Sahil Mehta, Daniel Garcia Mijares, Claudia Cañete Yaque, Majda Brahimi

INDENG 242 Project Report

❖ Motivation

The financial market is a dynamic and complex ecosystem influenced by numerous factors, ranging from economic indicators to geopolitical events. Investors constantly seek insights to navigate this landscape and make informed decisions. Our project addresses the need for predictive tools that can assist investors in understanding and anticipating fluctuations in stock prices. Traditional methods of stock price prediction often rely on historical data and technical indicators. While valuable, these approaches may overlook external factors that can swiftly impact market dynamics. One such factor is the influence of public sentiment, particularly in the age of social media dominance.

Elon Musk, the CEO of Tesla, exemplifies how individual actions and statements can reverberate throughout financial markets. Musk's tweets and public appearances could be correlated with significant fluctuations in Tesla's stock price. Understanding and quantifying this relationship can provide valuable insights for investors and analysts alike.

Given Musk's prominence and the observable impact of his behavior on Tesla's stock performance, we have chosen to concentrate our analysis on this specific case. By examining the correlation between Musk's sentiments, as reflected in social media, and Tesla's stock price movements, we aim to uncover actionable insights that can enhance predictive modeling in the financial domain.

Our primary objective is to develop a model that predicts future stock prices that incorporates Musk's social media sentiment as a key variable, alongside traditional financial indicators. By leveraging a diverse array of data sources, we strive to provide investors with a robust tool for anticipating changes in Tesla's stock price with some accuracy.

❖ Data

From a rich resource on Kaggle titled "Trading Tesla with Machine Learning and Sentiment Analysis", we obtained the raw tweets to work with. These were tweets extracted from several financial news X accounts such as WSJ, CNBC or Reuters that contained the word 'Tesla' (in total 223879 tweets), as well as Tweets from the official ElonMusk account. To clean the dataset, we deleted duplicate tweets, Urls, @mentions, and isolated the text using the Beautiful Soup library.

To obtain a sentiment score for each tweet, we employed the VADER (Valence Aware Dictionary and sEntiment Reasoner) module. VADER is a lexicon and rule-based sentiment analysis tool specially crafted for social media texts. It quantifies the polarity (positive, negative, neutral) and intensity of sentiments expressed in textual data, thereby enabling us to gauge the overall sentiment trends regarding Tesla from social media discourse.

Complementing the sentiment data, we obtained historical stock price data directly from Yahoo Finance for Tesla as well as its direct competitors (Toyota, Nissan, GM, Ford) and the SP500 variance. This dataset served as the cornerstone of our predictive modeling efforts, offering insights into past price movements and trends. Given the high frequency and granularity of stock price data, we opted to aggregate the data on a weekly basis to align with our prediction horizon, ensuring consistency and relevance in our analysis.

In addition to stock prices and sentiment scores, we also used the 'Tesla Motors' Wikipedia page views. This would provide us with a view of Tesla's popularity each week.

For the cleansing of these dataset, we ensured that the data collected for all features belonged to the same dates as those of the data available for Tesla's stock prices. We checked for NaN values, where we input an average of the last and first values of the missing interval.

In our task of predictive accuracy, we used a diverse ensemble of machine learning models, including Linear Regression, Decision Tree Regressor, Random Forest, and Gradient Boosting Trees. By training and evaluating these models on our augmented dataset, we aimed to discern the most effective approach for forecasting Tesla's stock price dynamics. Furthermore, we explored the potential synergies of ensemble modeling techniques, combining the strengths of individual models to enhance predictive performance.

In the Linear Regression method we developed a feature engineering analysis, where we started off by using a dataset with all the features and data that we obtained. In an iterative process, we trained the model and checked for those features with high (>5) VIF values as well as high (>0.5) p-values. We dropped one feature at a time and repeated the process in each iteration. When all the VIF and p-values were in the preferred interval, we were left with 'Toyota Stock Price', 'Previous Week's Sentiment Score' and 'Previous Week Tesla's Close Price'. These are the features that do not show correlation and have relatively larger relevance when predicting Tesla's Close Price.

- **Analytics models:**

Predicting Next Week's Stock Price (regression models)

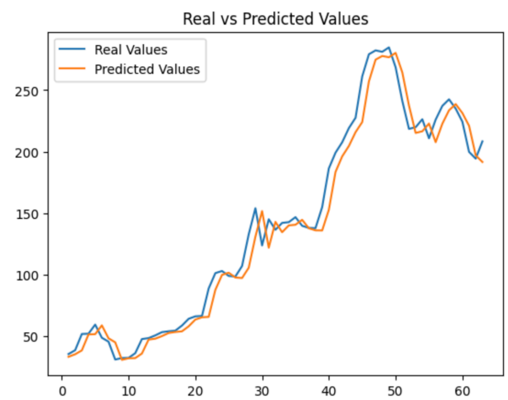
Five models were used to predict next week's Tesla stock price: linear regression, CART regression, Random Forest regression, Gradient Boosting regression, and an ensemble model. The CART and Random Forest models implemented 5-fold cross-validation. The CART model cross-validates on the cost complexity parameter ("ccp_alpha"), scoring based on R^2. The Random Forest model cross-validates on "max_features" (the number of features to consider when looking for the best split), with scoring based on R^2. The ensemble model was created by building a linear regression model whose inputs are the output values of the CART, Random Forest, and Gradient Boosting models.

For each of the models, the OSR2 (out of sample R squared) and mean absolute error percentage are calculated on the test set. The table below compares the results:

	Linear Regressor	Decision Tree Regressor	Random Forest	Gradient Boosted Trees	Ensemble Model
OSR2	0.994	0.984	0.988	0.984	0.984
Average Percent Test Error	0.0503	0.0661	0.0524	0.0577	0.0583

By creating plots of the real and predicted stock prices, we determined that our prediction model is very closely correlated with Tesla's real stock price trends (this can also be seen in the OSR2 values above). However, upon assessment of feature importance values for the Gradient Boosting regression model (see table below), we see that more than 99% of feature weight is on the previous week's stock price, indicating that these models may not have greater predictive power than predicting simply next week's price using this week's price. The plot above compares the predicted and real stock prices using the linear regression model. The prediction plot is nearly identical to the real plot, just shifted right one week, indicating the tendency to predict next week's price solely with this week's price.

	Toyota Stock Price	Previous Sentiment	Previous Week Tesla Stock Price
Feature Importance	0.005	0.001	0.994



In order to make accurate predictions that are not affected by the previous week's price, we subsequently attempted to **classify whether the price will increase or decrease each week, regardless of the actual value**, as seen in the next section.

Predicting Whether the Stock Price Goes Up or Down (classification models)

Four models were used to predict whether the stock price goes up or down the following week: baseline, CART classification, Random Forest classification, and Gradient Boosting classification. The baseline model will predict the stock price will increase for any observation. Similar to the regression implementation, the CART and Random Forest models implement 5-fold cross-validation (with the same hyperparameters as used for regression). For each of the models, the accuracy, true positive rate, false positive rate, and precision are calculated on the test set. The table below compares the results:

	Baseline	Decision Tree Classifier	Random Forest Classifier	Gradient Boosted Classifier
Accuracy	0.605	0.548	0.629	0.694
TPR	1.0000	0.6267	0.7600	0.7733
FPR	1.000	0.571	0.571	0.429
Precision	0.605	0.627	0.671	0.734

Confidence in Results

To assess confidence in our results, we calculated the bootstrapped sample variance for the result metrics on the test set. The following tables show the resulting variance values for the regression and classification models, which all are small compared to the metric values, allowing us to be fairly confident in our results.

	Linear Regressor	Decision Tree Regressor	Random Forest	Gradient Boosted Trees	Ensemble Model
Variance of Average Percent Test Error	0.000098	0.000127	0.000022	0.000022	0.000021

	Decision Tree Classifier	Random Forest Classifier	Gradient Boosted Classifier
Variance Accuracy	0.002	0.002	0.002
Variance TPR	0.0035	0.0025	0.0020
Variance FPR	0.005	0.005	0.005
Variance Precision	0.004	0.003	0.003

Extension of Analysis

For future analysis of predicting next week's stock price (regression) or whether the Tesla stock price goes up or down (classification), we can extend the model implementations in the following ways:

1. Experiment with multiple other possible hyperparameters for cross-validation, including "min_samples_leaf" (minimum number of samples required to be a leaf node), "n_estimators" (number of trees in the forest), and "random_state" (randomness factor).
2. Experiment with multiple scoring factors for cross validation. For regression, we can attempt to minimize the average percent error. For classification, we can attempt to minimize false positive rate, or maximize the accuracy or true positive rate.
3. Additional models we can implement for classification:
    a. A logistic regression model for predicting the probability that the price goes up or down
    b. An ensemble model that assesses the outputs of multiple individual classification models and aggregates them into a single prediction

❖ Impact

The potential impact of our work would be providing investors with a more comprehensive understanding of the factors influencing stock prices, particularly in the context of influential figures like Musk and their online presence.

Expanding the scope of our analysis could involve several approaches to enhance its impact further. Especially, rather than having one feature grouping the sentiment analysis from various Twitter accounts as presented in the Data section, it would be interesting to implement the sentiment analysis on each one Twitter account individually and assess if any account's sentiment correlates with stock price more strongly than others. For example, we would assume that the tweets that contain "Tesla" from randomly searched accounts would have more inaccurate information than those of well-known Twitter accounts like WSJ, therefore affecting the correlation between the tweets' sentiment and Tesla's stock price. The next step would therefore be to validate or disprove assumptions of this nature.

The data and code associated with this report can be found in the GitHub repository:
https://github.com/majda-br/Stock-trend-prediction/tree/master

Project Models - CART, Random Forest, Boosting

In [96]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import time

from sklearn.model_selection import train_test_split, GridSearchCV, KFold
from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor, RandomForestClassifier
from sklearn.linear_model import LinearRegression
from scipy import stats
from scipy.stats import pearsonr
from sklearn.metrics import mean_squared_error, mean_absolute_error, confusion_matrix, roc_curve, auc
from sklearn.utils import resample
from sklearn import datasets, linear_model
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [97]:
```
# Helper functions

def VIF(df, columns):
    values=sm.add_constant(df[columns]).values
    # the dataframe passed to VIF␣ ↪ must include the intercept term. We add it the same way we did before.
    num_columns=len(columns)+1#we added intercept
    vif=[variance_inflation_factor(values, i) for i in range(num_columns)]
    return pd.Series(vif[1:], index=columns)
def OSR2(model, y_train, x_test,y_test):
    y_pred=model.predict(x_test)
    SSE=np.sum((y_test-y_pred)**2)
    SST=np.sum((y_test-np.mean(y_train))**2)
    return 1-(SSE/SST)
def avg_error(y_pred, y_test):
    return np.average(abs((y_pred - y_test) / y_test))
```

Load Data - Train & Test split with time benchmark

In [141…
```
training_data = pd.read_csv('../features_train.csv')
training_data.drop(columns=['Unnamed: 0'], inplace=True)
training_data = training_data.drop(0)
training_data['Previous Sentiment'] = training_data['Sentiment'].shift(1)
training_data = training_data[1:]

testing_data = pd.read_csv('../features_test.csv')
testing_data.drop(columns=['Unnamed: 0'], inplace=True)
testing_data['Previous Sentiment'] = testing_data['Sentiment'].shift(1)

testing_data = testing_data[1:]
X_train = training_data.drop(columns=['Tesla Stock Close Price'])
y_train = training_data['Tesla Stock Close Price']
X_test = testing_data.drop(columns=['Tesla Stock Close Price'])
y_test = testing_data['Tesla Stock Close Price']
#use only the important features
features =X_train.columns
X_train.head()
```

Out[141]:

	Tesla Stock Open Price	Tesla Stock Volume	Tesla Stock Adj Close Price	Tesla Stock High	Tesla Stock Low	S&P 500 Variance	Ford Stock Price	GM Stock Price	Toyota Stock Price	Nissan Stock Price	Tesla Wikipedia Page Views	Sentiment	Previous Sentiment	Previous Week Tesla Stock Close Price
2	1.401200	13199100.0	1.388933	1.424133	1.354267	435.732436	13.0400	34.189999	72.763998	15.460	1102.2	0.140323	0.089603	1.384800
3	1.256267	12649800.0	1.232667	1.264800	1.215600	380.825787	12.5600	34.189999	70.756001	15.062	898.0	0.040080	0.140323	1.388933
4	1.255867	7230300.0	1.261200	1.282000	1.233067	316.106150	12.0020	34.189999	70.690001	15.200	801.0	0.188023	0.040080	1.232667
5	1.303333	8167800.0	1.312267	1.333600	1.282933	591.345328	11.3280	34.189999	69.030000	14.992	766.2	0.168206	0.188023	1.261200
6	1.357833	6225000.0	1.385000	1.403833	1.351167	615.993610	11.7975	34.189999	69.000000	15.745	808.5	0.119400	0.168206	1.312267

Linear Regression & Feature Engineering

In [142…
```
#Now we want to see how the model behaves. We train the linear regression.
#We will mostly focus on the p-values, the VIF values, and R2.
X=X_train
Y=y_train
X2 = sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, features))

pd.set_option('display.max_colwidth', None)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       1.000
Model:                                 OLS   Adj. R-squared:                  1.000
Method:                      Least Squares   F-statistic:                 1.079e+20
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:32   Log-Likelihood:                 7723.3
No. Observations:                      431   AIC:                        -1.542e+04
Df Residuals:                          416   BIC:                        -1.536e+04
Df Model:                               14
Covariance Type:                 nonrobust
=========================================================================================================
                                            coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
const                                  9.925e-14   2.55e-09   3.89e-05      1.000   -5.01e-09    5.01e-09
Tesla Stock Open Price                 4.796e-14   3.51e-09   1.36e-05      1.000   -6.91e-09    6.91e-09
Tesla Stock Volume                     3.918e-17   6.23e-18      6.285      0.000    2.69e-17    5.14e-17
Tesla Stock Adj Close Price               1.0000   3.43e-09   2.92e+08      0.000       1.000       1.000
Tesla Stock High                       4.619e-14   3.68e-09   1.25e-05      1.000   -7.24e-09    7.24e-09
Tesla Stock Low                        3.073e-13   3.64e-09   8.44e-05      1.000   -7.15e-09    7.15e-09
S&P 500 Variance                      -9.682e-17   1.81e-13     -0.001      1.000   -3.56e-13    3.56e-13
Ford Stock Price                      -5.607e-15   1.15e-10  -4.87e-05      1.000   -2.26e-10    2.26e-10
GM Stock Price                        -1.055e-15   6.12e-11  -1.72e-05      1.000    -1.2e-10     1.2e-10
Toyota Stock Price                     1.561e-17   2.04e-11   7.67e-07      1.000      -4e-11       4e-11
Nissan Stock Price                     1.735e-15   1.15e-10   1.51e-05      1.000   -2.26e-10    2.26e-10
Tesla Wikipedia Page Views             5.909e-18    1.9e-13   3.11e-05      1.000   -3.74e-13    3.74e-13
Sentiment                              2.665e-15   2.47e-09   1.08e-06      1.000   -4.85e-09    4.85e-09
Previous Sentiment                     2.665e-15   2.47e-09   1.08e-06      1.000   -4.85e-09    4.85e-09
Previous Week Tesla Stock Close Price -6.106e-16   2.97e-10  -2.05e-06      1.000   -5.84e-10    5.84e-10
==============================================================================
Omnibus:                       89.707   Durbin-Watson:                   0.194
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              161.440
Skew:                          -1.183   Prob(JB):                     8.78e-36
Kurtosis:                       4.841   Cond. No.                     3.32e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.32e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
Tesla Stock Open Price                   18624.002920
Tesla Stock Volume                           3.772195
Tesla Stock Adj Close Price              17738.592869
Tesla Stock High                         21097.838139
Tesla Stock Low                          19345.653437
S&P 500 Variance                             1.452583
Ford Stock Price                             2.228986
GM Stock Price                               2.887534
Toyota Stock Price                           4.921429
Nissan Stock Price                           1.523839
Tesla Wikipedia Page Views                   2.603870
Sentiment                                    1.444684
Previous Sentiment                           1.445891
Previous Week Tesla Stock Close Price      132.639340
dtype: float64
```

In [143…
```
#We see that the p-values are very high for some features, we should eliminate them.
#We eliminate Tesla's Financial Features which are very correlated with the Close Price we are trying to predict, since it has a very high p-value as well as a high VIF value.
X_train = X_train.drop(columns=['Tesla Stock Open Price','Tesla Stock Adj Close Price','Tesla Stock High', 'Tesla Stock Low'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

```
                                OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                     3759.
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:38   Log-Likelihood:                -515.49
No. Observations:                      431   AIC:                             1053.
Df Residuals:                          420   BIC:                             1098.
Df Model:                               10
Covariance Type:                 nonrobust
==============================================================================================
                                       coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                               -0.7734      0.507     -1.527      0.128      -1.769       0.222
Tesla Stock Volume                9.445e-10   9.05e-10      1.044      0.297   -8.34e-10    2.72e-09
S&P 500 Variance                  6.608e-06   3.28e-05      0.201      0.841   -5.79e-05    7.11e-05
Ford Stock Price                    -0.0227      0.023     -0.993      0.321      -0.068       0.022
GM Stock Price                       0.0132      0.012      1.088      0.277      -0.011       0.037
Toyota Stock Price                   0.0069      0.004      1.745      0.082      -0.001       0.015
Nissan Stock Price                   0.0008      0.023      0.033      0.974      -0.044       0.045
Tesla Wikipedia Page Views       -1.319e-05   3.78e-05     -0.349      0.727   -8.74e-05    6.11e-05
Sentiment                            0.4894      0.491      0.998      0.319      -0.475       1.454
Previous Sentiment                   0.6506      0.491      1.325      0.186      -0.314       1.615
Previous Week Tesla Stock Close Price     0.9759      0.011     88.529      0.000       0.954       0.998
==============================================================================
Omnibus:                       32.702   Durbin-Watson:                   1.624
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              134.970
Skew:                           0.048   Prob(JB):                     4.92e-30
Kurtosis:                       5.740   Cond. No.                     1.49e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.49e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
Tesla Stock Volume                       2.000454
S&P 500 Variance                         1.201987
Ford Stock Price                         2.213930
GM Stock Price                           2.846517
Toyota Stock Price                       4.606307
Nissan Stock Price                       1.492372
Tesla Wikipedia Page Views               2.587860
Sentiment                                1.437270
Previous Sentiment                       1.439137
Previous Week Tesla Stock Close Price    4.589308
dtype: float64
```

In [144…

```python
#Now we eliminate Nissan stock prices, since it has a very high p-value
X_train = X_train.drop(columns=['Nissan Stock Price'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

```
                                OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                     4187.
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:42   Log-Likelihood:                -515.49
No. Observations:                      431   AIC:                             1051.
Df Residuals:                          421   BIC:                             1092.
Df Model:                                9
Covariance Type:                 nonrobust
==============================================================================================
                                       coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                               -0.7611      0.346     -2.200      0.028      -1.441      -0.081
Tesla Stock Volume                9.396e-10   8.92e-10      1.054      0.293   -8.13e-10    2.69e-09
S&P 500 Variance                  6.555e-06   3.28e-05      0.200      0.841   -5.78e-05    7.09e-05
Ford Stock Price                    -0.0224      0.021     -1.090      0.277      -0.063       0.018
GM Stock Price                       0.0131      0.012      1.097      0.273      -0.010       0.037
Toyota Stock Price                   0.0069      0.004      1.749      0.081      -0.001       0.015
Tesla Wikipedia Page Views       -1.304e-05   3.75e-05     -0.348      0.728   -8.67e-05    6.06e-05
Sentiment                            0.4868      0.484      1.006      0.315      -0.464       1.437
Previous Sentiment                   0.6480      0.484      1.339      0.181      -0.303       1.599
Previous Week Tesla Stock Close Price     0.9759      0.011     88.645      0.000       0.954       0.998
==============================================================================
Omnibus:                       32.709   Durbin-Watson:                   1.624
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              135.043
Skew:                           0.047   Prob(JB):                     4.74e-30
Kurtosis:                       5.741   Cond. No.                     1.48e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.48e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
Tesla Stock Volume                       1.947392
S&P 500 Variance                         1.199182
Ford Stock Price                         1.788081
GM Stock Price                           2.778646
Toyota Stock Price                       4.603469
Tesla Wikipedia Page Views               2.551097
Sentiment                                1.400257
Previous Sentiment                       1.402002
Previous Week Tesla Stock Close Price    4.588212
dtype: float64
```

In [145…

```python
#Now we eliminate  SP500 Variance , since it has a very high p-value
X_train = X_train.drop(columns=['S&P 500 Variance'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

```
                                OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                     4721.
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:45   Log-Likelihood:                -515.51
No. Observations:                      431   AIC:                             1049.
Df Residuals:                          422   BIC:                             1086.
Df Model:                                8
Covariance Type:                 nonrobust
==============================================================================================
                                       coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                               -0.7486      0.340     -2.202      0.028      -1.417      -0.080
Tesla Stock Volume                9.458e-10    8.9e-10      1.062      0.289   -8.04e-10     2.7e-09
Ford Stock Price                    -0.0235      0.020     -1.188      0.235      -0.062       0.015
GM Stock Price                       0.0131      0.012      1.096      0.274      -0.010       0.037
Toyota Stock Price                   0.0069      0.004      1.767      0.078      -0.001       0.015
Tesla Wikipedia Page Views       -1.316e-05   3.74e-05     -0.352      0.725   -8.67e-05    6.04e-05
Sentiment                            0.4902      0.483      1.015      0.311      -0.459       1.439
Previous Sentiment                   0.6521      0.483      1.350      0.178      -0.297       1.601
Previous Week Tesla Stock Close Price     0.9760      0.011     88.958      0.000       0.954       0.998
==============================================================================
Omnibus:                       32.749   Durbin-Watson:                   1.625
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              134.846
Skew:                           0.057   Prob(JB):                     5.23e-30
Kurtosis:                       5.738   Cond. No.                     1.48e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.48e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
Tesla Stock Volume                       1.945036
Ford Stock Price                         1.658710
GM Stock Price                           2.778089
Toyota Stock Price                       4.585023
Tesla Wikipedia Page Views               2.550467
Sentiment                                1.398512
Previous Sentiment                       1.399460
Previous Week Tesla Stock Close Price    4.567839
dtype: float64
```

In [146…

```python
#Now we eliminate Tesla Wikipedia Page Views, since it has a very high p-value
X_train = X_train.drop(columns=['Tesla Wikipedia Page Views'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

```
                                OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                     5407.
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:47   Log-Likelihood:                -515.58
No. Observations:                      431   AIC:                             1047.
Df Residuals:                          423   BIC:                             1080.
Df Model:                                7
Covariance Type:                 nonrobust
=====================================================================================================
                                           coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
const                                   -0.7284      0.335     -2.176      0.030      -1.386      -0.071
Tesla Stock Volume                    8.331e-10    8.3e-10      1.004      0.316   -7.98e-10    2.46e-09
Ford Stock Price                        -0.0259      0.019     -1.397      0.163      -0.062       0.011
GM Stock Price                           0.0142      0.011      1.238      0.216      -0.008       0.037
Toyota Stock Price                       0.0065      0.004      1.743      0.082      -0.001       0.014
Sentiment                                0.4845      0.482      1.005      0.315      -0.463       1.432
Previous Sentiment                       0.6391      0.481      1.329      0.185      -0.306       1.585
Previous Week Tesla Stock Close Price    0.9750      0.011     92.598      0.000       0.954       0.996
==============================================================================
Omnibus:                       32.749   Durbin-Watson:                   1.624
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              134.287
Skew:                           0.065   Prob(JB):                     6.92e-30
Kurtosis:                       5.731   Cond. No.                     1.48e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.48e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
Tesla Stock Volume                       1.693017
Ford Stock Price                         1.461241
GM Stock Price                           2.575548
Toyota Stock Price                       4.178283
Sentiment                                1.396930
Previous Sentiment                       1.391243
Previous Week Tesla Stock Close Price    4.215253
dtype: float64
```

In [147…

```python
#Now we eliminate Volume, since it has a very high p-value
X_train = X_train.drop(columns=['Tesla Stock Volume'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

```
                                OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                     6307.
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:49   Log-Likelihood:                -516.09
No. Observations:                      431   AIC:                             1046.
Df Residuals:                          424   BIC:                             1075.
Df Model:                                6
Covariance Type:                 nonrobust
=====================================================================================================
                                           coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
const                                   -0.8204      0.322     -2.548      0.011      -1.453      -0.188
Ford Stock Price                        -0.0278      0.018     -1.512      0.131      -0.064       0.008
GM Stock Price                           0.0158      0.011      1.395      0.164      -0.006       0.038
Toyota Stock Price                       0.0076      0.004      2.142      0.033       0.001       0.015
Sentiment                                0.4640      0.482      0.964      0.336      -0.483       1.411
Previous Sentiment                       0.6860      0.479      1.433      0.153      -0.255       1.627
Previous Week Tesla Stock Close Price    0.9753      0.011     92.658      0.000       0.955       0.996
==============================================================================
Omnibus:                       36.592   Durbin-Watson:                   1.625
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              148.381
Skew:                           0.196   Prob(JB):                     6.02e-33
Kurtosis:                       5.848   Cond. No.                     1.71e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.71e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
Ford Stock Price                         1.444828
GM Stock Price                           2.523609
Toyota Stock Price                       3.803325
Sentiment                                1.394437
Previous Sentiment                       1.378139
Previous Week Tesla Stock Close Price    4.212122
dtype: float64
```

In [148…

```python
#Now we eliminate Sentiment scores, since it has a very high p-value
X_train = X_train.drop(columns=['Sentiment'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

```
                                OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                     7570.
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:54   Log-Likelihood:                -516.56
No. Observations:                      431   AIC:                             1045.
Df Residuals:                          425   BIC:                             1070.
Df Model:                                5
Covariance Type:                 nonrobust
=====================================================================================================
                                           coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
const                                   -0.8024      0.321     -2.497      0.013      -1.434      -0.171
Ford Stock Price                        -0.0263      0.018     -1.436      0.152      -0.062       0.010
GM Stock Price                           0.0154      0.011      1.357      0.175      -0.007       0.038
Toyota Stock Price                       0.0080      0.004      2.275      0.023       0.001       0.015
Previous Sentiment                       0.8597      0.443      1.938      0.053      -0.012       1.731
Previous Week Tesla Stock Close Price    0.9735      0.010     93.995      0.000       0.953       0.994
==============================================================================
Omnibus:                       36.760   Durbin-Watson:                   1.627
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              148.561
Skew:                           0.202   Prob(JB):                     5.50e-33
Kurtosis:                       5.848   Cond. No.                     1.36e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.36e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
Ford Stock Price                         1.434545
GM Stock Price                           2.519604
Toyota Stock Price                       3.748299
Previous Sentiment                       1.182733
Previous Week Tesla Stock Close Price    4.078710
dtype: float64
```

In [149…

```python
#Now we eliminate GM stock prices, since it has a very high p-value
X_train = X_train.drop(columns=['GM Stock Price'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

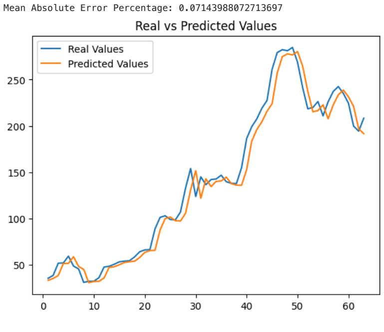
```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                     9443.
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:56   Log-Likelihood:                -517.49
No. Observations:                      431   AIC:                             1045.
Df Residuals:                          426   BIC:                             1065.
Df Model:                                4
Covariance Type:                 nonrobust
======================================================================================================
                                           coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------------
const                                   -0.6044      0.287     -2.109      0.036      -1.168      -0.041
Ford Stock Price                        -0.0184      0.017     -1.056      0.291      -0.053       0.016
Toyota Stock Price                       0.0096      0.003      2.863      0.004       0.003       0.016
Previous Sentiment                       0.7947      0.441      1.801      0.072      -0.073       1.662
Previous Week Tesla Stock Close Price    0.9778      0.010     99.085      0.000       0.958       0.997
==============================================================================
Omnibus:                       36.931   Durbin-Watson:                   1.631
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              151.574
Skew:                           0.195   Prob(JB):                     1.22e-33
Kurtosis:                       5.879   Cond. No.                     1.29e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.29e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
Ford Stock Price                         1.287819
Toyota Stock Price                       3.358456
Previous Sentiment                       1.168945
Previous Week Tesla Stock Close Price    3.695664
dtype: float64
```

In [150…
```python
#Now we eliminate Ford stock prices, since it has a very high p-value
X_train = X_train.drop(columns=['Ford Stock Price'])
X=X_train
X2=sm.add_constant(X)
lrm=sm.OLS(Y, X2).fit()
print(lrm.summary())
print(VIF(training_data, X_train.columns))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Tesla Stock Close Price   R-squared:                       0.989
Model:                                 OLS   Adj. R-squared:                  0.989
Method:                      Least Squares   F-statistic:                 1.259e+04
Date:                     Thu, 02 May 2024   Prob (F-statistic):               0.00
Time:                             11:46:57   Log-Likelihood:                -518.06
No. Observations:                      431   AIC:                             1044.
Df Residuals:                          427   BIC:                             1060.
Df Model:                                3
Covariance Type:                 nonrobust
======================================================================================================
                                           coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------------
const                                   -0.7304      0.261     -2.802      0.005      -1.243      -0.218
Toyota Stock Price                       0.0082      0.003      2.662      0.008       0.002       0.014
Previous Sentiment                       0.7427      0.439      1.693      0.091      -0.119       1.605
Previous Week Tesla Stock Close Price    0.9820      0.009    108.942      0.000       0.964       1.000
==============================================================================
Omnibus:                       38.299   Durbin-Watson:                   1.630
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              158.037
Skew:                           0.220   Prob(JB):                     4.82e-35
Kurtosis:                       5.934   Cond. No.                     1.27e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.27e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
Toyota Stock Price                       2.838186
Previous Sentiment                       1.154404
Previous Week Tesla Stock Close Price    3.082939
dtype: float64
```

In [151…
```python
from sklearn.metrics import r2_score

X_test = X_test.drop(columns=['Tesla Stock Open Price','Tesla Stock Volume','Tesla Stock Adj Close Price','Tesla Stock High','Tesla Stock Low','S&P 500 Variance', 'Tesla Wikipedia Page Views','Nissan Stock Price','Sentiment','Ford Stock Price','GM Stock 
X_test=sm.add_constant(X_test)
print(X_test.head())
y_pred = lrm.predict(X_test)

# Assuming y_test and y_pred are the actual and predicted values, respectively
r2 = r2_score(y_test, y_pred)
print("R2 Score:", r2)

osr2=OSR2(lrm, y_train, X_test, y_test)
print("OSR2 Score:", osr2)

mean_absolute_error = np.mean(np.abs(y_test - y_pred))
mean_absolute_error_percentage = mean_absolute_error / np.mean(y_test)
print("Mean Absolute Error:", mean_absolute_error)
print("Mean Absolute Error Percentage:", mean_absolute_error_percentage)

plt.title('Real vs Predicted Values')

plt.plot(y_test,  label= 'Real Values')
plt.plot(y_pred, label = 'Predicted Values')
plt.legend()
plt.show()
```

```
   const  Toyota Stock Price  Previous Sentiment  \
1    1.0          141.020000            0.170538
2    1.0          141.806000            0.092409
3    1.0          141.556000            0.134267
4    1.0          141.413998            0.231126
5    1.0          138.842499            0.205790

   Previous Week Tesla Stock Close Price
1                              33.525466
2                              35.679167
3                              38.820533
4                              51.983867
5                              52.226400
R2 Score: 0.9725787708603995
OSR2 Score: 0.9922626917883026
Mean Absolute Error: 10.098887378382408
Mean Absolute Error Percentage: 0.07143988072713697
```



Load Data - Train & Test split with time benchmark

Random split is needed for accuracy of tree-based models since the range of possible stock price values changes over time

In [152…
```python
# Resplit the full dataset using a random procedure
data = pd.read_csv("../data.csv")
train_data, test_data = train_test_split(data)
test_data = test_data.drop(columns=['since', 'until','Unnamed: 0'])
train_data = train_data.drop(columns=['since', 'until','Unnamed: 0'])

train_data['Previous Sentiment'] = train_data['Sentiment'].shift(1)
train_data = train_data[1:]
test_data['Previous Sentiment'] = test_data['Sentiment'].shift(1)
test_data = test_data[1:]
```

In [153…
```python
y_train = train_data['Tesla Stock Price']
X_train = train_data.drop(columns=['Tesla Stock Price','S&P 500 Variance', 'Tesla Wikipedia Page Views','Nissan Stock Price','Sentiment','Ford Stock Price','GM Stock Price'])

y_test = test_data['Tesla Stock Price']
X_test = test_data.drop(columns=['Tesla Stock Price','S&P 500 Variance', 'Tesla Wikipedia Page Views','Nissan Stock Price','Sentiment','Ford Stock Price','GM Stock Price'])
```

CART

In [154…
```
grid_values = {'ccp_alpha': np.linspace(0, 0.001, 51)}

dtr = DecisionTreeRegressor(min_samples_leaf=5, min_samples_split=20, random_state=88)
cv = KFold(n_splits=5,random_state=1,shuffle=True)
dtr_cv = GridSearchCV(dtr, param_grid=grid_values, scoring='r2', cv=cv, verbose=0)
dtr_cv.fit(X_train, y_train)
test_pred_cart, train_pred_cart = dtr_cv.predict(X_test), dtr_cv.predict(X_train)
```

RANDOM FORESTS

In [155…
```
grid_values = {'max_features': np.linspace(1,5,5, dtype='int32'),
              'min_samples_leaf': [5],
              'n_estimators': [500],
              'random_state': [88]}

rf2 = RandomForestRegressor()
cv = KFold(n_splits=5,random_state=333,shuffle=True)
rf_cv = GridSearchCV(rf2, param_grid=grid_values, scoring='r2', cv=cv,verbose=2)
rf_cv.fit(X_train, y_train)
test_pred_rf, train_pred_rf = rf_cv.predict(X_test), rf_cv.predict(X_train)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV] END max_features=1, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=1, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=1, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=1, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=1, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=2, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=2, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=2, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=2, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=2, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.6s
[CV] END max_features=3, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=3, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=3, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=3, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=3, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=4, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=4, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=4, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=4, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=4, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=5, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=5, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=5, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=5, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
[CV] END max_features=5, min_samples_leaf=5, n_estimators=500, random_state=88; total time=   0.7s
```

GRADIENT BOOSTED TREES

In [156…
```
reg = GradientBoostingRegressor(random_state=99)
reg.fit(X_train, y_train)
test_pred_reg, train_pred_reg = reg.predict(X_test), reg.predict(X_train)
```

Ensemble Model Blending

In [157…
```
train = pd.DataFrame({'Tesla_Stock_Price': y_train, 'val_pred_cart': train_pred_cart, 'val_pred_rf': train_pred_rf, 'val_pred_reg': train_pred_reg})
test = pd.DataFrame({'Tesla_Stock_Price': y_test, 'val_pred_cart': test_pred_cart, 'val_pred_rf': test_pred_rf, 'val_pred_reg': test_pred_reg})
ensemble_model = smf.ols(formula='Tesla_Stock_Price ~ val_pred_cart+val_pred_reg+val_pred_rf -1', data=train).fit()
```

Model Comparison

In [158…
```
comparison_data = {'Linear Regressor': ['{:.3f}'.format(OSR2(lrm, y_train, sm.add_constant(X_test), y_test)),
                                        '{:.4f}'.format(avg_error(lrm.predict(sm.add_constant(X_test)), y_test))],
                   'Decision Tree Regressor': ['{:.3f}'.format(OSR2(dtr_cv, y_train, X_test, y_test)),
                                        '{:.4f}'.format(avg_error(dtr_cv.predict(X_test), y_test))],
                  'Random Forest': ['{:.3f}'.format(OSR2(rf_cv, y_train, X_test, y_test)),
                                   '{:.4f}'.format(avg_error(rf_cv.predict(X_test), y_test))],
                  'Gradient Boosted Trees': ['{:.3f}'.format(OSR2(reg, y_train, X_test, y_test)),
                                         '{:.4f}'.format(avg_error(reg.predict(X_test), y_test))],
                 'Ensemble Model': ['{:.3f}'.format(OSR2(ensemble_model, y_train, test, y_test)),
                                         '{:.4f}'.format(avg_error(ensemble_model.predict(test), y_test))]}

comparison_table = pd.DataFrame(data=comparison_data, index=['OSR2', 'Average Percent Test Error'])
comparison_table
```

Out[158]:

	Linear Regressor	Decision Tree Regressor	Random Forest	Gradient Boosted Trees	Ensemble Model
OSR2	0.899	0.967	0.982	0.984	0.984
Average Percent Test Error	0.2401	0.0936	0.0711	0.0732	0.0737

In [ ]: