Company Substantiated Estimates of the Likelihood and Severity Assessment

using logistic regression analysis

Temi Ajulo | Capstone Project | May 6, 2022

## TABLE OF CONTENT

[ABSTRACT](#_tdu0j2rkwd2z)

[INTRODUCTION (Summary of SJM )](#_ecadkk585oul)

[OVERVIEW OF LOGISTIC REGRESSION](#_dl689448l11k)

[PROJECT PROCESS OUTLINE](#_70uggjp6gu2o)

[OBTAINING THE DATA](#_9zhwtm6jg4wg)

[DATA PREPARATION/CLEANING](#_e39pe3ke71kl)

[EXPLORATORY DATA ANALYSIS](#_jhoezl6msfih)

[Evaluation of the data set & Logistic Regression Assumption.](#_55itp8bi5xfx)

[feature importance](#_5nsi9gra6s6y)

[GOODNESS OF FIT](#_cw7yfipqokdi)

[ROC CURVE](#_mxactknyscqf)feature importance

[Severity /Impact.](#_q16z7im8oick)

## ABSTRACT

This paper will cover the usage of logistic regression as a modeling algorithm to estimate and substantiate the likelihood (probability) and the monetary impact (severity, or cost) of shareholder class action. We need to build a model that can predict whether a particular will be sued or not for Shareholder violation . We will be applying the model built to the assigned company The J. M. Smucker Company(SMJ). The Company also needs to know the factors (variables) which influence shareholder class action and how much they impact individually. This will help the company to improve those areas to reduce the risk to their organization. Companies need a decent model that can predict a good percentage of probability of litigation or not based on the industry they are under ., I will be covering my analysis and approach through different process flows in the data science pipeline.

Executive Summary

To answer the question of the likelihood (probability) and the monetary impact (severity or cost) of shareholder class action litigation. I used data science techniques and algorithms to get the most accurate estimation. followed a typical data science pipeline called “OSEMN.” We process the data from different sources: fundamentals, ratings, securities, stocks, and settlements, after Correctly planning and executing all pertinent and needed data manipulation, recoding, new variable creation, aggregation, imputation, and amalgamation steps. We made sure that the data met the logistic regression requirement. We ran a logistic regression model on python using the sklearn and scipy package to predict the probability of the SJM being involved in a shareholder class-action lawsuit. The results showed the probability/likelihood that The J. M. Smucker Co (SJM) will be involved in a lawsuit at 14.27%. Using the 50% threshold if the yes/No. We can say that the prediction is favorable to SJM of them not being sued. For the severity, we determined the weighted average using the 95% confidence interval. The settlement amount averages are +/- (or between $3,457,135.89 and $8,522,864.10). How might the average affect the company assignment? The settlement amount has little to no effect on SJM as the weight averages fall below 1% of the company market Cap. The Model accuracy was 89%. Using some goodness of fit methods, we see that The True Positive Rate is 0.99, The Precision is 0.89, The False Positive rate is 0.9, and The average False Negative Rate is 0.45. these stats tell that the model built can be trusted. In conclusion, we can say with confidence that SJM will likely not be sued.

## INTRODUCTION (Summary of SJM )

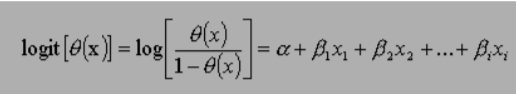
The Executive Threat Ecosystem model captures the dyadic interactions between business organizations and seven distinct stakeholder groups and considers the influence of three distinct expectation and norm forming forces. Overall, the ecosystem-based conceptualization supports a more thorough prospective assessment of the totality of threats faced by directors and managers of business organizations by expressly considering yet unrealized but theoretically plausible dangers, in addition to those whose recurrence patterns are established. This assignment's overall goal is to estimate and substantiate the likelihood (probability) and the monetary impact (severity or cost) of shareholder class action litigation. Shareholder litigation rights is an important topic as and when companies go bankrupt what would be the plight of the shareholders. For this project, we will be attempting the probability and the monetary impact that The J. M. Smucker Company will have. JM Smucker Company manufactures, and markets branded food and beverage products worldwide. It operates in four segments: US Retail Coffee, US Retail Consumer Foods, US Retail Pet Foods, and International and Away from Home. The company offers main-roasted, ground, single-serve, and premium coffee; peanut butter and specialties; fruit pastes, fats and oils, and frozen sandwiches; pet food and pet snacks; and catering hot drinks, catering portion control, and flour products, as well as dog and cat food, frozen hand products, juices and beverages, and bakery items. Folgers, Café Bustelo, Dunkin' Donuts, 1850, Jif, Smucker's, Crisco, Smucker's Uncrustables, Meow Blend, Kibbles 'n Bits, 9Lives, Nature's Recipe, Milk-Bone, PupPeroni, Rachael Ray Nutrish, Natural Balance, Robin Hood and Five Rose brands. It is traded with the code SJM'. JM Smucker Company is closely involved in actively managing the risks the company faces, including securing economically rationalized insurance coverage to protect its directors and officers from potential shareholder lawsuits. The Smucker Company was founded in 1897 and is headquartered in Orrville, Ohio. The company is currently traded on the New York Stock Exchange under the code 'SJM.' JM Smucker Company is closely involved in actively managing the risks the company faces, including securing economically rationalized insurance coverage to protect its directors and officers from potential shareholder lawsuits. Is closely involved in actively managing the risks the company faces, including securing economically rationalized insurance coverage to protect its directors and officers from potential shareholder lawsuits. JM Smucker Company is closely involved in actively managing the risks the company faces, including securing economically rationalized insurance coverage to protect its directors and officers from potential shareholder lawsuits. It is closely interested in actively managing the risks it faces. The Smucker Company was founded in 1897 and is headquartered in Orrville, Ohio. The company is currently traded on the New York Stock Exchange under 'SJM.'.

## OVERVIEW OF LOGISTIC REGRESSION

Logistic Regression commonly deals with the issue of how likely an observation is to belong to each group. This model is commonly used to predict the likelihood of an event occurring. In contrast to linear Regression, the output of Logistic Regression is transformed with a logit function. This makes the output either 0 or 1. This is a valuable model for this problem because we are interested in predicting whether a company will be sued (1) or not sued (0).

Another reason why logistic Regression is the preferred model of choice is because of its interpretability. Logistic Regression predicts the outcome of the response variable (Shareholder lawsuit) through a set of other explanatory variables, also called predictors. In the context of this domain, the value of our response variable is categorized into two forms: 0 (zero) or 1 (one). The value of 0 (zero) represents the probability of an employee not leaving the company, and the value of 1 (one) represents the probability of an employee leaving the company.

Logistic Regression models the probability of ‘success’ as:



The equation above shows the relationship between the dependent variable (success), denoted as (θ) and independent variables or predictor of event, denoted as xi. Where α is the constant of the equation and, β is the coefficient of the predictor variables.The equation above shows the relationship between, the dependent variable (success), denoted as (θ) and independent variables or predictor of event, denoted as xi. Where α is the constant of the equation and, β is the coefficient of the predictor variables.

## PROJECT PROCESS OUTLINE

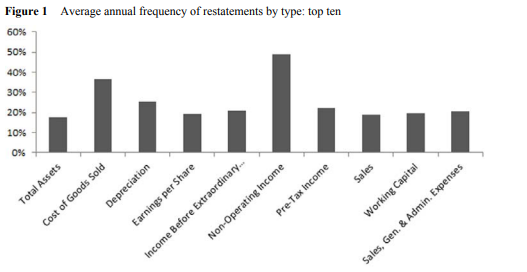
* + Obtaining the data is the first approach in solving the problem.
  + Scrubbing or cleaning the data is the next step. This includes data imputation of missing or invalid data and fixing column names. Feature engineering
  + Exploring the data will follow right after and allow further insight of what our dataset contains. Looking for any outliers or weird data.
  + Modeling the data will give us our predictive power on whether an employee will leave.
  + Interpreting the data is last. With all the results and analysis of the data, what conclusion is made? What factors contributed most to employee turnover? What relationship of variables was found?

## OBTAINING THE DATA

The data used for the project was found from different sources, which range from SEC-10k of different companies to SCA Filings and Settlements all in the various sectors of the economy.As such there are five different datasets which would merge and manipulated to make them ready for analysis. The dataset include, fundamental, Ratings, Securities, Stocks and SCA filings.The dataset provides various data points that describe the status of companies , which means that the features and observations used are all made up to real-world values that show the companies health .The number of observations given from from each of the dataset is as follows: fundamental (61429, 1768),Securities (32382, 55),Ratings(123679,21), Stocks (3645376, 76)and SCA filings (1892, 6)I will be using Python as the programming language for the analysis.

## DATA PREPARATION/CLEANING

The first step taken was to read the files into Python using the Read CSV function; after that the datasets checked to identify null or missing values. Based on the result obtained from algorithm.on python. The strategy used for handling missing data sets was to delete all empty columns from the data frame and so I eliminate all columns with more than 20% missing. All over columns within the 20% margin were replaced were the overall mean of the column.For the purpose of this project and reduce the dimensions to something more manageable we filtered the data by what sector of the assigned company.SJM fell in sector 30 which represents Consumer Staples: this sector covers the Food & Drug Retailing industry. After this ,the data is further filtered down by removing the rows with SUMM\_STD to eliminate duplicate values in the dataset. Lastly the data sets were grouped by and aggregated using the unique value of the “GV key and Tic” this column acts as a unique identifier of each company. Now that each dataset has been compressed to make sure all companies are presented in one row only. The fundamental data frame becoming the body for the new final dataset was reviewed to find what feature might be useful for determining the question of likelihood and estimation. The rationale for identifying the features were based on the readings assigned to understand the company and the litigation system. 44 features were selected from the fundamentals,2 from the Stocks, 2 from the Securities ,1 from Ratings and 2 from the SCA Filings. Two features in stock( Price -High – Daily and Price - Low - Daily were subtracted from each other to show trend if increasing or decreasing . the dataset were magered together, in order to reduce the chances of the model overfitting we standardize the data set. Which normalized all continuous variables by placing them between them 0 and 1. In addition, categorical variables in the data set like S&P Domestic Long Term Issuer Credit Rating In the ratings dataset categorical variables were assigned numeric values. For SCA data case statements were used to create a new column to identify if the companies had been sued or not.



Here is the list of features for the final dataset.

· gvkey- Global Company Key-

· dvpsx\_f -Dividends per Share - Ex-Date - Fiscal

· wcap Working Capital (Balance Sheet)

· xint Interest and Related Expense - Total

· xsga Selling, General and Administrative Expense

· cshtr\_c Common Shares Traded - Annual - Calendar

· dvpsp\_c Dividends per Share - Pay Date - Calendar

· dvpsx\_c Dividends per Share - Ex-Date - Calendar

· prch\_c Price High - Annual - Calendar

· adjex\_c Cumulative Adjustment Factor by Ex-Date - Calendar

· cshtr\_f Common Shares Traded - Annual - Fiscal

· dvpsp\_f Dividends per Share - Pay Date - Fiscal

· mkvalt Market Value - Total - Fiscal

· teq Stockholders Equity - Total

· prcc\_f Price Close - Annual - Fiscal

· prch\_f Price High - Annual - Fiscal

· prcl\_f Price Low - Annual - Fiscal

· ggroup GIC Groups

· gind GIC Industries

· gsector GIC Sectors

· gsubind GIC Sub-Industries

· sic Standard Industry Classification Code

· state State/Province

· txt Income Taxes - Total

· seq Stockholders Equity - Parent

· epsfi Earnings Per Share (Diluted) - Including Extraordinary Items

· tic Ticker Symbol

· at Assets - Total

· capx Capital Expenditures

· cogs Cost of Goods Sold

· cshfd Common Shares Used to Calc Earnings Per Share - Fully Diluted

· cshpri Common Shares Used to Calculate Earnings Per Share - Basic

· dltt Long-Term Debt - Total

· dp Depreciation and Amortization

· epsfx Earnings Per Share (Diluted) - Excluding Extraordinary Items

· sale Sales/Turnover (Net)

· epspi Earnings Per Share (Basic) - Including Extraordinary Items

· epspx Earnings Per Share (Basic) - Excluding Extraordinary Items

· ib Income Before Extraordinary Items

· ibmii Income before Extraordinary Items and Noncontrolling Interests

· mii Noncontrolling Interest (Income Account)

· ni Net Income (Loss)

· nopi Nonoperating Income (Expense)

· pi Pretax Income

· ppent Property, Plant and Equipment - Total (Net)

· reuna Retained Earnings - Unadjusted

· stko Stock Ownership Code

· trfm Monthly Total Return Factor

· trt1m Monthly Total Return

· SettlementAmount -SettlementAmount

· Sued Sued

· tic Ticker Symbol

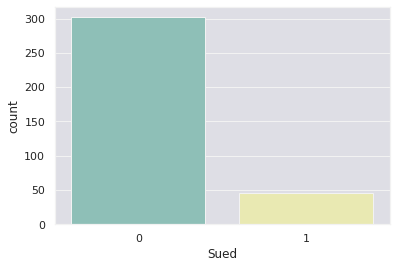
· cshoc Shares Outstanding

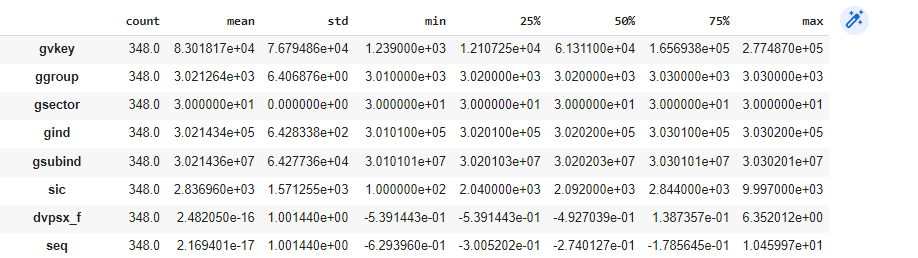
## EXPLORATORY DATA ANALYSIS

**Statistical Overview**

Here are some important numbers to keep in mind of the dataset:

* There are 348 entries of companies for the sector 30 and 49 features
* 47 independent variables
* 2 target variables.
* The Sued (target Variables show a much higher volume of companies not being sued )



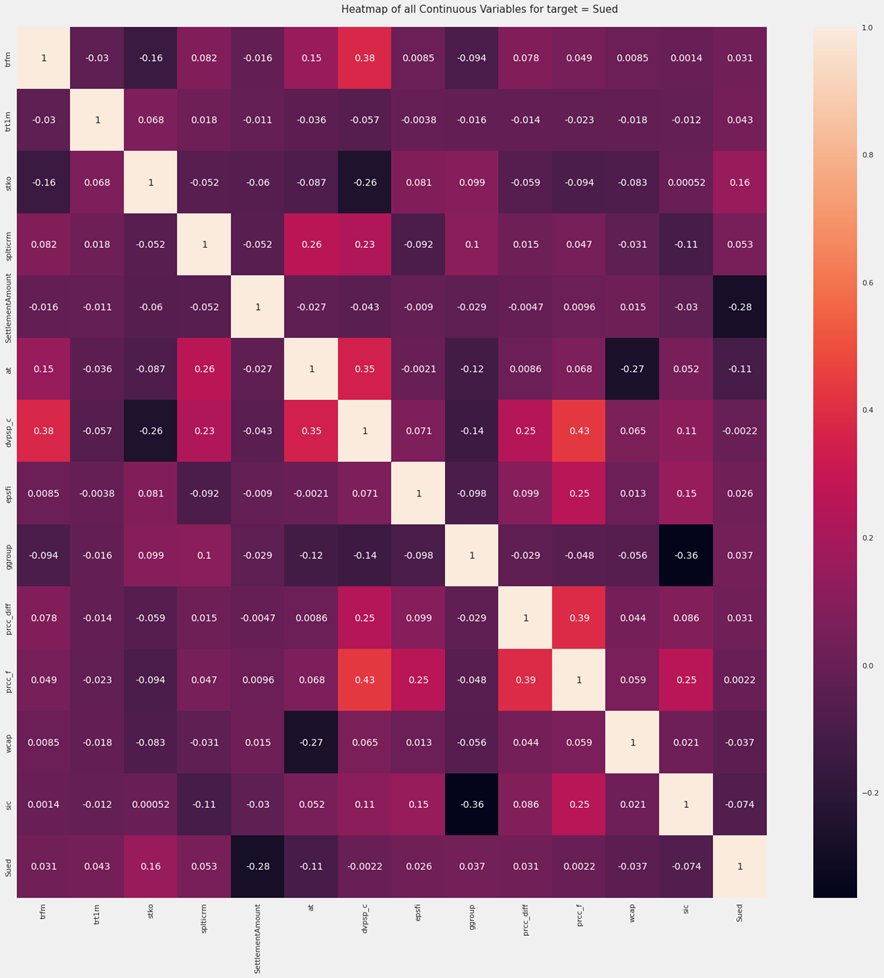


Correlation Matrix & Heatmap

There is a **positive (+)** correlation between the variables: **Sued** and Rest of the continuous variable on the date from the heatmap. This means that the companies with similar financial status will affect the significance of the predictors, as the features are saying the same thing and do not add more value to the model . For the **negative (-)** relationships, the most important feature that correlated with our target variable (Sued) is Settlement Amount. This should support our initial intuition that companies in the sector don't usually get SCA lawsuits .

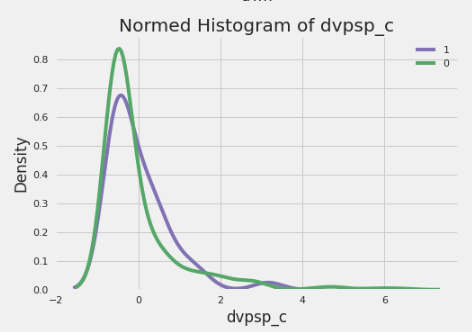
**Questions to Think About:**

* What features affect our target variable (Sued ) the most?
* What features have strong correlations with each other?
* Can we do a more in-depth examination of these features?

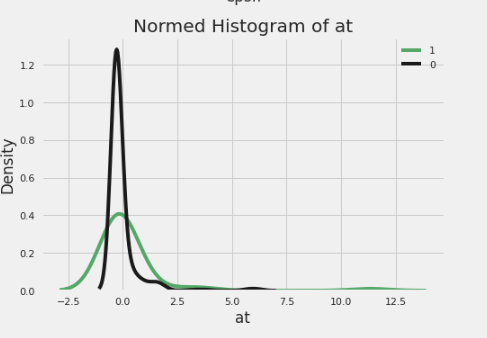


A few features that showed up as highly collated with one another . are show below.{'dltt','mkvalt','nopi','pi','ppent','prch\_c','prcl\_f','seq','txt','xsga'}.

Histogram (Used / Dividends per Share/ Assets - Totals/)

Dividends per Share: the display shows distributions can have two peaks. i notice that Dividends per Share is left skewed when measured in regards to whether the companies are sued or not. This also indicates that the mean is less than the median.

**Assets Totals:** the total assets of the data is similar to the dividends variable and also trend left skewed when measured in regards to whether the companies are sued or not. However the peaks for total show large difference as the companies who are sued are much less than those who are not when comparing both features to each other.



**Questions to Think About:**

* Is there a reason for these distributions on these graphs?
* Could employees be grouped distinctively with these features

## Evaluation of the data set & Logistic Regression Assumption.

After reviewing all the features of the finaldata set and identifying any outliers or oddies that would affect the model . I made sure to check if the data met the requirement that logistic regression assumptions .

* **First Assumption - Appropriate Outcome Type:**

logistic regression assumes that the outcome variable is **binary**,where the number of outcomes is two (e.g., Yes/No). We covered that by making sure that the target feature “sued” met that requirement.

* **Second Assumption - Linearity of independent variables and log-odds-**

One of the critical assumptions of logistic regression is that the relationship between the logit (aka log-odds) of the outcome and each continuous independent variable is linear. Which means we need to make sure that independent features are statistically significant. We will be handling this by running feature importance and using OLS results.

* **Assumption 3— No strongly influential outliers**

Logistic regression assumes that there are no highly influential outlier data points, as they distort the outcome and accuracy of the model. By standing the dataset earlier we were able to make sure that we met this requirement.

* **Assumption 4 — Absence of Multicollinearity**

Multicollinearity corresponds to a situation where the data contain highly correlated independent variables. This is a problem because it reduces the precision of the estimated coefficients, which weakens the statistical power of the logistic regression model. By running a correlation matrix Heatmap. We met the assumption.

* **Assumption 5— Independence of observations**

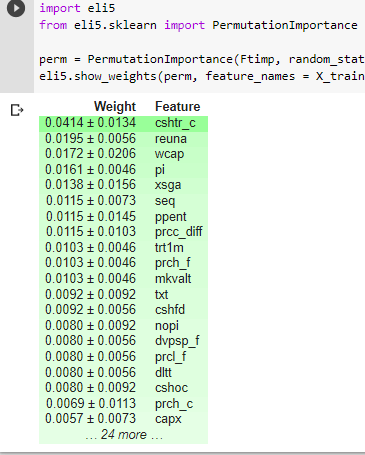
The observations must be independent of each other, i.e., they should not come from repeated or paired data. This means that each observation is not influenced by or related to the rest of the observations. That is why I made sure the feature selected from each of the separate data was not one that exists already in the other. I also made sure that I selected features that told a different aspect of the financial health for the companies.

* **Assumption 6 —** Sufficiently large sample size

There should be an adequate number of observations for each independent variable in the dataset to avoid creating an overfit model. Although the data size has considerably shrunk due to the aggregations. There is still a sizable amount to run the model.

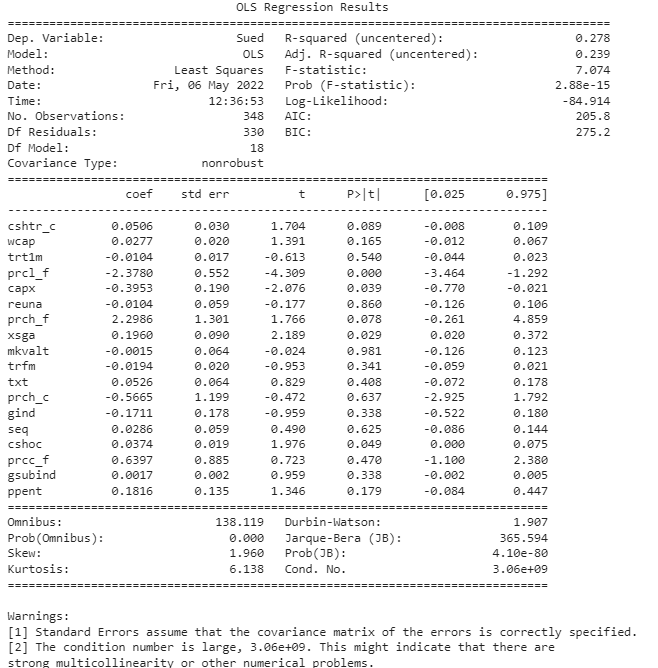
## feature importance

By using a RandomForest classifier and Permutation feature importance, it could rank the features used for the prediction. Permutation feature importance measures the increase in the prediction error of the model after we permuted the feature’s values, which breaks the relationship between the feature and the true outcome.The top three features were Common Shares Traded - Annual - Calendar, yearsAtCompany, Working Capital (Balance Sheet) and Retained Earnings - Unadjusted . This is helpful in creating our model for logistic regression because it’ll be more interpretable to understand what goes into our model when we utilize less features. By using the list below I was able to reduce the feature by only selecting the first 19 features on my first model run.



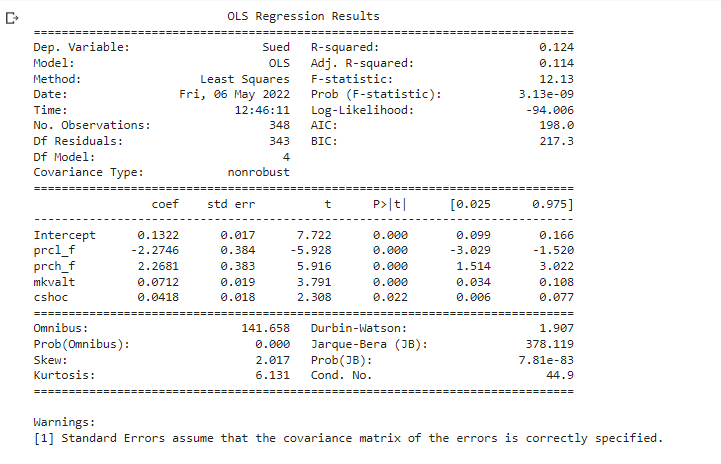
**Important Features:** To meet the requirement of having only statistically significant features we ran Ordinary Least Squares(OLS) to determine through the p values what features were equal or less 0.05 .This confidence interval approach is one of them. 5% is the standard significance level (∝) at which C.I’s are made.

C.I for B1 is **( b1 – t∝/2 s.e(b1) , b1 + t∝/2 s.e(b1) )**

****

Looking at the p value on the summary we can see that about six features are statistically significant as such I would remove and add several feature mixtures with one another to make only significant ones remain. This is because for features added or removed the p value changes.

After running the model several times with different version of feature and taken into account the highly correlated features and feature importance i was able to find four that were significant

****

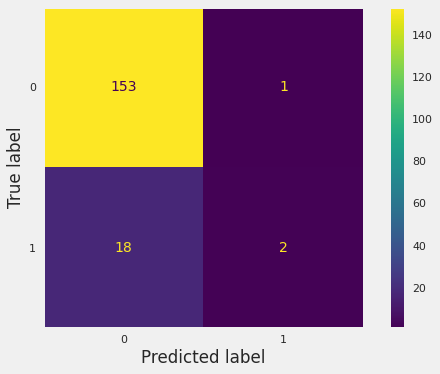
## goodness of fit

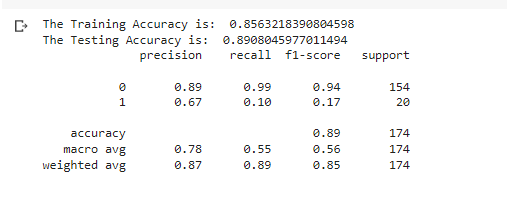
After determining what features fit, I ran the logistic regression model on sklearn. This was run by splitting the data into train and test at 50% due the fact the data as few companies that have sued in comparison to the overall data set. I also made sure to split the data due to overfitting because the sample size is not that large.

I'll use goodness-of-fit tests to see if the projected probability differs from the observed probabilities in ways that the binomial distribution cannot account for. If the goodness-of-fit test p-value is less than the set significance level (0.05),

**Confusion matri**x: Starting with the No row of the table, we can see that 153 companies were not sued and were accurately predicted not to have been sued, whereas only one company was not sued and was correctly anticipated to have been sued. a percentage indicating that the model correctly predicted non-sued for 89% of those who did not get sued. So far, everything has gone well.

Look at the second row now. It indicates that among those who were sued, the model was better at predicting whether they would be sued than those who did not (i.e., 18 versus 2). So, only 67 percent of the time, the algorithm properly predicts that company who will sue.

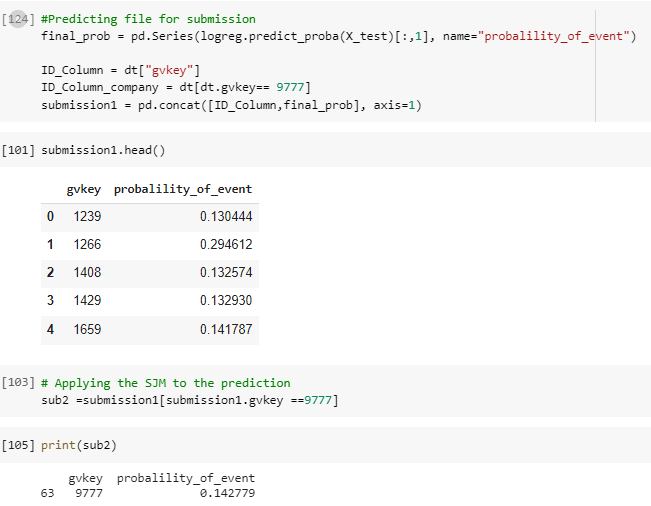
****

The model shows that the accuracy is 89% . 

F1 score can also be described as the harmonic mean or weighted average of precision and recall. As such, the weight average of the model shows us at 0.85.

Applying the model to SJM

After finding that the model is performing well I plugged in my assigned company to find the prediction of the probability of SJM becoming involved in an SCA lawsuit . Using the “gvkey” I was able to narrow down the result.

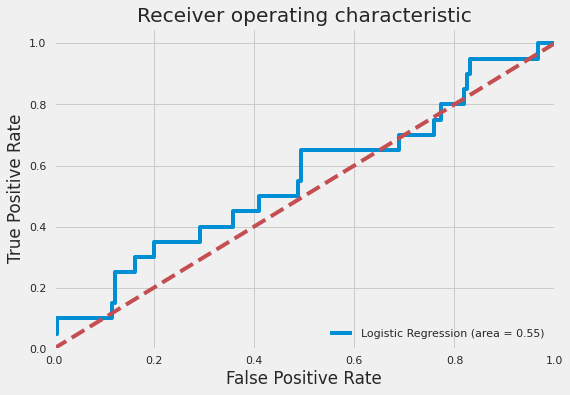


This model shows the probability/likelihood that The J. M. Smucker Co (SJM) is going to be involved in a lawsuit at 14.27% which is great because it's lower than the 50% percent threshold that binary choice presents . As such we can be sure that our company is likely not sued.

## ROC CURVE

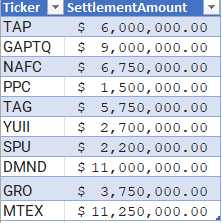
The best cut-off has the lowest false positive rate and the highest true positive rate.

The area under a ROC curve is used to compare the usefulness of tests because it is a measure of the usefulness of a test in general, with a larger area indicating a more valuable test. According to our graph, the True Positive Rate is almost 99 percent.



## Severity /Impact.

To predict the severity of the likelihood predicted earlier we would have to use the variable settlement amount. Looking into the feature I found that out of 347 observations only 46 of them had numeric values. Further investigation showed that only 10 companies had actually settled amounts.



As such due to lack of data I would not be pursuing any Multivariate Analysis (MVA) algorithms. Instead I would just calculate a weight average with a 95% confidence interval. Attempts to use the market cap for the companies were unsuccessful as some for the companies list did not have listed online either due to the business closing or some other reason. So I will be comparing the weighted average impact with just the market cap of SJM which is 15.118Billion dollars as of today. So we can say The 95% confidence intervals for the settlement amount averages are +/- (or between $3,457,135.89 and $8,522,864.10).

APPENDIX

# -\*- coding: utf-8 -\*-

"""likelihood & severity .ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/19PaiG11Es7vg1ljTpdBtXjbF0-a8GWh5

#Importing / Installing packages

"""

# Commented out IPython magic to ensure Python compatibility.

import pandas as pd

import numpy as np

import statsmodels.api as sm

import scipy.stats as st

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import confusion\_matrix

import matplotlib.mlab as mlab

# %matplotlib inline

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn import preprocessing

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

from matplotlib import pyplot as plt

# Increases the size of sns plots

sns.set(rc={'figure.figsize':(12,10)})

"""Full Display of outputs """

pd.options.display.max\_columns= None

pd.options.display.max\_rows= None

"""Loading the Raw Data"""

from google.colab import drive

drive.mount('/content/drive')

# reading the Final data file

rootdir="/content/drive/MyDrive"#update your root directory

df=pd.read\_csv(rootdir+"/finalstuff.csv")

"""## Exploritory Data Analysis"""

# getting the structure of the final data set.

df.info()

df.describe().transpose()

df.shape

df.describe().transpose()

#checking for missing values

df.isnull().sum()

# Investigate all the elements whithin each Feature

for column in df:

unique\_values = pd.unique(df[column])

nr\_values = len(unique\_values)

if nr\_values <= 10:

print("The number of values for feature {} is: {} -- {}".format(column, nr\_values, unique\_values))

else:

print("The number of values for feature {} is: {}".format(column, nr\_values))

# Investigating the distr of y

sns.countplot(x = 'Sued', data = df, palette = 'Set3')

df.columns

# Looping through all the features by our y variable - see if there is relationship

features = [ 'ggroup', 'gsector', 'gind', 'gsubind', 'state', 'sic','Sued',]

for f in features:

sns.countplot(x = f, data = df, palette = 'Set3', hue = 'Sued')

plt.show()

!pip install autoviz

import autoviz

from autoviz.AutoViz\_Class import AutoViz\_Class

av=AutoViz\_Class()

autoviz\_eda=av.AutoViz('/content/drive/MyDrive/finalfinal1.csv', depVar="Sued",sep = ",",lowess=False,chart\_format="svg",max\_rows\_analyzed=150000,max\_cols\_analyzed=30,)

"""## Feature Selection- With Correlation

\*\*Steps of Running Feature Importance\*\*

\* Split the data into X & y

\* Run a Tree-based estimators (i.e. decision trees & random forests)

\* Run Feature Importance

"""

dt=df.copy()

dt.head()

dt.columns

X = dt.drop(["gvkey","Sued","tic","state","SettlementAmount"],axis=1) #Feature Matrix

y = dt["Sued"]

X.head()

#separate dataset into train and test

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X,

y,

test\_size=0.5,

random\_state=0)

X\_train.shape, X\_test.shape

"""Feature Importance """

from sklearn.ensemble import RandomForestClassifier

Ftimp = RandomForestClassifier(random\_state=0,criterion ='entropy', max\_depth = 10)

Ftimp.fit(X,y)

!pip install eli5

import eli5

from eli5.sklearn import PermutationImportance

perm = PermutationImportance(Ftimp, random\_state=0).fit(X\_train, y\_train)

eli5.show\_weights(perm, feature\_names = X\_train.columns.tolist())

"""correlation """

X\_a = dt[['cshtr\_c' , 'reuna' , 'wcap' , 'pi' , 'xsga' , 'seq' , 'ppent' , 'prcc\_diff' , 'trt1m' , 'prch\_f' , 'mkvalt' , 'txt' , 'cshfd' , 'nopi' , 'dvpsp\_f' , 'prcl\_f' , 'dltt' , 'cshoc' , 'prch\_c']]

#create a new split with top 10 feature

# separate dataset into train and test

from sklearn.model\_selection import train\_test\_split

X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(

X\_a,

y,

test\_size=0.5,

random\_state=0)

X\_train.shape, X\_test.shape

X\_a.corr()

import seaborn as sns

#Using Pearson Correlation

plt.figure(figsize=(20,20))

cor = X\_a.corr()

sns.heatmap(cor, annot=True, cmap=plt.cm.CMRmap\_r)

plt.show()

# with the following function we can select highly correlated features

# it will remove the first feature that is correlated with anything other feature

def correlation(dataset, threshold):

col\_corr = set() # Set of all the names of correlated columns

corr\_matrix = dataset.corr()

for i in range(len(corr\_matrix.columns)):

for j in range(i):

if (corr\_matrix.iloc[i, j]) > threshold: # we are interested in absolute coeff value

colname = corr\_matrix.columns[i] # getting the name of column

col\_corr.add(colname)

return col\_corr

corr\_features = correlation(X\_a, 0.8)

len(set(corr\_features))

corr\_features

"""Feature selection

##Evaluate model parameters using statsmodels

"""

# module imports

from patsy import dmatrices

import numpy as np

import statsmodels.api as sm

from statsmodels.tools import add\_constant as add\_constant

#fit regression model

model = sm.OLS(y, X\_a).fit()

#view summary of model fit

print(model.summary())

"""## Logistic Regression final take.

"""

# module imports

from patsy import dmatrices

import numpy as np

import statsmodels.api as sm

from statsmodels.tools import add\_constant as add\_constant

y2, X2 = dmatrices('Sued ~ + prcl\_f + prch\_f + mkvalt + cshoc ', dt, return\_type = 'dataframe')

#add constant to predictor variables

X2 = sm.add\_constant(X2)

#fit regression model

model = sm.OLS(y2, X2).fit()

#view summary of model fit

print(model.summary())

"""The p-values for all of the variables are very small, below the 0.05 threshold as such there are are significant to the model.

saving the final model

"""

#import pickle

# save the model to disk

#rootdir="/content/drive/MyDrive

#filename = '/content/drive/MyDrive/finalized\_model.sav'

#pickle.dump(model, open(filename, 'wb'))

# load the model from disk

loaded\_model = pickle.load(open(filename, 'rb'))

result = loaded\_model.predict(X\_test, Y\_test)

"""### Logistic Regression Model Fitting"""

# Creating new Independent feature based of significance on the OLS Regression Results

X\_last = dt[["prcl\_f","prch\_f","mkvalt","cshoc"]]

# separate dataset into train and test

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X\_last,

y,

test\_size=0.5,

random\_state=0)

X\_train.shape, X\_test.shape

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

logreg = LogisticRegression(random\_state=0, solver = 'lbfgs')

logreg.fit(X\_train, y\_train)

"""###Predicting the test set results and caculating the accuracy"""

# predict - Predict class labels for samples in X

logreg.predict(X\_train)

ypred = logreg.predict(X\_train)

# predict\_proba - Probability estimates

pred\_proba = logreg.predict\_proba(X\_train)

# coef\_ - Coefficient of the features in the decision function

logreg.coef\_

# score- Returns the mean accuracy on the given test data and labels - below

y\_pred = logreg.predict(X\_test)

print (y\_pred)

y\_pred1 = logreg.predict\_proba(X\_test)[:,1]

print (y\_pred1)

#Predicting file for submission

final\_prob = pd.Series(logreg.predict\_proba(X\_test)[:,1], name="probalility\_of\_event")

ID\_Column = dt["gvkey"]

ID\_Column\_company = dt[dt.gvkey== 9777]

submission1 = pd.concat([ID\_Column,final\_prob], axis=1)

submission1.head()

# Applying the SJM to the prediction

sub2 =submission1[submission1.gvkey ==9777]

print(sub2)

print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X\_test, y\_test)))

#print the regression coefficients

print("The intercept b0= ", logreg.intercept\_)

print("The coefficient b1= ", logreg.coef\_)

"""###Cross Validation"""

from sklearn import model\_selection

from sklearn.model\_selection import cross\_val\_score

kfold = model\_selection.KFold(n\_splits=10)

modelCV = LogisticRegression()

scoring = 'accuracy'

results = model\_selection.cross\_val\_score(modelCV, X\_train, y\_train, cv=kfold, scoring=scoring)

print("10-fold cross validation average accuracy: %.3f" % (results.mean()))

"""###Confusion Matrix"""

from sklearn.metrics import confusion\_matrix

confusion\_matrix = confusion\_matrix(y\_test, y\_pred)

print(confusion\_matrix)

"""The result is telling us that we have 150+18 correct predictions and 4+2 incorrect predictions."""

#def plot\_confusion\_matrix(confusion\_matrix, classes=None, title='Confusion matrix'):

# """Plots a confusion matrix."""

#if classes is not None:

# sns.heatmap(confusion\_matrix, confusion\_matrixap="YlGnBu", xticklabels=classes, yticklabels=classes, vmin=0., vmax=1., annot=True, annot\_kws={'size':50})

# else:

#sns.heatmap(confusion\_matrix, vmin=0., vmax=1.)

# plt.title(title)

# plt.ylabel('True label')

# plt.xlabel('Predicted label')

# Visualizing confusion\_matrix

import matplotlib.pyplot as plt

from sklearn.metrics import ConfusionMatrixDisplay

disp = ConfusionMatrixDisplay(confusion\_matrix=confusion\_matrix,display\_labels=logreg.classes\_)

disp.plot()

plt.tick\_params(axis=u'both', which=u'both',length=0)

plt.grid(b=None)

plt.show()

# Getting the predicted values on the train set and showing first 10 predictions in terms of probabilities

y\_train\_pred = logreg.predict(X\_test)

y\_train\_pred[:10]

"""###Evaluating the Model"""

# Accuracy on Train

print("The Training Accuracy is: ", logreg.score(X\_train, y\_train))

# Accuracy on Test

print("The Testing Accuracy is: ", logreg.score(X\_test, y\_test))

# Classification Report

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))

"""Interpretation:

Of the entire test set, 88% of the promoted term deposit were the term deposit that the customers liked. Of the entire test set, 90% of the customer's preferred term deposit were promoted.

"""

confusion\_matrix/confusion\_matrix.sum(axis=1)

confusion\_matrix

confusion\_matrix.sum(axis=0)

np.diag(confusion\_matrix)

# Calculating False Positives (FP), False Negatives (FN), True Positives (TP) & True Negatives (TN)

FP = confusion\_matrix.sum(axis=0) - np.diag(confusion\_matrix)

FN = confusion\_matrix.sum(axis=1) - np.diag(confusion\_matrix)

TP = np.diag(confusion\_matrix)

TN = confusion\_matrix.sum() - (FP + FN + TP)

# Sensitivity, hit rate, recall, or true positive rate

TPR = TP / (TP + FN)

print("The True Positive Rate is:", TPR)

# Precision or positive predictive value

PPV = TP / (TP + FP)

print("The Precision is:", PPV)

# False positive rate or False alarm rate

FPR = FP / (FP + TN)

print("The False positive rate is:", FPR)

# False negative rate or Miss Rate

FNR = FN / (FN + TP)

print("The False Negative Rate is: ", FNR)

##Total averages :

print("")

print("The average TPR is:", TPR.sum()/2)

print("The average Precision is:", PPV.sum()/2)

print("The average False positive rate is:", FPR.sum()/2)

print("The average False Negative Rate is:", FNR.sum()/2)

"""###ROC Curvefrom sklearn import metrics"""

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

logit\_roc\_auc = roc\_auc\_score(y\_test, logreg.predict(X\_test))

fpr, tpr, thresholds = roc\_curve(y\_test, logreg.predict\_proba(X\_test)[:,1])

plt.figure()

plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit\_roc\_auc)

plt.plot([0, 2], [0, 2],'r--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic')

plt.legend(loc="lower right")

plt.savefig('Log\_ROC')

plt.show()

"""####Severity or cost of the Likelyhood """

dt.SettlementAmount.unique()

dt['SettlementAmount']=dt['SettlementAmount'].replace(np.nan,'')

dt.head(64)

dt.shape

impact = dt[["gvkey","tic","SettlementAmount"]]

impact.head(380)

import numpy as np

import scipy.stats as st

#define sample data

data = [ 6000000.0, 9000000.0, 6750000.0, 1500000.0, 5750000.0,11250000.0, 2700000.0, 2200000.0, 11000000.0,3750000.0]

#create 95% confidence interval for population mean weight

st.t.interval(alpha=0.95, df=len(data)-1, loc=np.mean(data), scale=st.sem(data))

6000000.0, 9000000.0, 6750000.0, 1500000.0, 5750000.0,

11250000.0, 2700000.0, 2200000.0, 11000000.0,

3750000.0