Hacking The Big Numbers

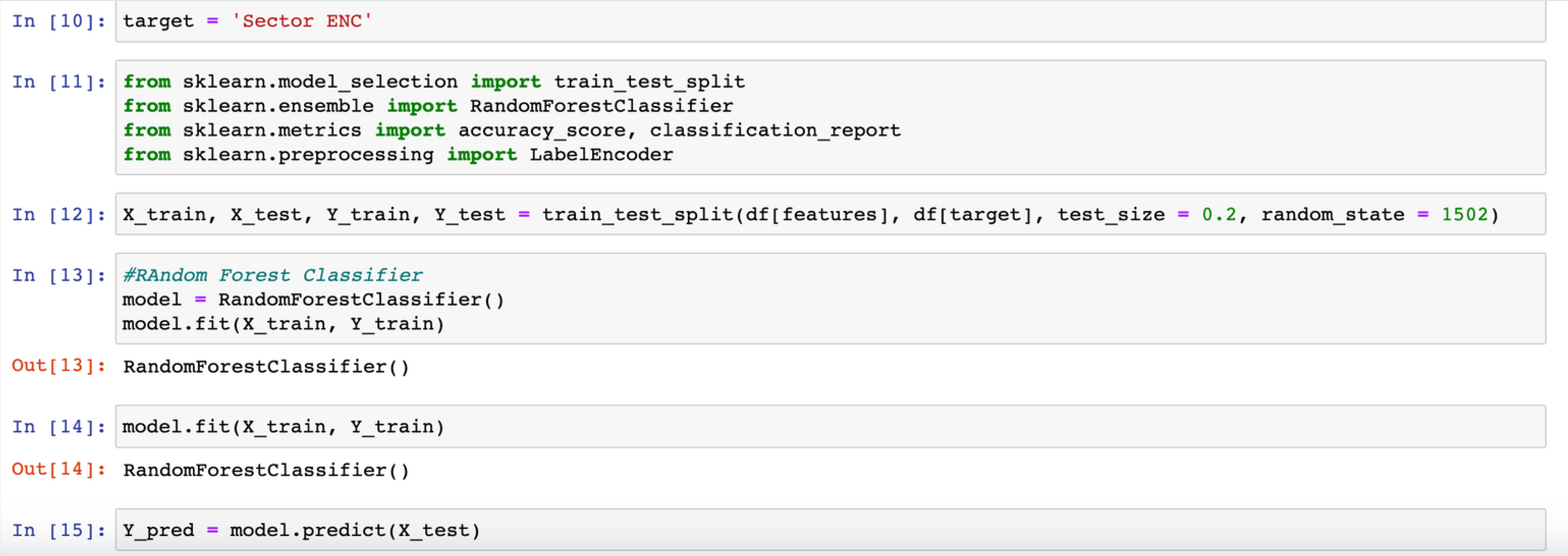
EBR project

Serban Timofte

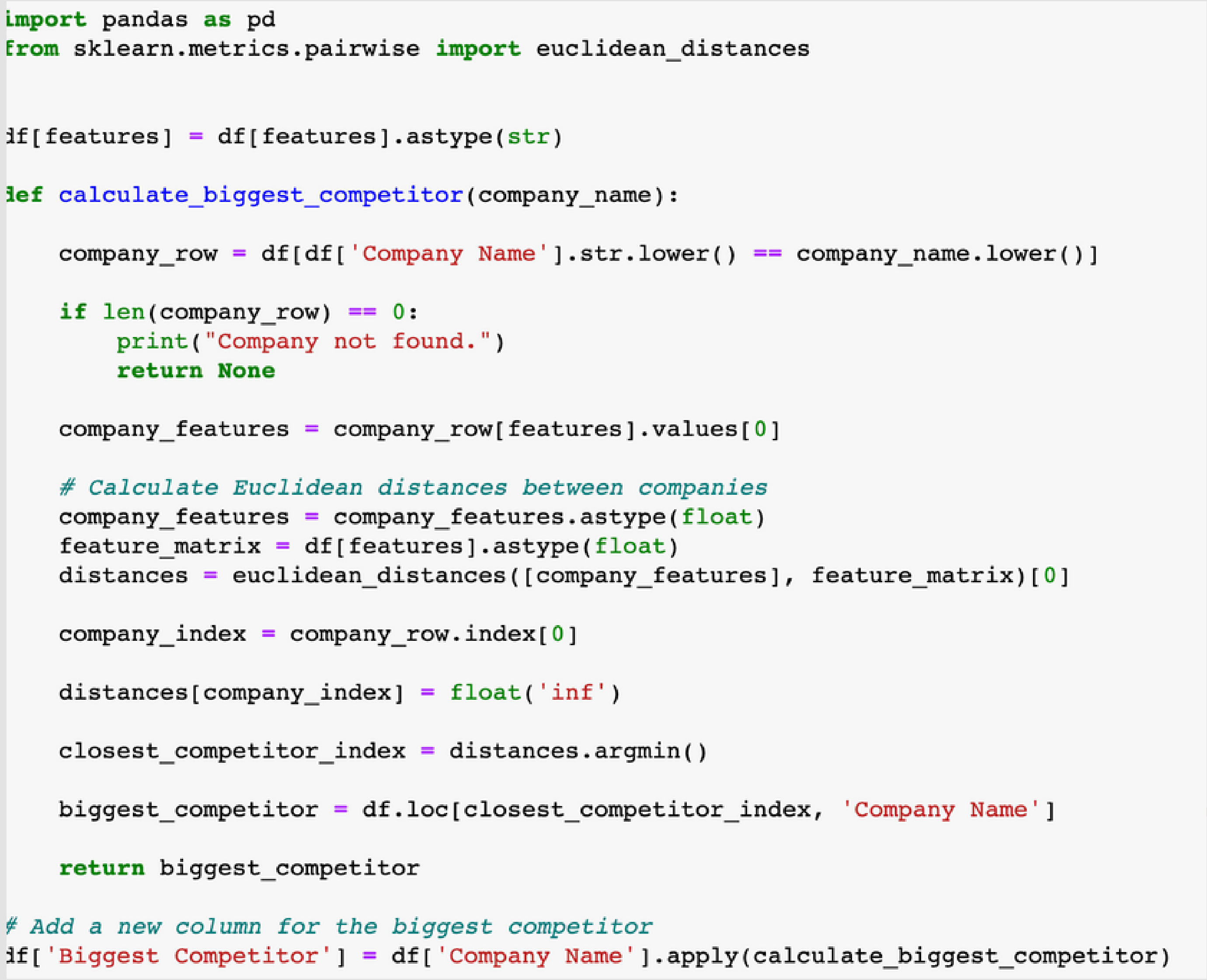
2 3 4

1

Classification & Competition



```
In [10]: target = 'Sector ENC'

In [11]: from sklearn.model_selection import train_test_split
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy_score, classification_report
         from sklearn.preprocessing import LabelEncoder

In [12]: X_train, X_test, Y_train, Y_test = train_test_split(df[features], df[target], test_size = 0.2, random_state = 1502)

In [13]: #RAndom Forest Classifier
         model = RandomForestClassifier()
         model.fit(X_train, Y_train)

Out[13]: RandomForestClassifier()

In [14]: model.fit(X_train, Y_train)

Out[14]: RandomForestClassifier()

In [15]: Y_pred = model.predict(X_test)
```

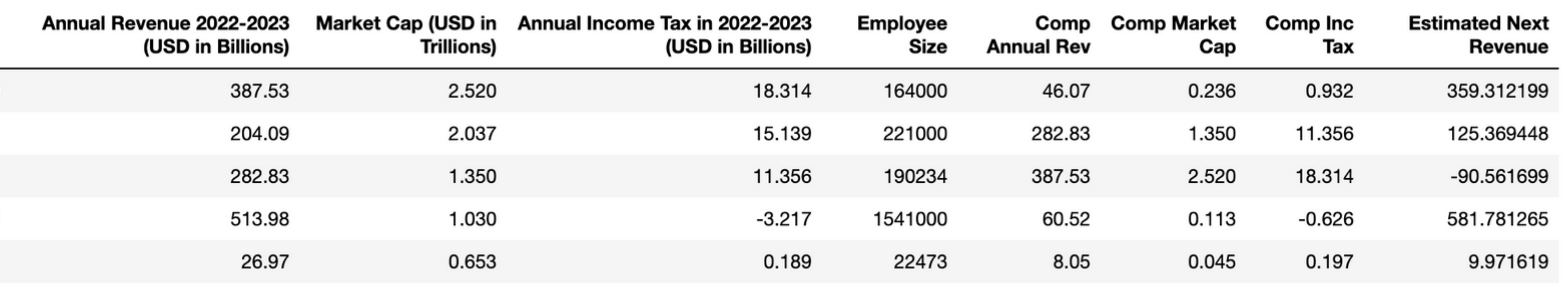


```
import pandas as pd
from sklearn.metrics.pairwise import euclidean_distances


if[features] = df[features].astype(str)

lef calculate_biggest_competitor(company_name):

    company_row = df[df['Company Name'].str.lower() == company_name.lower()]

    if len(company_row) == 0:
        print("Company not found.")
        return None

    company_features = company_row[features].values[0]

    # Calculate Euclidean distances between companies
    company_features = company_features.astype(float)
    feature_matrix = df[features].astype(float)
    distances = euclidean_distances([company_features], feature_matrix)[0]

    company_index = company_row.index[0]

    distances[company_index] = float('inf')

    closest_competitor_index = distances.argmin()

    biggest_competitor = df.loc[closest_competitor_index, 'Company Name']

    return biggest_competitor

# Add a new column for the biggest competitor
if['Biggest Competitor'] = df['Company Name'].apply(calculate_biggest_competitor)
```

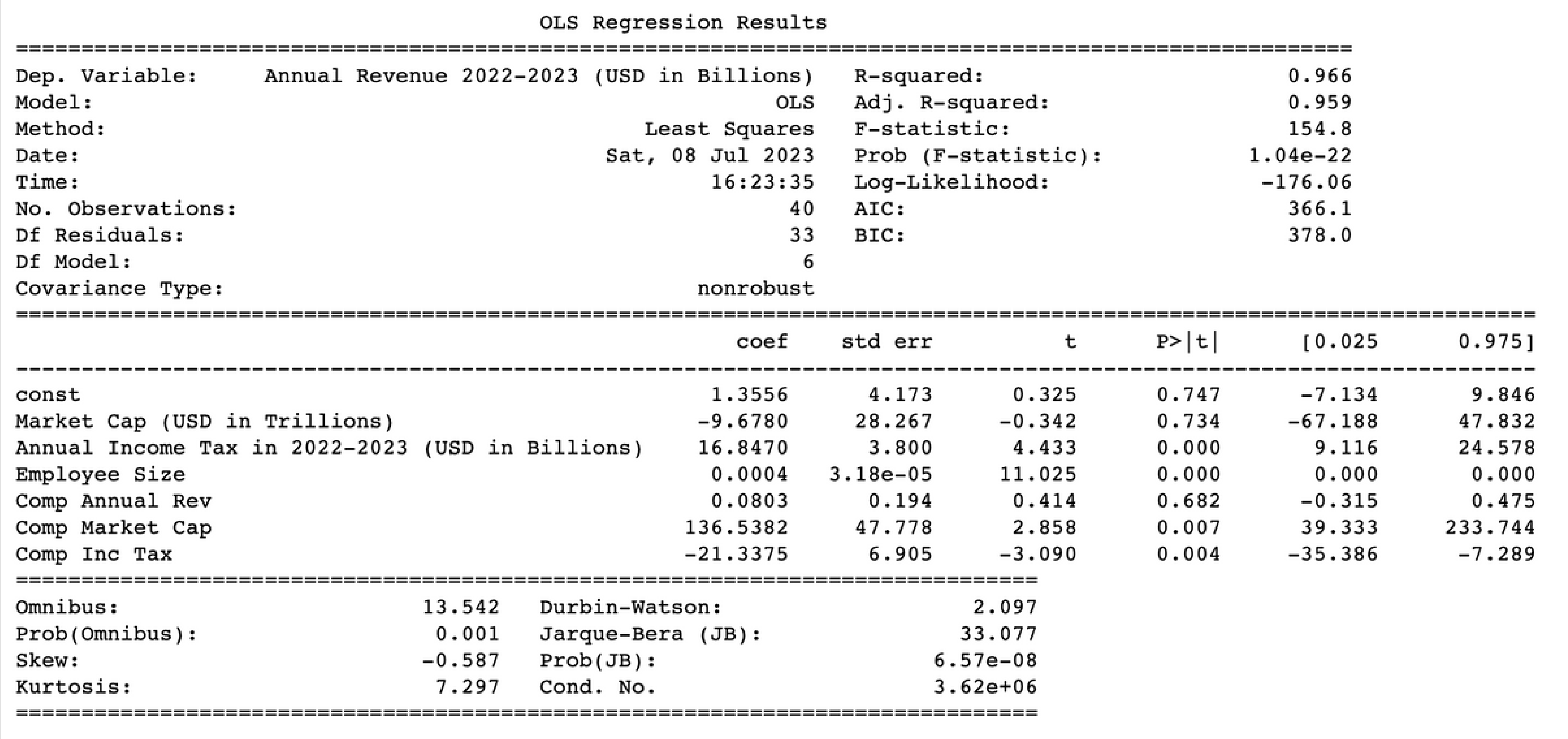
Company	**Biggest Resulted Competitor**
Microsoft Corporation	Alphabet (Google)
NVIDIA Corporation	Microchip Tech Inc.
Intel Corporation	Dell Technologies

3 4

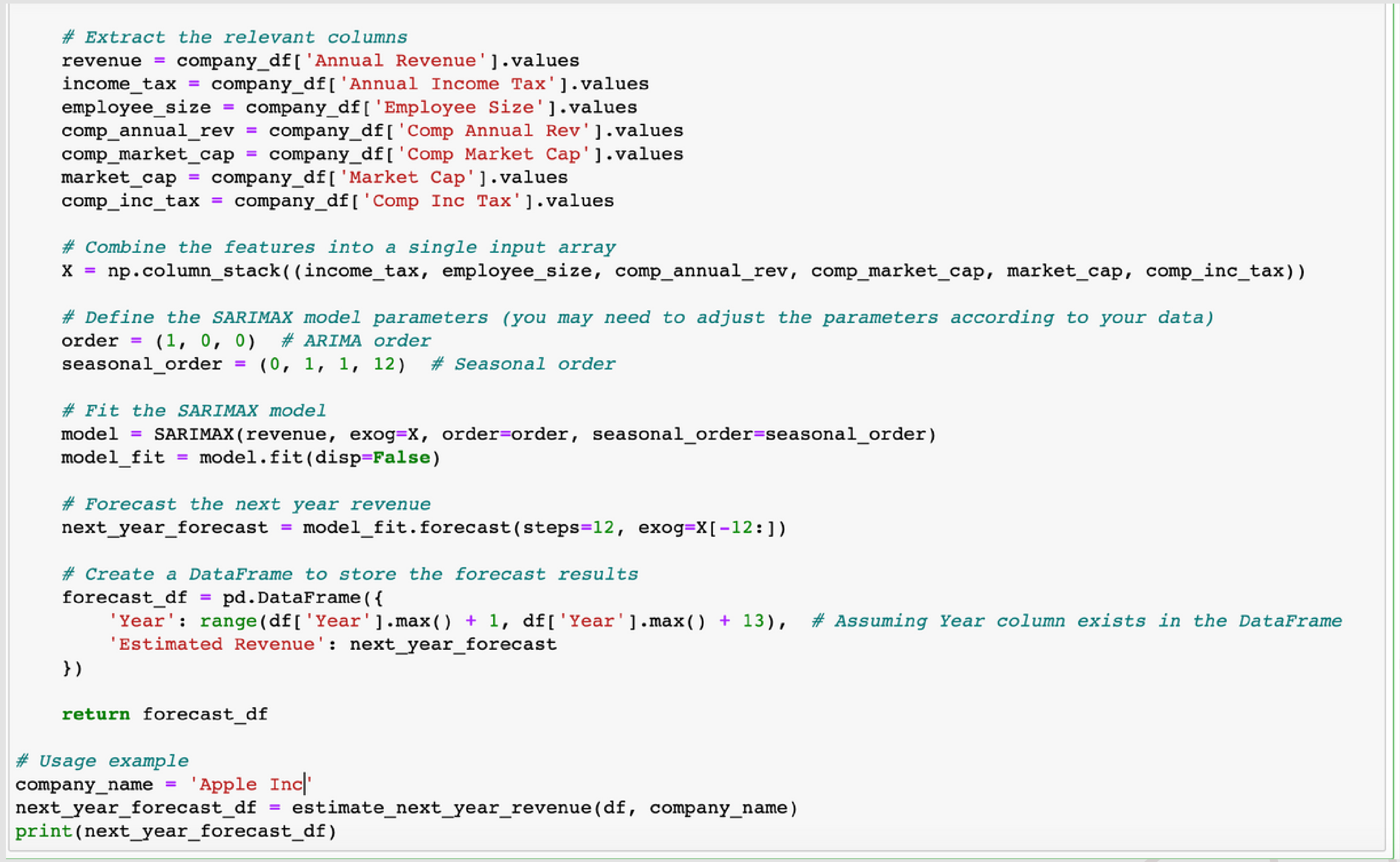
2 Fiscal Impact of Competition



Annual Revenue 2022-2023 (USD in Billions)	Market Cap (USD in Trillions)	Annual Income Tax in 2022-2023 (USD in Billions)	Employee Size	Comp Annual Rev	Comp Market Cap	Comp Inc Tax	Estimated Next Revenue
387.53	2.520	18.314	164000	46.07	0.236	0.932	359.312199
204.09	2.037	15.139	221000	282.83	1.350	11.356	125.369448
282.83	1.350	11.356	190234	387.53	2.520	18.314	-90.561699
513.98	1.030	-3.217	1541000	60.52	0.113	-0.626	581.781265
26.97	0.653	0.189	22473	8.05	0.045	0.197	9.971619



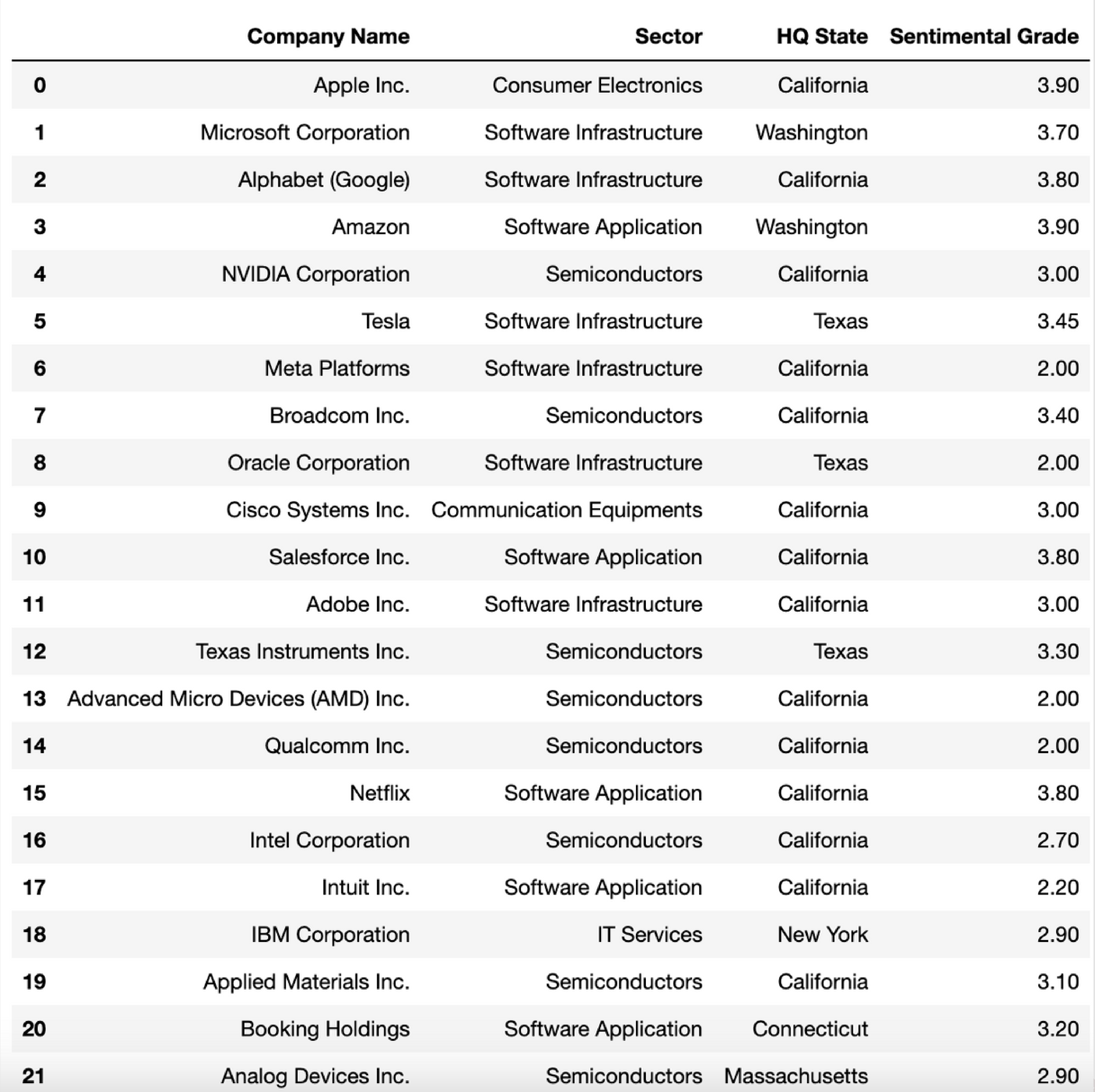
```
                                          OLS Regression Results
==============================================================================================
Dep. Variable:     Annual Revenue 2022-2023 (USD in Billions)   R-squared:                       0.966
Model:                                                    OLS   Adj. R-squared:                  0.959
Method:                                         Least Squares   F-statistic:                     154.8
Date:                                        Sat, 08 Jul 2023   Prob (F-statistic):           1.04e-22
Time:                                                16:23:35   Log-Likelihood:                -176.06
No. Observations:                                          40   AIC:                             366.1
Df Residuals:                                              33   BIC:                             378.0
Df Model:                                                   6
Covariance Type:                                    nonrobust
===============================================================================================================
                                                coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------------
const                                         1.3556      4.173      0.325      0.747      -7.134       9.846
Market Cap (USD in Trillions)                -9.6780     28.267     -0.342      0.734     -67.188      47.832
Annual Income Tax in 2022-2023 (USD in Billions)    16.8470      3.800      4.433      0.000       9.116      24.578
Employee Size                                 0.0004   3.18e-05     11.025      0.000       0.000       0.000
Comp Annual Rev                               0.0803      0.194      0.414      0.682      -0.315       0.475
Comp Market Cap                             136.5382     47.778      2.858      0.007      39.333     233.744
Comp Inc Tax                                -21.3375      6.905     -3.090      0.004     -35.386      -7.289
==============================================================================
Omnibus:                       13.542   Durbin-Watson:                   2.097
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               33.077
Skew:                          -0.587   Prob(JB):                     6.57e-08
Kurtosis:                       7.297   Cond. No.                     3.62e+06
==============================================================================
```



```python
    # Extract the relevant columns
    revenue = company_df['Annual Revenue'].values
    income_tax = company_df['Annual Income Tax'].values
    employee_size = company_df['Employee Size'].values
    comp_annual_rev = company_df['Comp Annual Rev'].values
    comp_market_cap = company_df['Comp Market Cap'].values
    market_cap = company_df['Market Cap'].values
    comp_inc_tax = company_df['Comp Inc Tax'].values

    # Combine the features into a single input array
    X = np.column_stack((income_tax, employee_size, comp_annual_rev, comp_market_cap, market_cap, comp_inc_tax))

    # Define the SARIMAX model parameters (you may need to adjust the parameters according to your data)
    order = (1, 0, 0)  # ARIMA order
    seasonal_order = (0, 1, 1, 12)  # Seasonal order

    # Fit the SARIMAX model
    model = SARIMAX(revenue, exog=X, order=order, seasonal_order=seasonal_order)
    model_fit = model.fit(disp=False)

    # Forecast the next year revenue
    next_year_forecast = model_fit.forecast(steps=12, exog=X[-12:])

    # Create a DataFrame to store the forecast results
    forecast_df = pd.DataFrame({
        'Year': range(df['Year'].max() + 1, df['Year'].max() + 13),  # Assuming Year column exists in the DataFrame
        'Estimated Revenue': next_year_forecast
    })

    return forecast_df

# Usage example
company_name = 'Apple Inc'
next_year_forecast_df = estimate_next_year_revenue(df, company_name)
print(next_year_forecast_df)
```

3 Twitter Sentimental Analytics

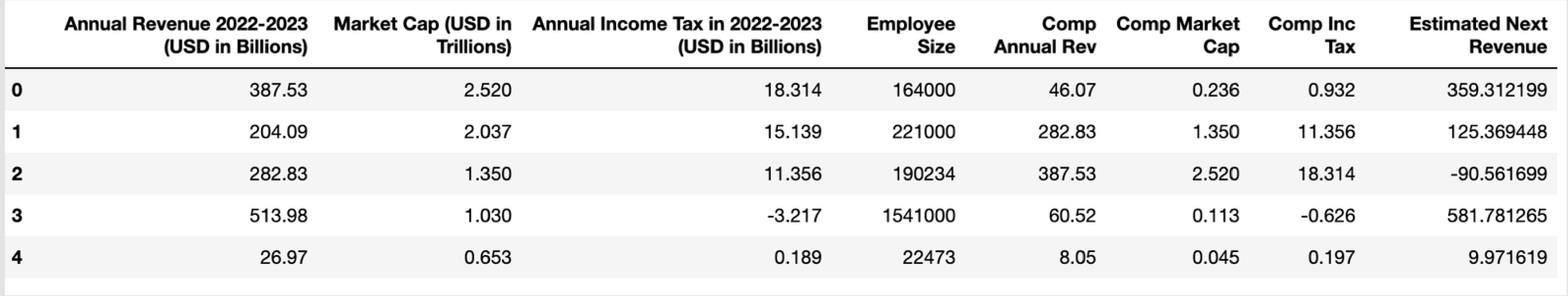
Adding a dataset with Engagement, Number of Likes, Comments and Shares of the companies on Twitter, I created a social media score using SVM model



	Company Name	Sector	HQ State	Sentimental Grade
0	Apple Inc.	Consumer Electronics	California	3.90
1	Microsoft Corporation	Software Infrastructure	Washington	3.70
2	Alphabet (Google)	Software Infrastructure	California	3.80
3	Amazon	Software Application	Washington	3.90
4	NVIDIA Corporation	Semiconductors	California	3.00
5	Tesla	Software Infrastructure	Texas	3.45
6	Meta Platforms	Software Infrastructure	California	2.00
7	Broadcom Inc.	Semiconductors	California	3.40
8	Oracle Corporation	Software Infrastructure	Texas	2.00
9	Cisco Systems Inc.	Communication Equipments	California	3.00
10	Salesforce Inc.	Software Application	California	3.80
11	Adobe Inc.	Software Infrastructure	California	3.00
12	Texas Instruments Inc.	Semiconductors	Texas	3.30
13	Advanced Micro Devices (AMD) Inc.	Semiconductors	California	2.00
14	Qualcomm Inc.	Semiconductors	California	2.00
15	Netflix	Software Application	California	3.80
16	Intel Corporation	Semiconductors	California	2.70
17	Intuit Inc.	Software Application	California	2.20
18	IBM Corporation	IT Services	New York	2.90
19	Applied Materials Inc.	Semiconductors	California	3.10
20	Booking Holdings	Software Application	Connecticut	3.20
21	Analog Devices Inc.	Semiconductors	Massachusetts	2.90

4 Refactoring the simulation with TW Impact Score

Even the differences between companies in TW Sentimental Grade were important, these does not significantly affect firms' revenue predictions!



	Annual Revenue 2022-2023 (USD in Billions)	Market Cap (USD in Trillions)	Annual Income Tax in 2022-2023 (USD in Billions)	Employee Size	Comp Annual Rev	Comp Market Cap	Comp Inc Tax	Estimated Next Revenue
0	387.53	2.520	18.314	164000	46.07	0.236	0.932	359.312199
1	204.09	2.037	15.139	221000	282.83	1.350	11.356	125.369448
2	282.83	1.350	11.356	190234	387.53	2.520	18.314	-90.561699
3	513.98	1.030	-3.217	1541000	60.52	0.113	-0.626	581.781265
4	26.97	0.653	0.189	22473	8.05	0.045	0.197	9.971619

- So, how efficient there are social media platforms for big players?
- If they give up at social media, the community will take their place?

WHAT OTHER CONCLUSIONS DO YOU THINK WE SHOULD ANALYZE?