CA 4010 - Data Mining and Data Warehousing Assignment

Eoin Clayton – 16326173

Conor Reilly – 16478326

18/12/2019

Table of Contents

1 Introduction

The objective of our project was to analyse and predict future results of the Fortune 500 list of companies. The Fortune 500 is a list of the 500 biggest corporations in the US based on several aspects we will cover. This report includes our thought process, the technologies we used throughout and the results we found using various data analysis and data mining techniques.

1.1 Glossary

Python - The Programming Language we used to handle algorithms and data handling

Matplotlib - The Library used to implement graphs, pie charts and line charts

UiPath - The program used to web scrape information

MySQL - The program used to manage database information

2 General Description

From the initial research we did into our project, we knew that this idea was going to be a difficult task. The Fortune 500 can be highly unpredictable at times. For example a company could be sold or go bankrupt in the space of a year and predicting this can be extremely difficult. We look to create a dataset of data from several years of Fortune 500 lists and make several predictions from this information.

2.1 Prediction Description

We found that there are five key aspects that determine the list of companies; Total Revenue, Total Profits, Company Assets, Number of Employees and Market Evaluation. We decided that these were the attributes we would look to help us determine our predictions. We aim to predict future Fortune 500 lists as well as how these aspects determine trends in the overall scheme. Dealing with outliers and anomalies is also another key aspect of our project.

3. Dataset Information

3.1 Dataset Compilation

To create our dataset we had to overcome many difficulties. The initial dataset we had intended to use was one we had obtained on GitHub. However there were several issues with the dataset, most obvious was that it was missing key attributes that we needed for data analysis.

rank	company	revenue ($ millions)	profit ($ millions)
1	Walmart	500343.00	9862.00
2	Exxon Mobil	244363.00	19710.00
3	Berkshire Hathaway	242137.00	44940.00
4	Apple	229234.00	48351.00
5	UnitedHealth Group	201159.00	10558.00
6	McKesson	198533.00	5070.00
7	CVS Health	184765.00	6622.00
8	Amazon.com	177866.00	3033.00
9	AT&T	160546.00	29450.00
10	General Motors	157311.00	-3864.00

Fig 3.11. https://github.com/cmusam/fortune500/blob/master/2015-2018/output/fortune500-2018.csv

As you can see we are missing attributes such as Total Profits, Number of Employees and Total Market Value. These were listed as key aspects of our project data analysis and caused us to look elsewhere to gather more data.

We found several resources but very few that satisfied all of our needs in one dataset, this led us to instead webscrape all data from the official Fortune500.com website and combine results we had already from our initial dataset. To do this we used UiPath and MySQL to help create our official database for the project.

We decided to use UiPath as our main webscraper as Eoin had previous experience using this while on his INTRA Placement. This gave a great advantage to getting started with building our dataset. We chose MySQL to manage and manipulate our data as we both had previous experience using this on INTRA and our previous module Introduction to Databases.

The webscraping aspect of our project became an unexpectedly difficult section of our project. We had not realised that the data from Fortune500.com changes slightly every year in the layout of page and certain requirements.

RANK	NAME	REVENUES ($M)	REVENUE PERCENT CHANGE	PROFITS ($M)	PROFITS PERCENT CHANGE	ASSETS ($M)	MARKET VALUE — AS OF MARCH 29, 2019 ($M)	CHANGE IN RANK (FULL 1000)	EMPLOYEES	CHANGE IN RANK (500 ONLY)
1	Walmart	$514,405.0	2.8%	$6,670.0	-32.4%	$219,295.0	$279,880.3	-	2,200,000	-
2	Exxon Mobil	$290,212.0	18.8%	$20,840.0	5.7%	$346,196.0	$342,172.0	-	71,000	-
3	Apple	$265,595.0	15.9%	$59,531.0	23.1%	$365,725.0	$895,667.4	1	132,000	1
4	Berkshire Hathaway	$247,837.0	2.4%	$4,021.0	-91.1%	$707,794.0	$493,870.3	-1	389,000	-1

Fig 3.12. 2018 Fortune 500 - https://fortune.com/fortune500/2018/search/

RANK	NAME	REVENUES	REVENUE PERCENT CHANGE	PROFITS	PROFITS PERCENT CHANGE	TOTAL ASSETS	TOTAL SHAREHOLDER EQUITY	MARKET VALUE (ON MARCH 31, 2014)	SALES	ASSETS	STOCKHOLDERS' EQUITY	PROFIT AS A % OF SALES
1	Wal-Mart Stores	476,294	1.5	16022	-5.7	204,751	76,255	246,805	3.4	7.8	21	3.4
2	Exxon Mobil	407,666	-9.4	32580	-27.4	346,808	174,003	422,098	8	9.4	18.7	8
3	Chevron	220,356	-5.8	21423	-18.2	253,753	149,113	227,014	9.7	8.4	14.4	9.7
4	Berkshire Hathaway	182,150	12.1	19476	31.4	484,931	221,890	308,003	10.7	4	8.8	10.7

Fig 3.13 2014 Fortune 500 https://fortune.com/fortune500/2014/search/

As you can see several aspects of the official list change significantly, especially when looking at the right hand columns. We can see that some columns remain in the same positions while other change position such as Assets moving significanlty. We also found that column names also changed which led to more difficulties, as you can see Assets changes to Total Assets depending on the year.

Another difficulty we encountered was that certain years took different aspects into consideration. For example in 2018 and Fig 3.12 we can see Employees listed as a column heading and attribute whereas in 2014 Number of Employees was not taken into consideration. This led to a change in our dataset, data preparation and our final calculations.

We then had a dataset that included our five attributes we initially looked to analyse. However, we still had to prepare our data for accurate testing.

Rank	Name	Revenues ($M)	Profits ($M)	Assets ($M)	Employees	Market Value
1	Walmart	$482,130	$14,694	$199,581	2,300,000	$215,356
2	Apple	$233,715	$53,394	$290,479	110,000	$604,304
3	Berkshire Hathaway	$210,821	$24,083	$552,257	331,000	$350,279
4	McKesson	$181,241	$1,476	$53,870	70,400	$35,945
5	UnitedHealth Group	$157,107	$5,813	$111,383	200,000	$122,542
6	CVS Health	$153,290	$5,237	$93,657	199,000	$113,947
7	General Motors	$152,356	$9,687	$194,520	215,000	$48,543
8	Ford Motor	$149,558	$7,373	$224,925	199,000	$53,758
9	AT&T	$146,801	$13,345	$402,672	281,450	$240,943
10	General Electric	$140,389	$-6,126	$492,692	333,000	$295,174
11	AmerisourceBergen	$135,962	$-135	$27,736	17,000	$19,511
12	Verizon	$131,620	$17,879	$244,640	177,700	$220,646
13	Chevron	$131,118	$4,587	$266,103	61,500	$179,653
14	Costco	$116,199	$2,377	$33,440	161,000	$69,183
15	Fannie Mae	$110,359	$10,954	$3,221,917	7,300	$1,621

Fig 3.14 2014 Final Dataset for Initial Testing

3.2 Dataset Preparation

Given we had already webscraped the columns we intended to use, we didn't need to remove any attributes in that area. We did however notice some irregularities in our final datasets.

We found that the names of corporations could change slightly over time. For example in Fig 3.12 and Fig 3.13 we can see "Walmart" changes to "Wal-mart Stores." This lead to us making very difficult changes that were crucial to our calculations. We had to create a program that would link the most likely companies from year to year. We took into account their names and previous positioning to ensure we had the correct links.

Another issue we encountered was for several companies there was not listed Market Value, for example in 2018 there was 14 accounts of this occurring. We are not sure as to why this is but we noticed a trend was that a majority of these were insurance based companies. To amend this we manually searched for companies each individual Market Value on Google. Although this may make our data slightly less accurate, it was a compensation we deemed necessary and given the small number of occurrences it affected we were ok will the results.

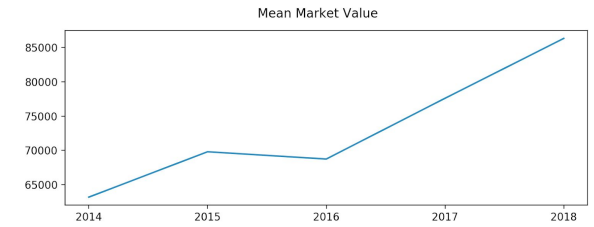
We also began to remove things like commas, dollar signs and other symbols to allow for easier calculations and parsing of data.

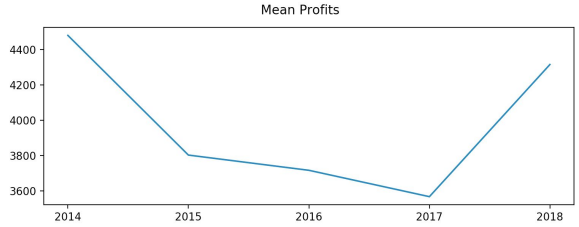
```
with open('mode.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    row1 = next(csv_reader)
    row2 = next(csv_reader)
    row3 = next(csv_reader)
    row4 = next(csv_reader)
    row5 = next(csv_reader)

    labels = []
    year = []
    row1Values = []
    row2Values = []
    row3Values = []
    row4Values = []
    row5Values = []


    for k in row1.keys():
        if k != "":
            labels.append(k)
        if k == "":
            year.append(row1[k])
        else:
            row1Values.append(row1[k])
```

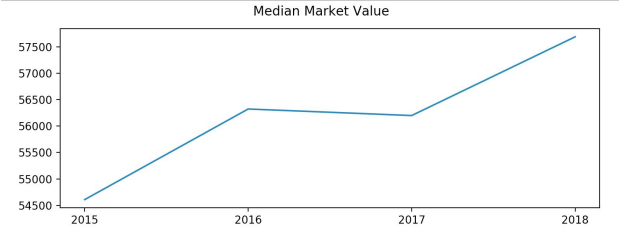
3.3 Dataset Analysis

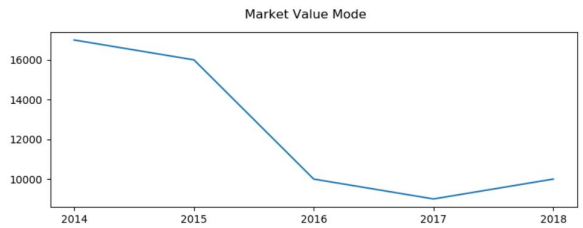
Our initial workshop was to test the different central tendencies of our dataset. This included the mean, median and mode of each of our aspects we wished to evaluate. We decided to test whether the overall central tendencies were increasing or decreasing over the years.

We found that on average, each central tendency was on average increasing from 2014. This meant that each year Total Revenue, Company Assets, Market Value, Number of Employees and Profits increased in most occurrences. This was initially surprising to us as we thought there would be less of a trend in increases and more of a stable analysis would be expected.









4 Algorithm Research

When researching for the best algorithm to predict future results, we decided to go for a linear regression based approach, to do this we took years 2014 to 2018 as our training data and left 2019 as our testing data. This means we had 83% of our data for training our model and 17% for testing it.

To train and test our data we used python and sklearn to handle the algorithms and predictions. We also used pandas library to handle the database and csv file operations.

Linear regression is a commonly used type of predictive analysis. The idea of regression is mainly to examine two things: 1. Does a set of predictor variables do a good job in predicting an outcome variable? 2. Which variables are significant to predicting the outcome of a desired variable. We chose this model after doing research and finding it is the most common type of search when dealing with scenarios involving finances for example: stock markets and market share predictors.

4.1 Implementation

The first step was to read in our data from a csv file and split all the attributes into their own array for each individual company. To do this we had to iterate through each of the rows in the dataset, each time storing the necessary data into the relevant array to be used later in the program.

```
regression_model = LinearRegression()

dataset = pd.read_csv('dataset.csv')

Date = np.array([[2015],[2016], [2017], [2018]])

for index, row in dataset.iterrows():

    Name = row['Name']

    Revenue = np.array([row['Revenues 2015'], row['Revenues 2016'], row['Revenues 2017'], row['Revenues 2018']])
    Profits = np.array([row['Profits 2015'], row['Profits 2016'], row['Profits 2017'], row['Profits 2018']])
    Assets = np.array([row['Assets 2015'], row['Assets 2016'], row['Assets 2017'], row['Assets 2018']])
    Employees = np.array([row['Employees 2015'], row['Employees 2016'], row['Employees 2017'], row['Employees 2018']])
    MarketValue = np.array([row['Market Value 2015'], row['Market Value 2016'], row['Market Value 2017'], row['Market Value 2018']])
```

Fig 4.11 Initial Iteration through our csv file.

We then trained our model using this data using the sklean framework. We also applied weights to each of the attributes depending on how important we saw this data. From our research in the previous weeks we determined that Revenue and Market Value were main determining factor as to how the list was determined. This meant these were the most important attributes we were looking for when predicting our future list. As you can see from Fig.11 we have also implemented LinearRegression() to help us with our predictions.

After training our model to sort what we thought was the best way possible, we began to predict each of Revenue, Profits, Assets, Employees and Market Value. Again we used the sklearn library and it's predict() function.

```
regression_model.fit(Date, Revenue)
Revenue_predict = regression_model.predict([[2019]])

#Profits
regression_model.fit(Date, Profits)
Profit_predict = regression_model.predict([[2019]])

#Assets
regression_model.fit(Date, Assets)
Assets_predict = regression_model.predict([[2019]])

#Employees
regression_model.fit(Date, Employees)
Employees_predict = regression_model.predict([[2019]])

#Market Value
regression_model.fit(Date, MarketValue)
MV_predict = regression_model.predict([[2019]])
```

Fig 4.12 Our predictions of each attribute

A new dataframe then had to be created to test our prediction against the 2019 results. We created this dataframe after the first prediction was made and followed the same structure as the previous dataset by including Name, Revenue, Profits, Assets, Employees and Market Value.

We excluded the rank as this had to be calculated later. Once the dataframe was created we were able to append the rest of the predictions as soon as they were made.

The next step was to sort the dataframe. We sorted this using weighted attributes calculations as previously mentioned, from this we had our final ranking of companies in the Fortune 500 list.

```
    Rank = row['Rank']

    if row['Rank'] == 1:
        Data = pd.DataFrame({
            "Name": Name,
            "Revenue": Revenue_predict,
            "Profits": Profit_predict,
            "Assets": Assets_predict,
            "Employees": Employees_predict,
            "Market Value": MV_predict
            })
        df = DataFrame(Data, columns = ["Name", "Revenue", "Profits", "Assets", "Employees", "Market Value"])

    else:
        Data = pd.DataFrame({

            "Name": Name,
            "Revenue": Revenue_predict,
            "Profits": Profit_predict,
            "Assets": Assets_predict,
            "Employees": Employees_predict,
            "Market Value": MV_predict
            })
        df2 = df.append(DataFrame(Data, columns = ["Name", "Revenue", "Profits", "Assets", "Employees", "Market Value"]))
        df = df2


final_df = df.sort_values(by=['Revenue'], ascending=False)
```

Fig. 4.13 Creation of predicted 2019 list

Here is a sample from our 2019 prediction dataset.

Rank	Name	Revenue	Profits	Assets	Employees	Market Value
1	Walmart	514,405.00	6,670.00	219,295.00	2200000	279,880.30
2	Exxon Mobil	290,212.00	20,840.00	346,196.00	71000	342,172.00
3	Apple	265,595.00	59,531.00	365,725.00	132000	895,667.40
4	Berkshire Ha	247,837.00	4,021.00	707,794.00	389000	493,870.30
5	Amazon.com	232,887.00	10,073.00	162,648.00	647500	874,709.50
6	UnitedHealth	226,247.00	11,986.00	152,221.00	300000	237,255.50
7	McKesson	208,357.00	67	60,381.00	68000	22,455.10
8	CVS Health	194,579.00	-594	196,456.00	295000	69,951.60
9	AT&T	170,756.00	19,370.00	531,864.00	268220	228,444.70
10	Amerisource	167,939.60	1,658.40	37,669.80	20500	16,785.90
11	Chevron	166,339.00	14,824.00	253,863.00	48600	234,049.70
12	Ford Motor	160,338.00	3,677.00	256,540.00	199000	35,028.00
13	General Mot	147,049.00	8,014.00	227,339.00	173000	52,291.70
14	Costco Whol	141,576.00	3,134.00	40,830.00	194000	106,512.60
15	Alphabet	136,819.00	30,736.00	232,792.00	98771	816,824.20
16	Cardinal Hea	136,809.00	256	39,951.00	50200	14,349.50
17	Walgreens B	131,537.00	5,024.00	68,124.00	299000	59,691.70
18	JPMorgan Ch	131,412.00	32,474.00	2,622,532.00	256105	331,451.50

Fig 4.14 2019 Fortune 500 Predictions

5. Results and Analysis

This section contains our results we obtained and how we further analyzed this data to gain a better understanding of these results. This section includes;

- Initial Analysis
- Cross Validation for Linear Regression
- Coefficient of determination.

5.1 Initial Analysis

The results of our analysis weren't as expected. We were able to output a prediction for the total revenue, profits, assets, employees, market value and the rank of the company as we intended but the accuracy of the rank was a lot harder to predict than anticipated. Our rank of predicted companies had a low percentage of matches to the official 2019 fortune 500 list, however we were able to predict a higher percentage of companies to within 10 places of the official rank. (eg. 2 of our top 10 matched perfectly, however we predicted 9 companies that are in the official top 10.)

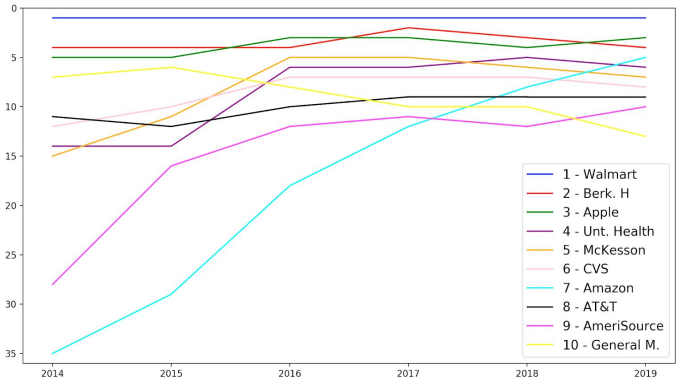


Fig 5.11. Predicted Top 10

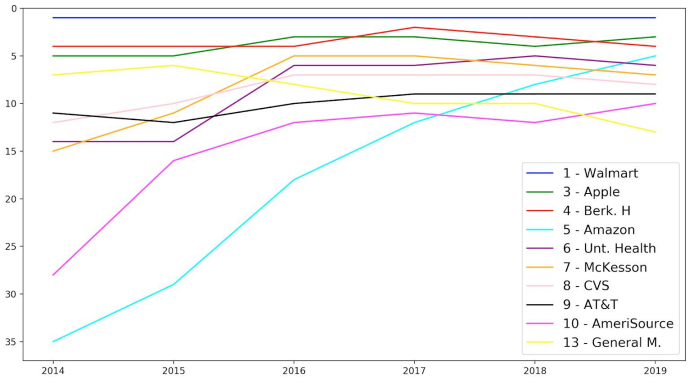


Fig 5.12 Actual Placement of these 10

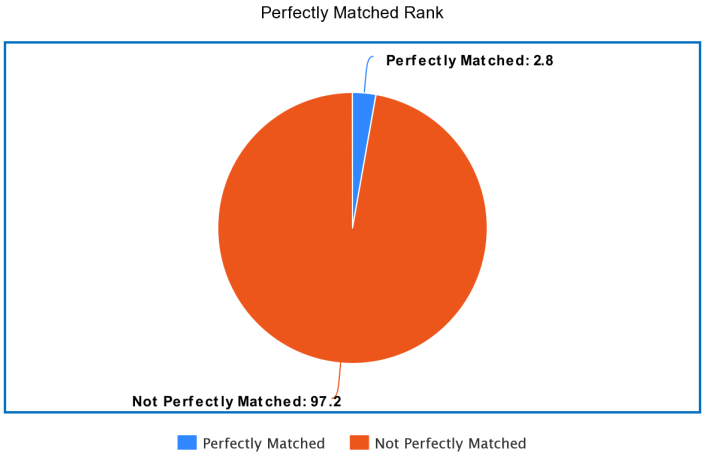
This indicated that we could not predict the exact placement of companies on the list accurately. However, if we were to broaden the range of our searches we could possibly find better results. Although our initial goal was to predict the exact outcome of the Fortune 500, we quickly realised this was not feasible.

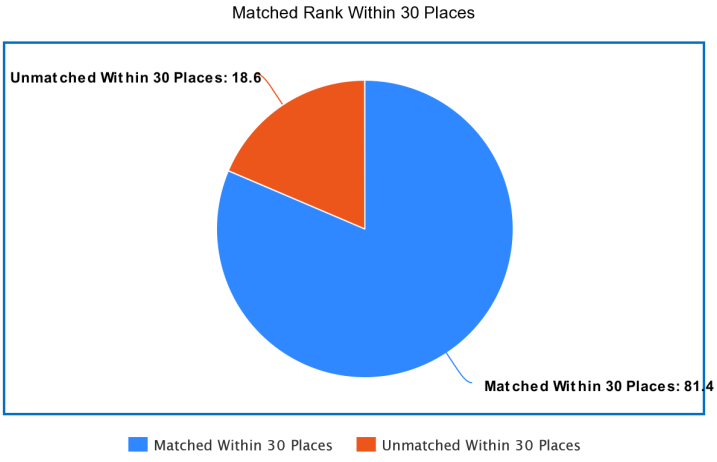
5.2 Initial Entire Data Analysis

We then decided to look at our entire dataset as a whole. We approached this is a similar fashion by first testing how many exact ranks we got correct and then broadening our search inputs. We found we correctly predicted around 2.8% perfectly and up to 81.4% within 60 places (30 places above and 30 places below).

```
Perfectly Matched:
0.027989821882951654
```

```
Within 60:
0.8142493638676844
```





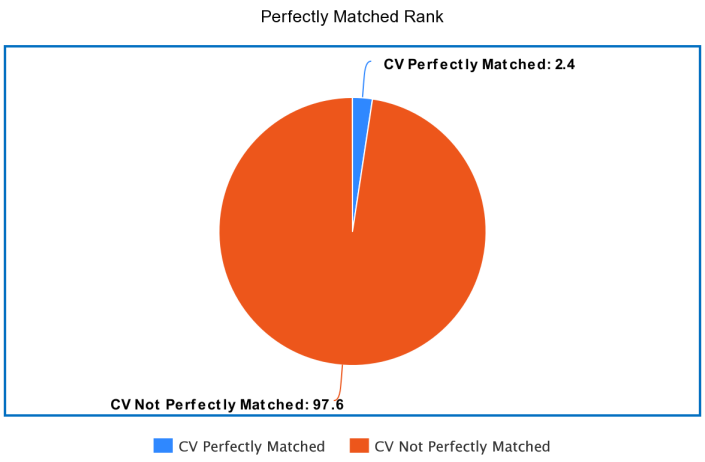
5.3 K-Fold Cross Validation for Linear Regression

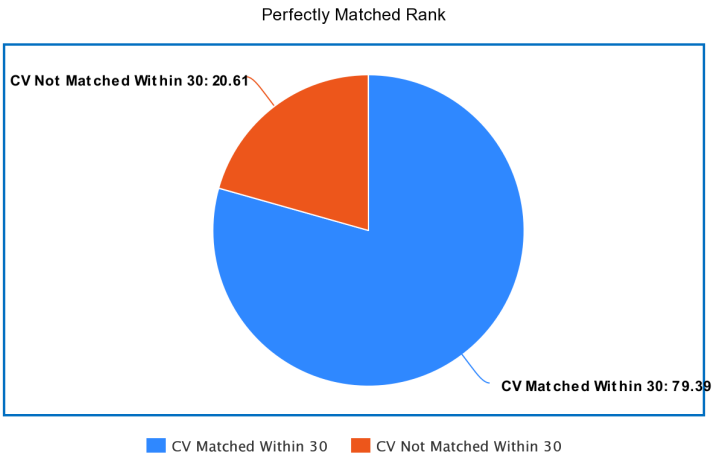
We decided to use a type similar to k-fold validation for linear regression to test our data accuracy. This involved shuffling our dataset randomly, then taking a small section of this dataset and testing it against our results. We decided to take only 20% of our dataset into consideration.

We first used this model to predict the exact ranking of the Fortune 500, we found that we were exactly correct on a companies rank 2.32% of times. We also implemented the search on the same variation as we did in the initial search at 30 above and below this left us with a result of 79.4%

```
Perfectly Matched:
0.0232470161
```

```
Within 60:
0.7938931297709924
```

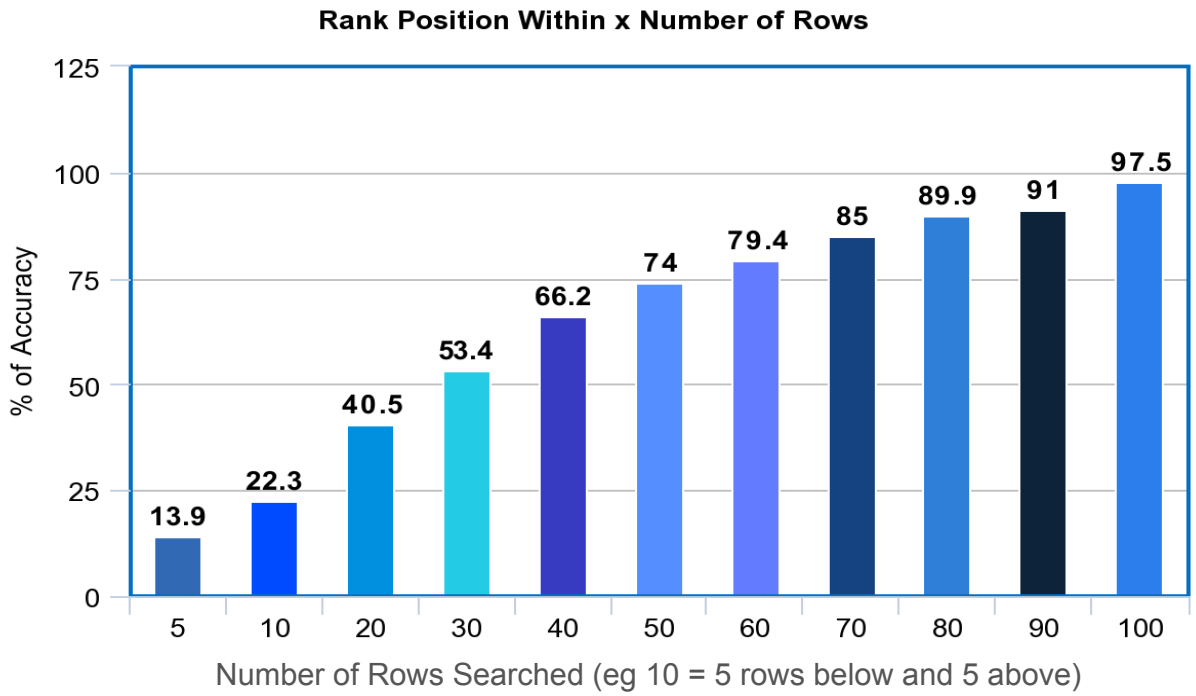




From this data we can see it is extremely hard to predict the outcome of the Fortune 500 to an exact number. However, using the Cross Validation for Linear Regression to test our data was a good way to see how consistent our outcomes were.

By taking a small percentage of the entire dataset to test with rather than the entire thing we gain a better understanding of how accurate our dataset was as a whole as the possibility of getting things like outliers is slightly higher meaning less accurate results as was the case.

After seeing this results we decided to do the same test but this time recording each instance and increasing the number of possible rows to searches each time.



From these results we see a clearer picture of how difficult it is to predict future Fortune 500 lists. We only receive a 74% accuracy of finding the companies rank when searching rows in a size 0f 50 (25 above / 25 below)

We believe we found it difficult to find rankings as there is a constant change every year due to many different aspects such as: Company goes Bankrupt, Company goes into Liquidation, Company is sold to a competitor or Company gains a large investment.

```
df1 = pd.read_csv('dataset_prediction.csv')

df2 = pd.read_csv('2019.csv')

df1['Name2'] = df2['Name'] #add the Price2 column from df2 to df1
df1['Rank2'] = df2['Rank']

for index, row in df1.iterrows():
    count = count + 1

    if row['Name'] == row['Name2']:
        true1 = true1 + 1

    for index2, row2 in df1.iterrows():
        if row['Name'] == row2['Name2'] and row2['Rank2'] > row['Rank'] - 5 and row2['Rank2'] < row['Rank'] + 5:
            true2 = true2 + 1
            break
```

An example of our code that is reading in from our 2019 prediction file and the actual results from 2019. It is then iterating through the rows testing the 5 rows above and 5 rows below it.

After rank we then tested this method on several other factors such as Revenue, Company Profits and Market Value.

For Revenue we rounded the attribute up to each 1,000 and found that we received an accuracy of about 24.68%,

```
Perfectly Matched Revenue:
0.24681933842
```

For Company Profits we rounded the attribute up to each 1,000 and found that we received an accuracy of about 38.27%,

```
Perfectly Matched Profits:
0.38273618327
```

For Company Profits we rounded the attribute up to each 1,000 and found that we received an accuracy of about 21.36%,

```
Perfectly Matched Market Value:
0.21361845262
```

5.4 Coefficient of determination for linear regression

Coefficient of determination for linear regression, also known as R-Squared is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable and ranks this on a scale from 0 to 1. In other words, the coefficient of determination tells one how well the data fits the model otherwise known as the “goodness” of fit.

$$\textbf{Coefficient of Determination} = r^2$$

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{\left[n\Sigma x^2 - (\Sigma x)^2\right]\left[n\Sigma y^2 - (\Sigma y)^2\right]}}$$

However, it does not disclose information about the causation relationship between the independent and dependent variables as well as it does not indicate the correctness of the regression model. This means although it represents data accuracy well it should

not be taken as the sole provider of information when trying to make informed decisions.

```
df = pd.read_csv('r2Score.csv')

print("Coefficient of determination:")

print(r2_score(df['Rank P'], df['Rank 2019']))
```

To make this calculation we also used sklearns function r_score(). We used it to show our fit of data from our “2019 Prediction” and the “Actual 2019”. This showed our determination to be 0.919 which is a very high score.

```
Coefficient of determination:
0.9190529701243848
```

6. Conclusion

In conclusion we found that it was very difficult to predict the exact ranking of every Fortune 500 company accurately. However using less strict constraints it it easier to predict a general sense of the table.

We believe this to be expected, after researching this there are a number of different variants that change the list every year, examples of this include: companies going into liquidation, companies getting sold to competitors, companies getting large investments and companies going bankrupt.

In future we would like to take more samples of data to make our prediction, given we were limited back to 2014 we believe this made our prediction less accurate than if we had more years to base our answer off of. We would also like to use more tests to test how accurate we were.

Our analysis of data was a particularly enjoyable part of our assignment, seeing the data we predicted and testing how accurate it was was very satisfying. We thoroughly enjoyed every aspect of this project as it taught us every stage of data mining and data analysis.