Stock Data Analysis

By: Sasha Roberts, Troy Zhongyi Zhang, Parnian Rao, Yuyang Zhang

Business Problem

The purpose of our project is to look at the historical information of the S&P 500 companies, and build models to:

- ➔ **Predict profitability (classification)**
- ➔ **Predict bankruptcy (classification & regression)**

Dataset

1. **S&P 500 companies historical prices with fundamental data**
    a. **prices-split-adjusted.csv: raw, as-is daily prices (2010-2016) with adjustments for splits**
    b. **securities.csv: general description of each company with division on sectors**
    c. **fundamentals.csv: metrics extracted from annual SEC 10K filings (2012-2016)**
2. **Bankruptcy forecast data**
    a. **bankruptcy_Train.csv - the training set with both predictors and response variable**

Data Preparation and Feature Engineering

01 Merge datasets

- Merged fundamentals with Split Prices on date column
- Merged securities on the ticker symbol

02 Missing Values

- Shares Outstanding with zeroes
- Year extracted from date
- Used KNN for Cash, Quick and Current Ratios

03 Categorization

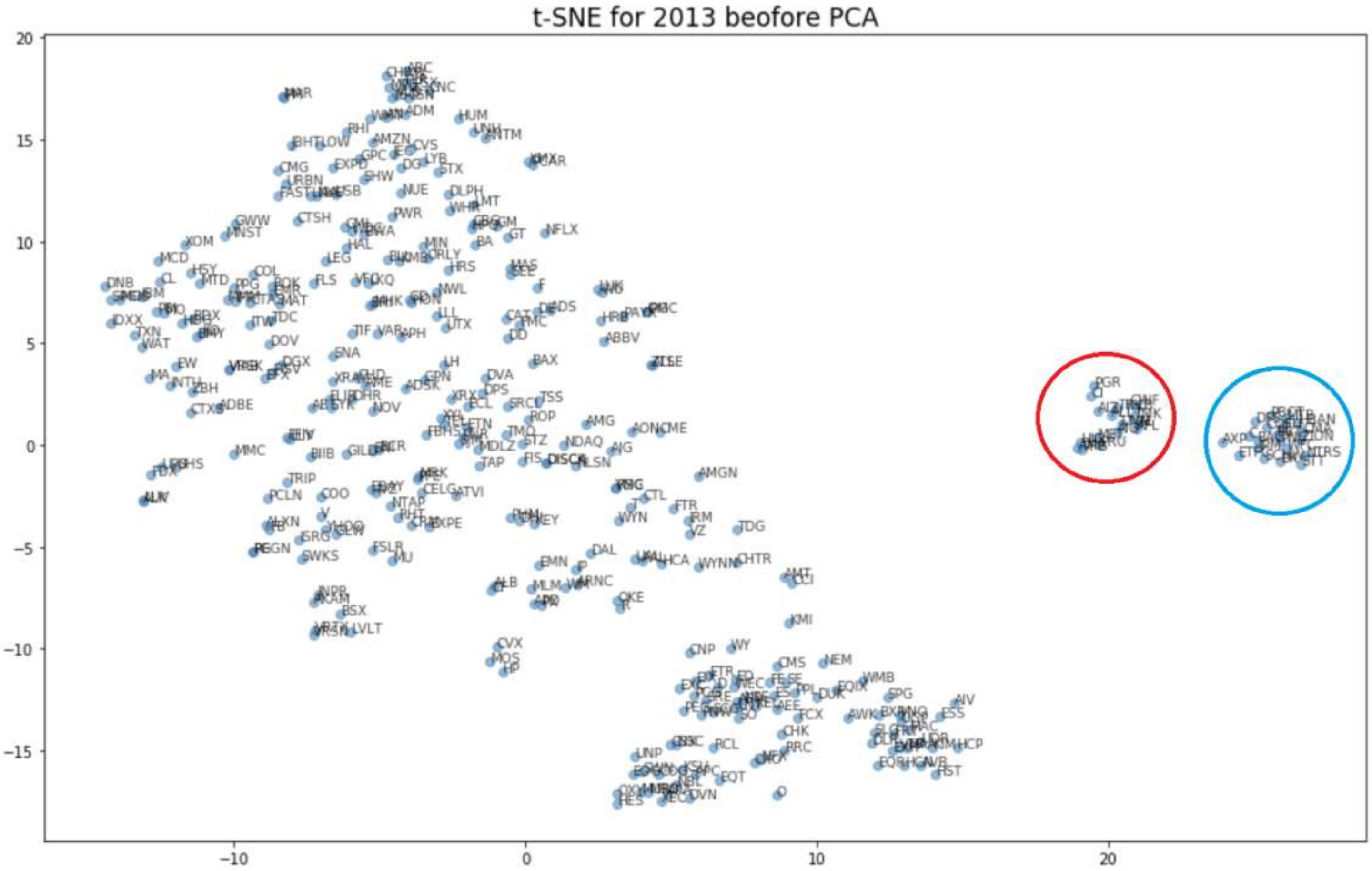
- GICS sector
- GICS Sub Industry

04 Features

- Removed columns not adding value:Headquarters, SEC Filing
- Added three new features: PE Ratio, Trend and Z-Score

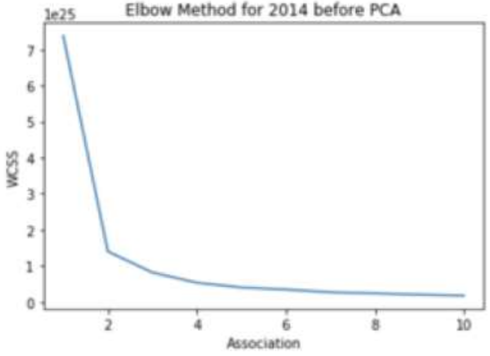
t-SNE and PCA Preview

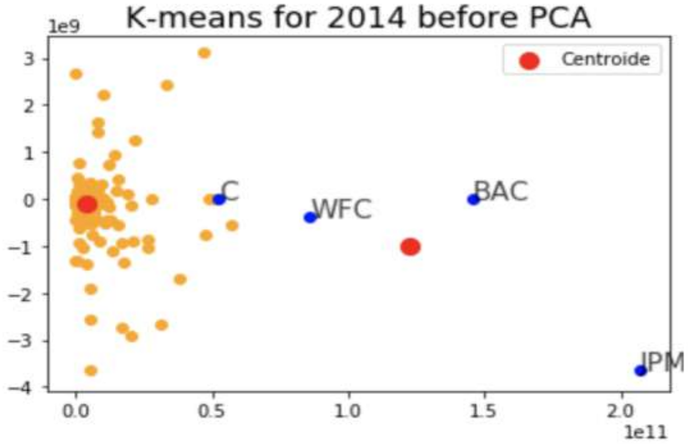
- **Clustering**
- **86 columns** – “Symbol, Date, Year/For Year, trend” 4 columns = **82 columns**
- Flexibility; effective and reasonable dimension reduction; tricky to interpret.
- **t-SNE**
- Blue - **Financial Services**
- Red - **Insurance**
- Internal relationship and similarities
- **PCA** - Select top twenty highest variations features; adding up to 83.5% - “reduced”

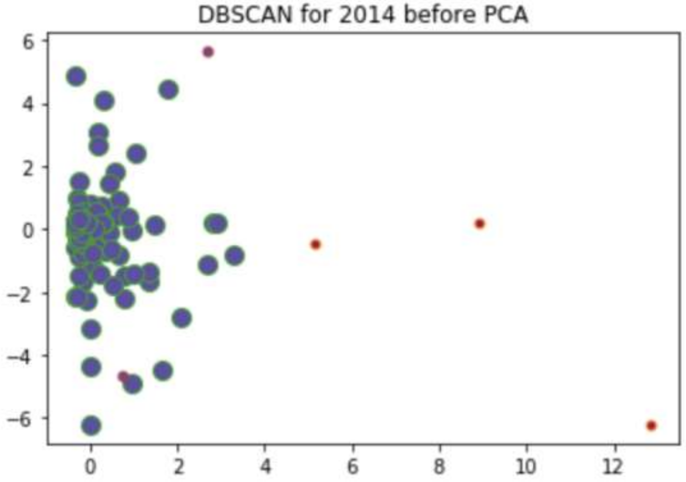


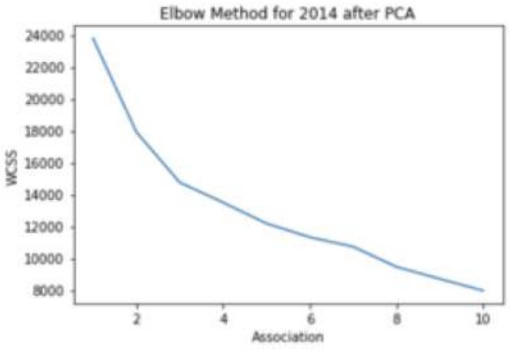
Clustering

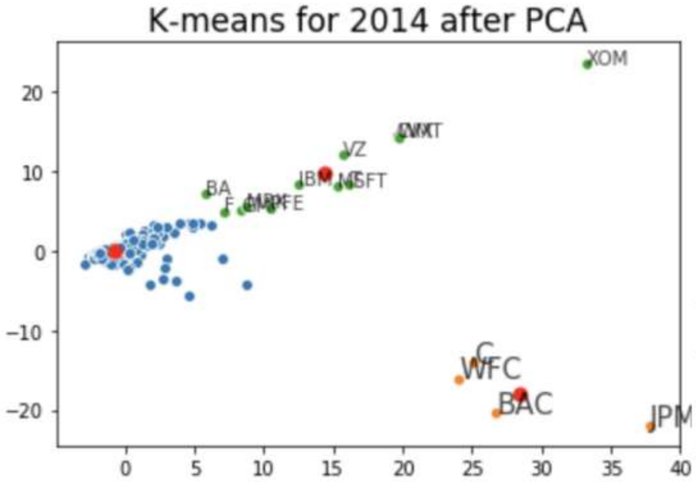
Citibank, Wells Fargo, BOA, and JP Morgan

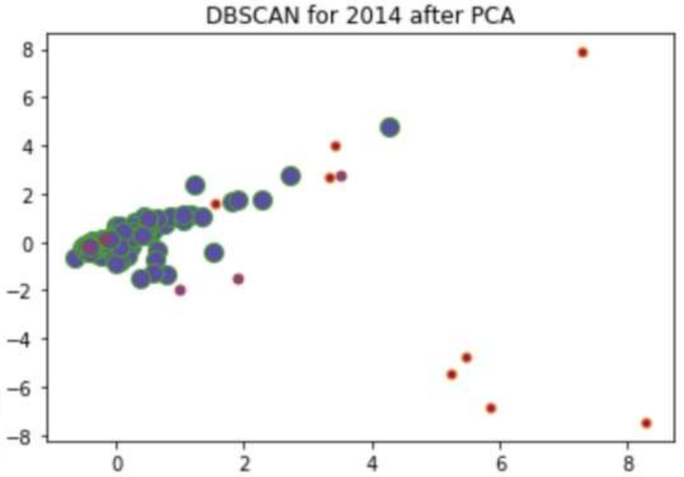




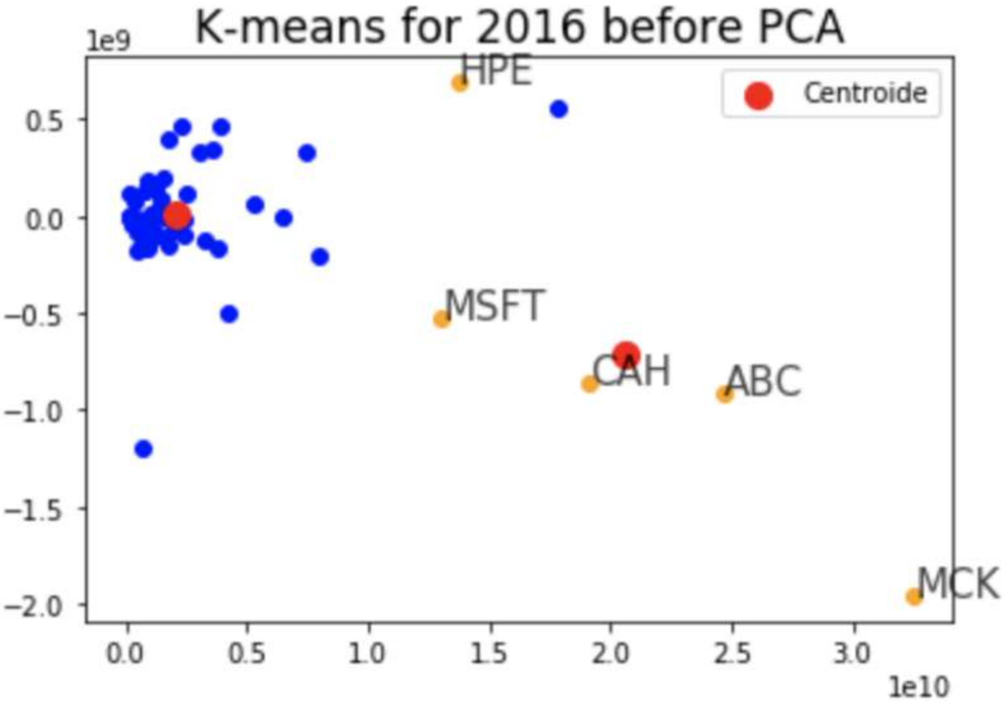


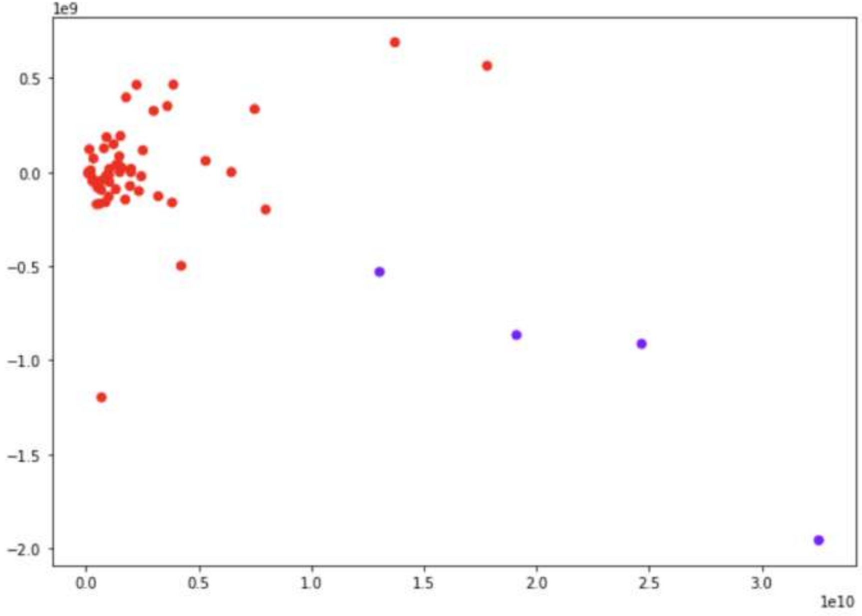






Discussion - Ward's method for HC

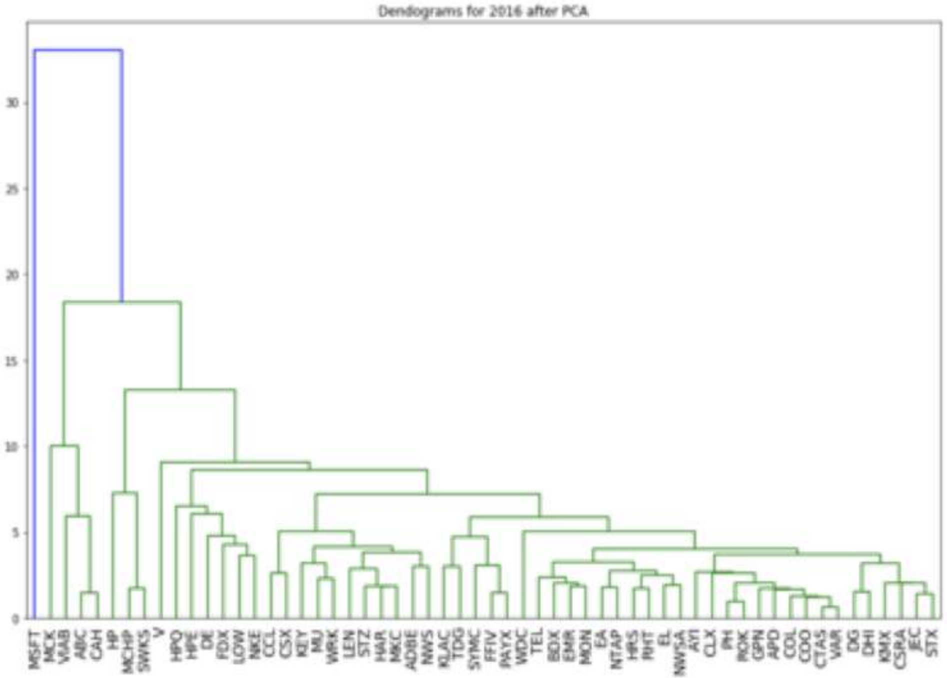


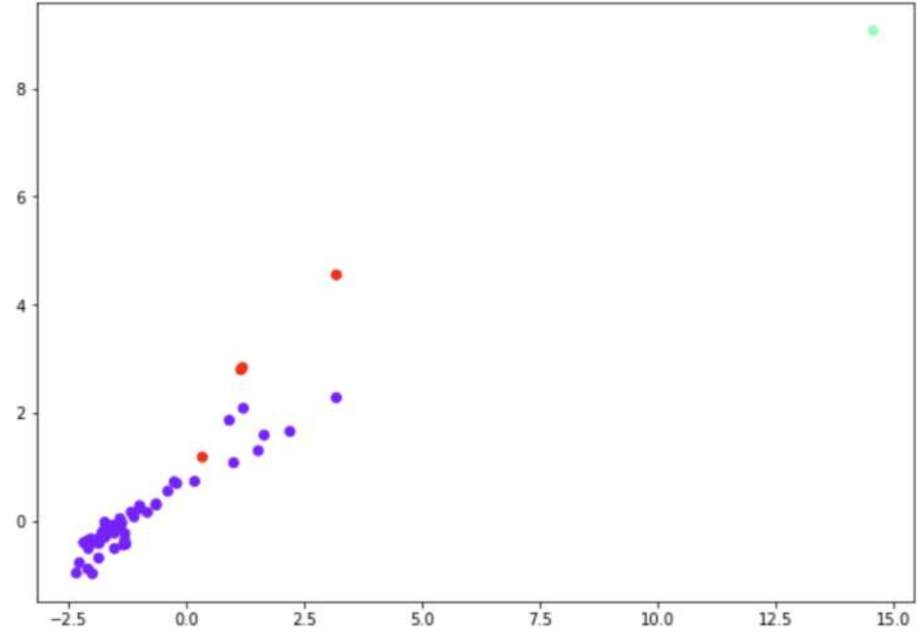


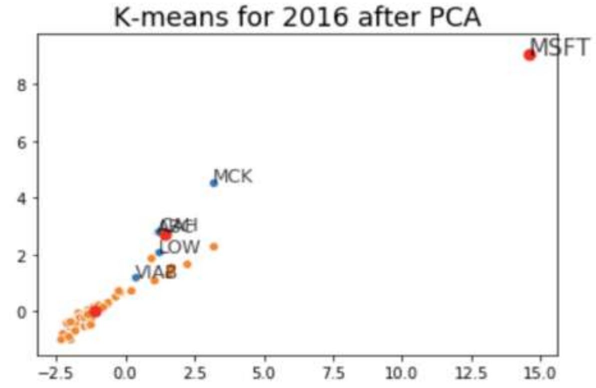
Hierarchical Clustering

Compared and obtained information from K-means

MSFT - Microsoft Corp.







Profitability Classification

Y-variable: **Trend**

PE is the ratio of the company's share price to the company's earnings per share (EPS).

- ➔ Trend is “1” when PE positive
- ➔ Trend is “0” when PE negative

Steps:

- Drop columns directly related to Y-variable (PE and Earnings Per Share)
- Attempt modeling with and without PCA
- Generate more samples with bootstrapping, KDE, and SMOTE to address overfitting

Algorithms Used

1. Logistic Regression
2. K-Nearest Neighbors
3. Gaussian Naive Bayes
4. Support Vector Classifier (SVC)
5. Decision Trees
6. AdaBoost
7. Gradient Tree Boosting
8. Random Forest
9. Stochastic Gradient Descent (SGD)
10. Perceptron (single layer)

Profitability Classification

Most model scores were lower with PCA than without

➔ Why? PCA will treat a feature that has large variance as important, but a feature with large variance can have nothing to do with the prediction target.

Without PCA, ensemble methods performed the best

➔ Why? PCA accuracy gains will almost always be minimal because an ensemble model already deals well with correlated predictors and high dimensional data sets.

SMOTE produced the best Random Forest model

➔ Why? There were few “0s” in Trend. Machine learning algorithms have trouble learning when one class dominates the other. SMOTE added more “0”s to learn from.

Precision = True Positive/(True Positive + False Positive)
Recall = True Positive/(True Positive + False Negative)
Accuracy = True Positive + True Negative/Total

Without PCA

Score	Model	F1
99.51	Decision Trees	1.00
99.51	AdaBoost	1.00
99.26	Random Forest	1.00
97.79	Gradient Tree Boosting	0.99
90.93	Logistic Regression	0.95
90.93	K-Nearest Neighbors	0.95
89.71	Support Vector Classifier (SVC)	0.95
89.71	Perceptron	0.95
88.24	Stochastic Gradient Descent (SGD)	0.93
23.77	Gaussian Naive Bayes	0.27

With PCA

Score	Model	F1
91.42	Stochastic Gradient Descent (SGD)	0.95
91.18	Logistic Regression	0.95
90.44	Random Forest	0.95
89.95	K-Nearest Neighbors	0.95
89.95	Support Vector Classifier (SVC)	0.95
89.71	Decision Trees	0.95
87.99	Gradient Tree Boosting	0.93
87.99	Perceptron	0.93
83.58	AdaBoost	0.91
32.60	Gaussian Naive Bayes	0.42

Classification Matrix for Random Forest w/ SMOTE

```
ROC Train Accuracy: 0.96 | ROC Train Error: 0.04
ROC Test Accuracy: 0.99 | ROC Test Error: 0.01
OVERFIT: False
UNDERFIT: True
              precision    recall  f1-score   support

         0.0       0.95      0.98      0.96        42
         1.0       1.00      0.99      1.00       366

   micro avg       0.99      0.99      0.99       408
   macro avg       0.98      0.99      0.98       408
weighted avg       0.99      0.99      0.99       408

F1: 0.9958960328317372
```

Bankruptcy Regression - Z Score

Z-Score<1.8

- Company is likely headed for Bankruptcy

1.8<Z-Score<3.0

- Company is in the caution zone

Z-Score>3.0

- Company not likely to go bankrupt

Z-Score = 1.2A + 1.4B + 3.3C + 0.6D + 1.0E

A = working capital / total assets
B = retained earnings / total assets
C = earnings before interest and tax / total assets
D = market value of equity / total liabilities
E = sales / total assets

Bankruptcy Regression

Random Forest - With the current data available

Test RMSE	**2.26**
CV RMSE	**2.09**
TRAIN RMSE	**1.38**

MODEL IS OVERFITTING
(CV RMSE>Train RMSE)

Complexity — Reduce Model Complexity

Generate Data — KDE
SMOTE

USING KDE:
500 more training observations

Test RMSE	**2.64**
CV RMSE	**2.48**
TRAIN RMSE	**2.29**

Bankruptcy Regression

- Data Generation - **SMOTE**
- *Classify rare events as 1, all other as 0*
- *For our data, z-score between 1.8 and 3 was rare*
- *Oversample the rare events in dataset using SMOTE*
- *Remove the class variable and use the new training data to model*

	PE	z_score	class
	13.338085	0.734268	0
	54.638889	2.551277	1
	-37.834396	10.406381	0
	37.908629	1.031081	0
	4.304511	0.174830	0

RESULTS

- Reduced difference between CV RMSE and Train RMSE - But still overfitting

Test RMSE	**2.62**
CV RMSE	**2.24**
Train RMSE	**2.06**

Bankruptcy Regression - Results

After KDE data generation

Model	CV error
Random Forest	2.482684
Decision Tree	2.611628
Linear Regression	2.939044

After SMOTE data generation

Model	CV error
Random Forest	2.238498
Decision Tree	2.332016
Linear Regression	2.560746

All models after SMOTE data generation performed better!

Bankruptcy: Classification and Transfer Learning

Transfer learning is a machine learning technique that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

- ➔ Why? Our fundamental dataset only includes financial metrics and doesn’t include companies’ operating status, that is to say, we couldn't tell which companies were bankrupt.
- ➔ Train different classification models using an external dataset “bankruptcy_Train” which has 10,000 records.

bankruptcy_train.csv

Attr10	...	Attr56	Attr57	Attr58	Attr59	Attr60	Attr61	Attr62	Attr63	Attr64	class
0.126789	...	0.014367	0.005457	-0.014143	-0.020924	0.068399	-0.214478	-0.013915	-0.173939	-0.046788	0
0.073759	...	0.008492	-0.008385	-0.008666	-0.023095	-0.033498	-0.205796	-0.015174	-0.073056	-0.027236	0
-0.071287	...	0.010819	0.006779	-0.009437	-0.007919	-0.043455	0.019740	-0.011736	-0.291624	-0.033580	0
-0.085266	...	0.010683	0.005384	-0.010840	0.001381	-0.042828	-0.350519	0.002969	-0.554685	-0.046823	0
0.076880	...	0.010970	0.025295	-0.011056	-0.022535	-0.035892	-0.181557	-0.015623	-0.027841	-0.023694	0

Fundamental.csv

Ticker Symbol	Period Ending	Accounts Payable	Accounts Receivable	Add'l income/expense items	After Tax ROE	Capital Expenditures	Capital Surplus	Cash Ratio	...	Total Current Assets
AAL	2012-12-31	3.068000e+09	-222000000.0	-1.961000e+09	23.0	-1.888000e+09	4.695000e+09	53.0	...	7.072000e+09
AAL	2013-12-31	4.975000e+09	-93000000.0	-2.723000e+09	67.0	-3.114000e+09	1.059200e+10	75.0	...	1.432300e+10
AAL	2014-12-31	4.668000e+09	-160000000.0	-1.500000e+08	143.0	-5.311000e+09	1.513500e+10	60.0	...	1.175000e+10
AAL	2015-12-31	5.102000e+09	352000000.0	-7.080000e+08	135.0	-6.151000e+09	1.159100e+10	51.0	...	9.985000e+09
AAP	2012-12-29	2.409453e+09	-89482000.0	6.000000e+05	32.0	-2.711820e+08	5.202150e+08	23.0	...	3.184200e+09

Bankruptcy: Classification and Transfer Learning

SMOTE Technique to generate more samples in the training dataset and balance the amount of bankruptcy records and non-bankruptcy record.

```
Before OverSampling, counts of label '1': 130
Before OverSampling, counts of label '0': 6870

After OverSampling, the shape of train_X: (13740, 48)
After OverSampling, the shape of train_y: (13740,)

After OverSampling, counts of label '1': 6870
After OverSampling, counts of label '0': 6870
```

Models trained

- Logistic Regression, Decision Tree Classifier and Random Forest Classifier models
- Based on SMOTE data and No SMOTE data
- **The models were improved** a lot after SMOTE technique applied.

NO SMOTE

Model	Accuracy score	Precision	Recall
Random Forest Classifier	0.974333	0.166667	0.013699
Logistic Regression	0.973000	0.000000	0.000000
Decision Tree Classifier	0.975667	0.000000	0.000000

SMOTE

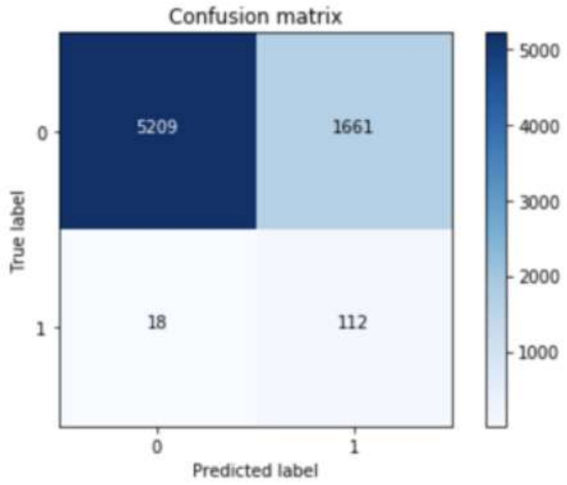
Model	Accuracy score	Precision	Recall
Logistic Regression	0.746000	0.075216	0.835616
Decision Tree Classifier	0.693333	0.054679	0.712329
Random Forest Classifier	0.967000	0.240000	0.164384

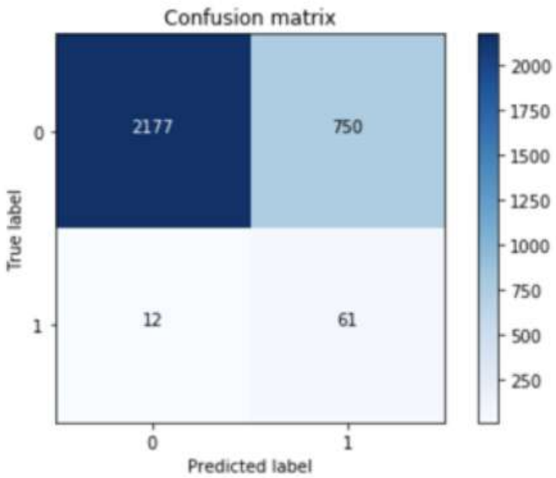
Bankruptcy: Classification and Transfer Learning

"Recall" is more important than "Precision" and "Accuracy" in this case. It is fine to include some companies which actually will not be bankruptcy (false positive) in our narrow down list. We would like to avoid missing any companies which actually will be bankruptcy(false negative).

Logistic Regression model is the best model, with the highest recall score

- 86.15% for training dataset
- 83.56% for testing dataset (right chart)





Bankruptcy: Classification and Transfer Learning

Apply the logistic regression model to the Fundamental dataset, which has **1781 records**.

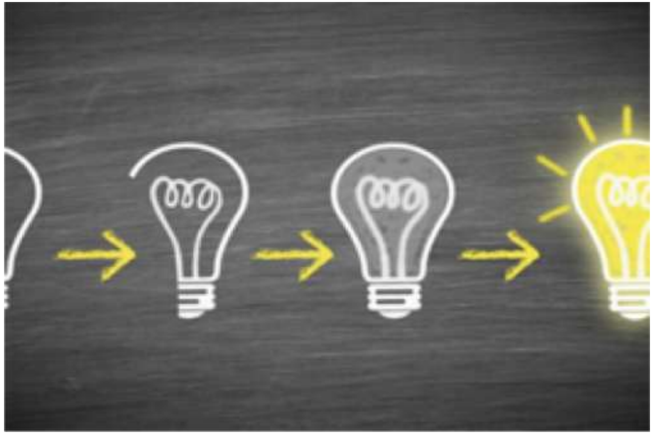
- Narrowed down a list of **21 companies** which might be bankruptcy and need further track or investigation.
- Analyzed the Z-score of those 21 companies and identified companies with low/dropping Z-scores
  - BDX (Becton Dickinson), KMX (CarMax), CVX (Chevron Corporation), MKC (McCormick & Company)

```
data[data['symbol']=='KMX']
```

Capital Expenditures	Capital Surplus	Cash Ratio	Cash and Cash Equivalents	...	Estimated Shares Outstanding	open	close	low	high	volume	GICS Sector	GICS Sub Industry	PE	z_score
-235707000.0	9.722500e+08	98.0	673651000.0	...	2.285705e+08	38.619999	38.410000	38.400002	39.080002	1393900.0	0	103	20.215789	2.599891
-310317000.0	1.038209e+09	101.0	887200000.0	...	2.239027e+08	48.630001	48.430000	48.099998	48.750000	919900.0	0	103	22.013636	2.528942
-315584000.0	1.130822e+09	38.0	381223000.0	...	2.030710e+08	47.169998	46.259998	46.250000	47.380001	2119700.0	0	103	15.068403	2.061834

Conclusion

- SMOTE worked well for all problems



- Overfitting was a major problem
  - Gather more actual data
- Improve clustering results by removing outliers

QUESTIONS