MACD_50_OPENHIGH',
'AVERAGE_SMA_50_HIGHLOW', 'EMA_50_HIGHLOW', 'MACD_50_HIGHLOW',
'AVERAGE_SMA_50_VOLUME', 'EMA_50_VOLUME', 'MACD_50_VOLUME',
'DAYS_UNTIL_END_OF_MONTH', 'DAYS_UNTIL_END_OF_TRIMESTER', 'DINAMIC3',
'company_ordinal']

B. Partition Data

To prevent overfitting, it is common practice to split the dataset into partitions using the train_test_split function with a test size of 40%. This splits the data into a training set, which contains 60% of the data and is used to develop the model, and a validation set, which contains 40% of the data and is used to evaluate the model's performance on new data.

```
In [312]: train_X.shape
Out[312]: (30360, 75)

In [316]: valid_X.shape
Out[316]: (20241, 75)
```

C. Techniques

a) Since the outcome variable is categorical, we have selected logistic regression and classification tree as our analysis methods. The main advantage of classification models is their ability to predict categorical outcomes, while logistic regression models are effective in modeling the probability of a binary outcome.

a) ***Algorithm and Measures***

In this scenario we have used the LogisticRegression() function to fit multiple predictors logistic regression for training partition.

Parameters of Logistic Regression Model with Multiple Predictors :

```
Parameters of Logistic Regresion Model with Multiple Predictors
Intercept: -4e-09
Coefficients for Predictors
                  age      year         month           day          hour  \
Coeff: -4.219000e-07 -0.000008 -2.700000e-08 -6.210000e-08 -5.900000e-08

              minute        volume          high           low         close  \
Coeff: -1.187000e-07  1.680000e-08 -4.966000e-07 -1.180000e-07 -7.395000e-07

        ...  AVERAGE_SMA_50_HIGHLOW  EMA_50_HIGHLOW  MACD_50_HIGHLOW  \
Coeff:  ...                    -0.0       -0.000002         0.000005

        AVERAGE_SMA_50_VOLUME  EMA_50_VOLUME  MACD_50_VOLUME  \
Coeff:          -3.640000e-07            0.0            -0.0

        DAYS_UNTIL_END_OF_MONTH  DAYS_UNTIL_END_OF_TRIMESTER      DINAMIC3  \
Coeff:            -7.120000e-08                -7.120000e-08 -4.319000e-07

        company_ordinal
Coeff:    -7.674000e-07
```

Going forward we have found the first 20 classifications for the validation partition

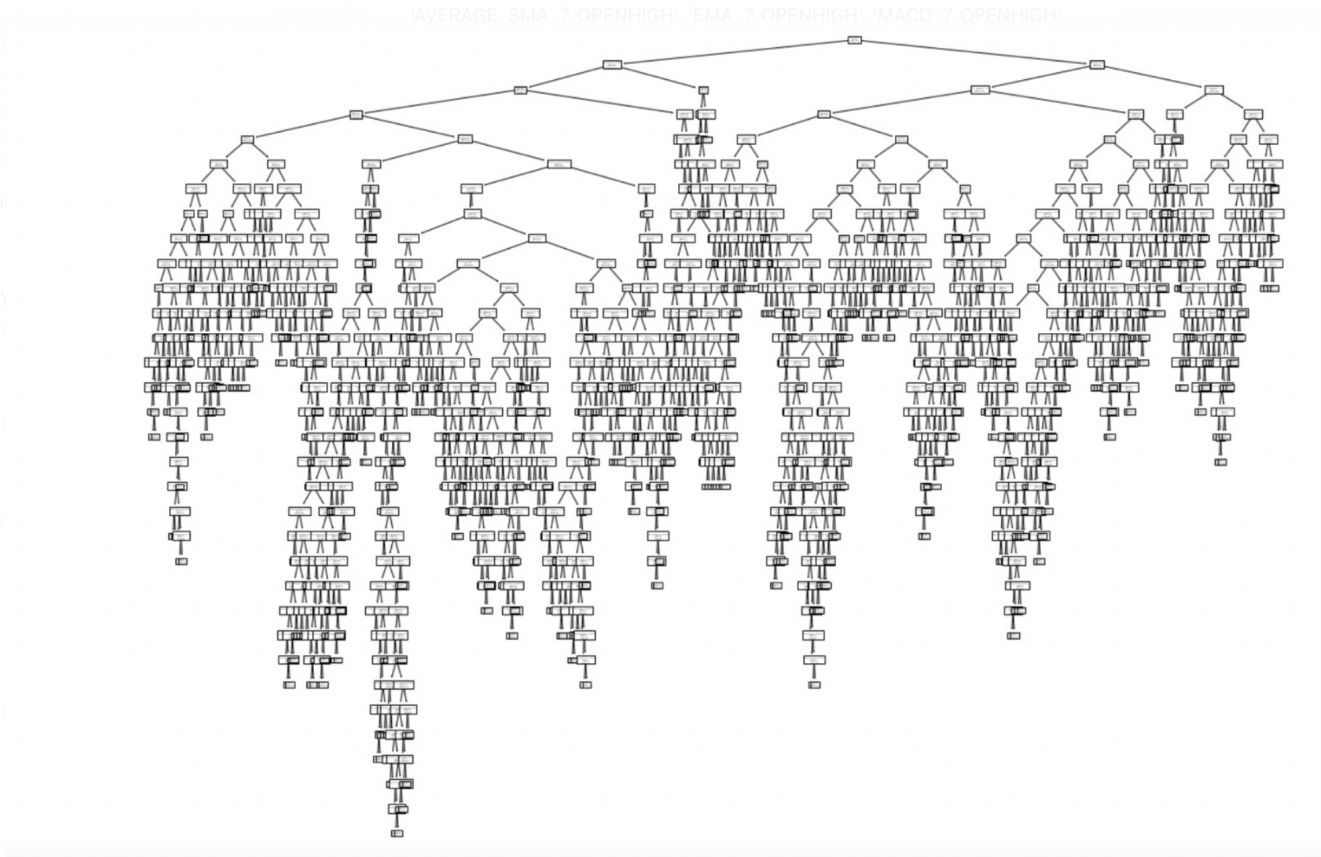
```
Classification for Validation Partition
      Actual  Classification    p(0)    p(1)
7351     0.0             0.0  0.9953  0.0047
7106     0.0             0.0  1.0000  0.0000
7018     0.0             0.0  0.9997  0.0003
5084     0.0             0.0  0.9993  0.0007
4836     0.0             0.0  0.8526  0.1474
4076     0.0             0.0  0.5557  0.4443
2632     0.0             0.0  0.9292  0.0708
1945     0.0             0.0  0.8494  0.1506
3373     0.0             0.0  0.9809  0.0191
7190     0.0             0.0  1.0000  0.0000
2928     0.0             0.0  0.9307  0.0693
1793     0.0             0.0  0.9962  0.0038
4154     0.0             0.0  0.5732  0.4268
968      0.0             0.0  0.6274  0.3726
3954     0.0             0.0  0.5771  0.4229
4302     0.0             0.0  0.7892  0.2108
3727     0.0             1.0  0.2340  0.7660
3422     0.0             0.0  0.9176  0.0824
2842     0.0             0.0  0.7522  0.2478
186      0.0             0.0  0.8963  0.1037
```

	age	year	month	day	hour	minute	volume	high	low	close	...	EMA_50_HIGHLOW	MACD_50_HIGHLOW	AVERAGE_SMA_50_VOLUME	EMA_
0	21.0	2021	11	10	15	30	1276090	6168.00	5.91	5.91	...	240000.0	10000.0	1119416.0	-1
1	22.0	2021	11	9	15	30	2271889	6.20	5.80	6.06	...	240000.0	10000.0	1133273.3	-5
2	23.0	2021	11	8	15	30	1670784	6.34	6.06	6.10	...	240000.0	0.0	1104294.2	4
3	24.0	2021	11	5	14	30	4124330	6.74	6.15	6.21	...	240000.0	10000.0	1091732.5	8
4	25.0	2021	11	4	14	30	1129657	7.21	7005.00	7.03	...	230000.0	20000.0	1019806.6	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7448	48.0	2021	10	4	15	30	246373	37.63	35.80	35.91	...	1510000.0	270000.0	342675.7	-5
7449	49.0	2021	10	1	15	30	408509	37.80	36.75	37.75	...	1530000.0	330000.0	347417.8	-4
7450	50.0	2021	9	30	15	30	225525	37.22	35.69	36.65	...	1530000.0	270000.0	344393.2	1
7451	51.0	2021	9	29	15	30	491022	37.05	36.03	36.11	...	1530000.0	250000.0	345491.6	1
7452	52.0	2021	9	28	15	30	327982	36505.00	35.11	35.67	...	1530000.0	210000.0	343620.5	2

7453 rows × 76 columns

Above displayed is the data frame with 7453 rows and 76 columns for the given data set.

We have also tried using classification tree Model



But here the disadvantage of a classification tree when dealing with a large number of predictor variables is that it can become computationally expensive and may suffer from overfitting, leading to poor performance on new, unseen data. This is because the tree can become too complex and overly specific to the training data, making it difficult to generalize to new data.

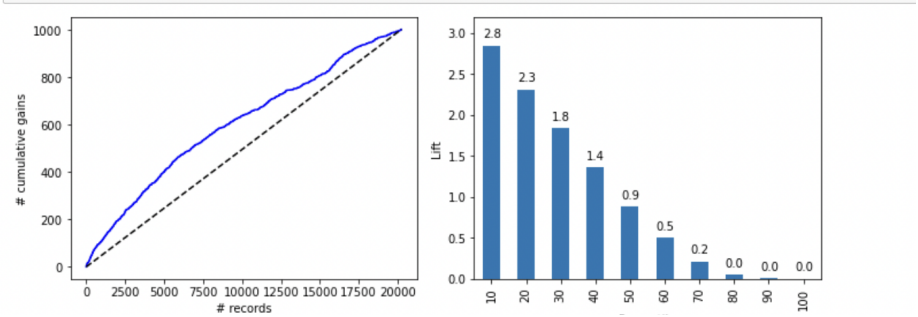
4. Interpret Results

b) Logistic Regression - Profit/Loss Prediction

The confusion matrices of the model show that it has a high level of accuracy 94% for the validation partition and 95% for training partition, indicating a good fit. Moreover, there is no indication of overfitting since the model performed better on the validation set.

```
Training Partition
Confusion Matrix (Accuracy 0.9525)

       Prediction
Actual     0     1
     0 28893    39
     1  1404    24

Validation Partition
Confusion Matrix (Accuracy 0.9497)

       Prediction
Actual     0     1
     0 19212    27
     1   991    11
```

Based on the lift chart, the model's top 10% selection has a 2.8 times higher likelihood of being a profit compared to randomly selecting the same proportion. This suggests that the model's selection is significantly more effective than a random approach.



```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                 TARGET   No. Observations:                30360
Model:                            GLM   Df Residuals:                    30288
Model Family:                Binomial   Df Model:                           71
Link Function:                  Logit   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -5523.7
Date:                Sun, 14 May 2023   Deviance:                       11047.
Time:                        21:04:23   Pearson chi2:                 5.00e+04
No. Iterations:                   100   Pseudo R-squ. (CS):            0.01538
Covariance Type:            nonrobust
```

Above displayed are the general statistics of the model.

We have used the Backward Elimination Algorithm to get the best predictor variables.

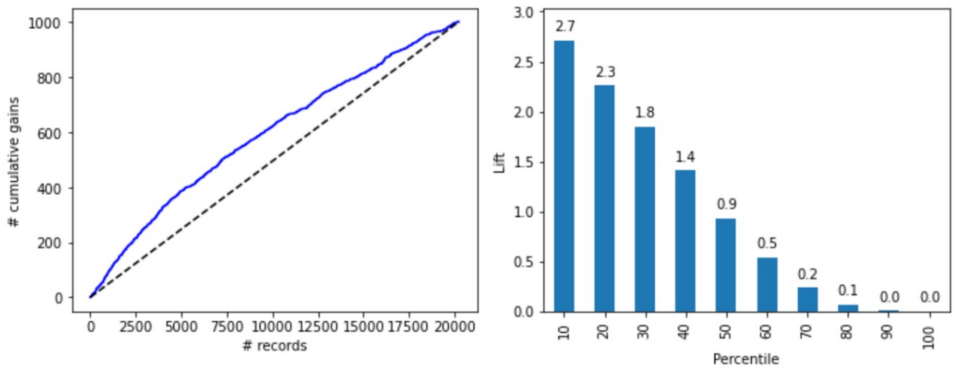
```
Best Variables from Backward Elimination Algorithm
['age', 'hour', 'volume', 'AVERAGE_SMA_3_CLOSE', 'EMA_3_HIGHLOW', 'MACD_7_HIGHLOW', 'EMA_20_CLOSE', 'MACD_20_CLOSE',
'AVERAGE_SMA_20_OPENHIGH', 'EMA_20_OPENHIGH', 'AVERAGE_SMA_20_HIGHLOW', 'EMA_20_HIGHLOW', 'MACD_20_HIGHLOW', 'MACD_20
_VOLUME', 'AVERAGE_SMA_50_CLOSE', 'MACD_50_CLOSE', 'EMA_50_OPENHIGH', 'MACD_50_OPENHIGH', 'EMA_50_HIGHLOW', 'MACD_50_
HIGHLOW', 'MACD_50_VOLUME', 'DINAMIC3', 'company_ordinal']
```

```
Training Partition
Confusion Matrix (Accuracy 0.9521)

       Prediction
Actual     0     1
     0 28905    27
     1  1426     2

Validation Partition
Confusion Matrix (Accuracy 0.9495)

       Prediction
Actual     0     1
     0 19217    22
     1  1001     1
```

Above displayed is the Confusion matrix of the Backward Elimination Algorithm

Although after implementing the Backward Elimination Algorithm the accuracy measures are the same.

So, we considered Logistic Regression as a pretty good model for the data set we have taken.



Here is the Lift chart of the Backward Elimination Model.

Conclusion:

Based on the report, we have acquired and preprocessed a dataset with 98192 rows and 77 columns. After dropping irrelevant and null values, we were left with 50601 rows and 76 columns. We then converted non-numeric data into numeric data using dummy variable encoding and performed ordinal encoding on the 'Company' column. The data was then split into training and validation sets using the train_test_split function with a test size of 40%.

We selected logistic regression and classification trees as our analysis methods for predicting the binary variable "Target", which determines profit or loss depending on various factors. Logistic regression models are effective in modeling the probability of a binary outcome, while classification models are able to predict categorical outcomes.

The logistic regression model was fit on the training partition and the accuracy score of the model was found to be 95.25%, and for the Validation Partition it is found to be 94.97%. This indicates that the model is able to classify the profit or loss outcome correctly for approximately 95% of the observations.

In conclusion, we have successfully preprocessed and analyzed the dataset using logistic regression and classification tree models. The accuracy of the logistic regression model was found to be good.