DS 2500 Project

Niam Bashambu, Adam Ancheta, TJ Kalapatapu

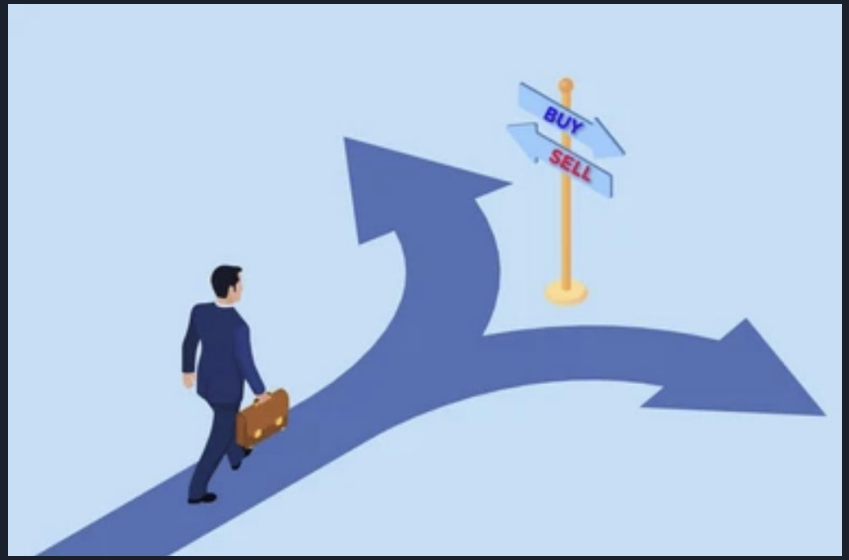
Table of Contents

1. What is our project?
2. How is this applicable?
3. What companies included in dataset?
4. How did we edit data?
5. Logistic Regression Model
6. Linear Regression Model
7. Linear Regression Stock Prediction

What is our project?

- Analyzed the stock data of 10 different companies
- We used different models to predict the future returns on the stock prices
- Created graphs to show predicted close prices of the stocks
- Allows investors to make a more accurate decision when putting their money into stocks to get the most money back

How is this applicable?



What companies are included in Data Set?

Uber

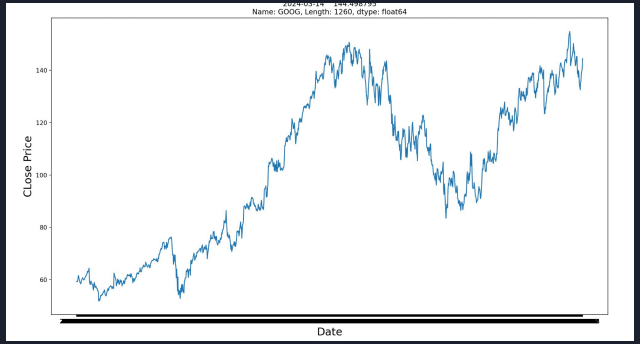
adidas

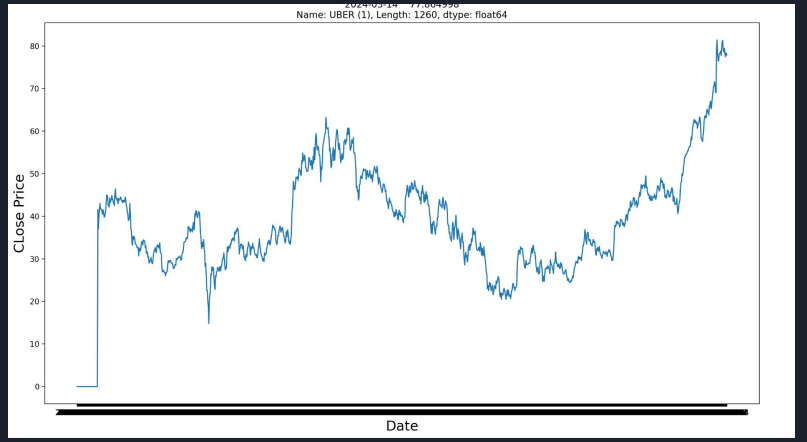
NIKE

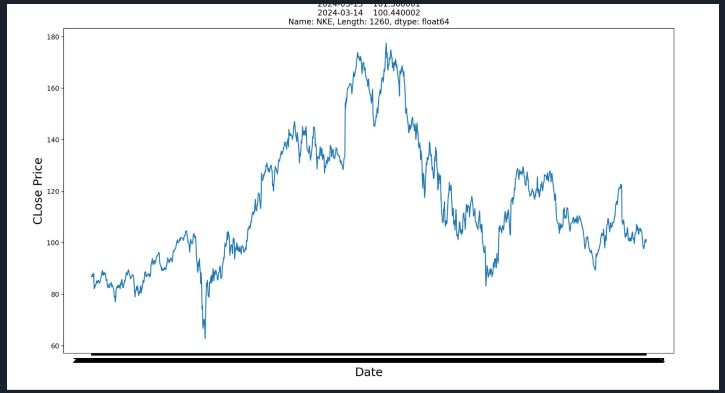
NVIDIA

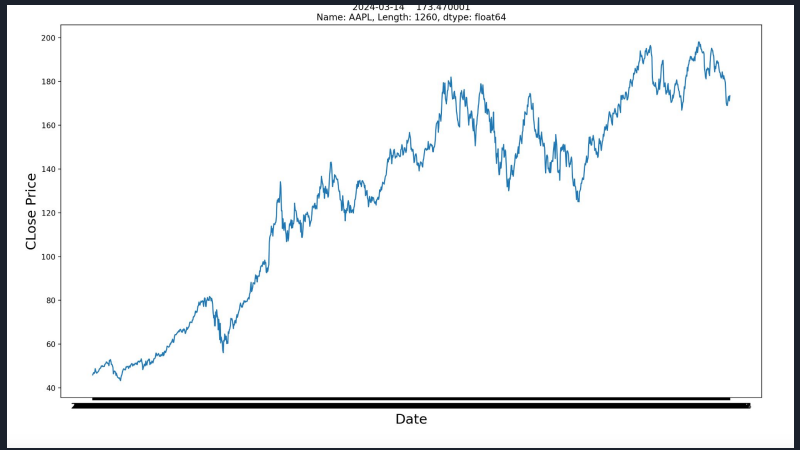
Meta

Our Datasets - Example Plots









How we edited our Data Set?

- Created a combined dataset of all 10 stocks
  - Using their individual csv files
- Dropped everything but Close Price
- Pivoted the dates to show the stocks in relation to one another

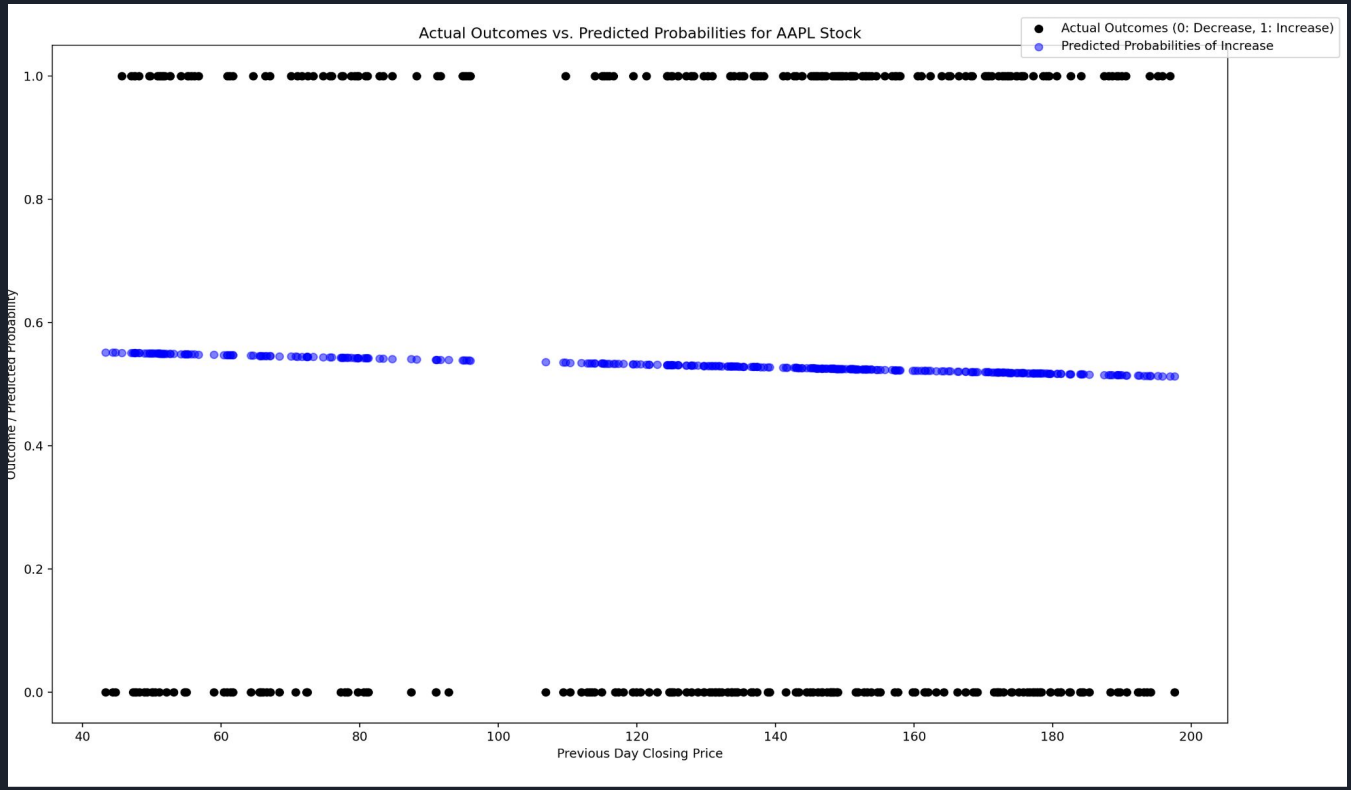
filename	AAPL	ADDYY	AMD	GOOG	INTC	META	MSFT	NKE	NVDA	UBER (1)
Date										
2019-03-14	45.932499	120.349998	22.820000	59.277500	53.439999	170.169998	114.589996	86.870003	41.389999	0.000000
2019-03-15	46.529999	120.410004	23.290001	59.223000	54.330002	165.979996	115.910004	86.800003	42.452499	0.000000
2019-03-18	47.005001	118.089996	23.250000	59.213001	54.099998	160.470001	117.570000	87.820000	42.237499	0.000000
2019-03-19	46.632500	119.099998	26.000000	59.942501	54.169998	161.570007	117.650002	87.690002	43.927502	0.000000
2019-03-20	47.040001	119.919998	25.700001	61.198502	53.820000	165.440002	117.519997	86.690002	43.599998	0.000000
...	...	...	...	...	...	...	...	...	...	...
2024-03-08	170.729996	102.959999	207.389999	136.289993	44.000000	505.950012	406.220001	99.160004	875.280029	78.699997
2024-03-11	172.750000	103.379997	198.389999	138.940002	44.860001	483.589996	404.519989	101.080002	857.739990	77.470001
2024-03-12	173.229996	105.349998	202.759995	139.619995	45.240002	499.750000	415.279999	100.180000	919.130005	78.320000
2024-03-13	171.130005	109.519997	194.789993	140.770004	43.230000	495.570007	415.100006	101.360001	908.880005	78.250000
2024-03-14	173.470001	112.043098	186.179596	144.498795	42.910000	494.170013	426.804993	100.440002	882.799988	77.864998

Logistic Regression Code

- X Axis: Previous Day Closing Price
- Y Axis: Outcome/Predicted Probability
- The predicted probabilities of increase for AAPL stock
    - Based on previous day's closing price

```
    ### this is a logistic regression model that converst the values of hte price to 0 and 1 determ
    stocks = ['AAPL', 'GOOG', 'UBER (1)', 'NKE']
    for stock in stocks:

        pivoted_df[stock + '_Diff'] = pivoted_df[stock].diff()

        pivoted_df[stock + '_Target'] = (pivoted_df[stock + '_Diff'] > 0).astype(int)

#  AAPL
    X = pivoted_df[['AAPL']].iloc[1:]
    y = pivoted_df['AAPL_Target'].iloc[1:]

# split into train test
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# create and train logistic regression model
    lr = LogisticRegression()
    lr.fit(X_train, y_train)

# evaluate said model
    predictions = lr.predict(X_test)
    print("Accuracy:", accuracy_score(y_test, predictions))

    predictions_proba = lr.predict_proba(X_test)[:, 1]
    print(predictions_proba)  #get probability
```

Logistic Regression Model

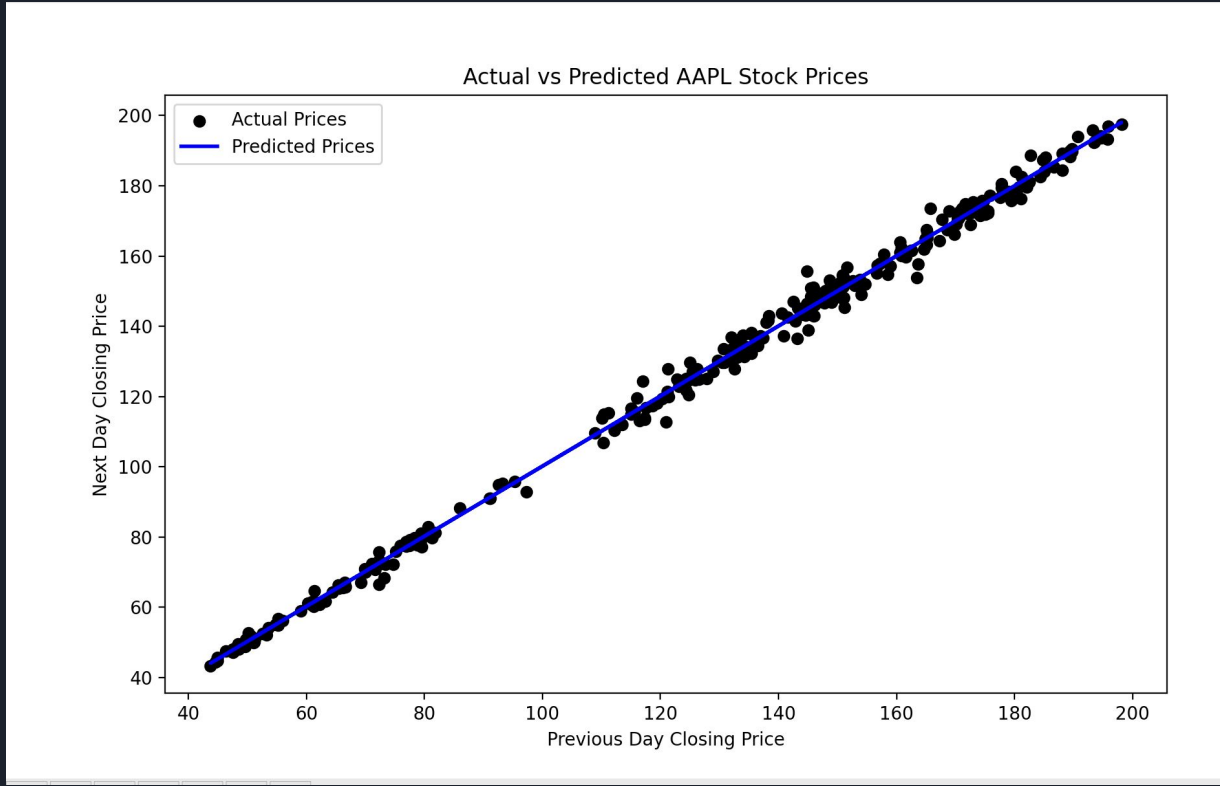


Linear Regression Code

- X Axis: Previous Day Closing Price
- Y Axis: Next Day Closing Price
- Linear Regression model
  - Predicted prices of a chosen stock
    - In this case AAPL
  - Shows the actual prices as a reference

```
    #making the linear regression model to show the predicted prices

    pivoted_df['AAPL_lagged'] = pivoted_df['AAPL'].shift(1)
    pivoted_df.dropna(inplace=True)  # Drop the first row which now contains NaN

# Features and target variable
    X = pivoted_df[['AAPL_lagged']]  # Features: Previous day's closing price
    y = pivoted_df['AAPL']


# split into train test
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

    lr = LinearRegression()
    lr.fit(X_train, y_train)
    predictions = lr.predict(X_test)
    actual = y_test
    #print(predictions)



    print(mean_squared_error(predictions,actual))

    #need to make the graphs


#Comparison of both linear regression in one graph
    plt.figure(figsize=(10, 6))
    plt.scatter(X_test, actual, color='black', label='Actual Prices')
    plt.plot(X_test, predictions, color='blue', linewidth=2, label='Predicted Prices')
    plt.title('Actual vs Predicted AAPL Stock Prices')
    plt.xlabel('Previous Day Closing Price')
    plt.ylabel('Next Day Closing Price')
    plt.legend()
    plt.show()
```

Linear Regression Model Stock Prediction



Linear Regression Stock Prediction

```
92  #doing the prediction logic
93  #obviously very basic as this is a linear regression model and we are only looking at close price and no other features
94  predicted_change_percentage = (predictions - X_test[stock_variable + '_lagged']) / X_test[stock_variable + '_lagged'] * 100
95
96  # Determine the average predicted change
97  average_predicted_change = predicted_change_percentage.mean()
98
99
100 # investment threshold
101 investment_threshold = 0.1  # Example threshold: 0.1% increase
102
103 if average_predicted_change >= investment_threshold:
104     print(f"The average predicted increase in the next day closing price for {stock_variable} is {average_predicted_change:.2f}%.")
105     print("This stock shows a promising upward trend.")
106 else:
107     print(f"The average predicted increase in the next day closing price for {stock_variable} is {average_predicted_change:.2f}%.")
108     print("This stock does not show a strong enough upward trend based on our model's predictions.")
```

Input Feature:

Stock chosen

```
[1260 rows x 10 columns]
Give a stock value that you want to do the model on AAPL
```

Output Feature:

```
The average predicted increase in the next day closing price for AAPL is 0.16%.
This stock shows a promising upward trend.
```

Thank you for listening!
Have any Questions?