Financial Distress Prediction

Project Final Report

ITCS 5156 - APPLIED MACHINE LEARNING

Submitted by **Group 8**

Name	Student ID
Bentic Sebastian	801170368
Navneeth Sreenivasan	801210187
Sharath Chandra Mouli Reddy Kancharla	801165873
Sumanth Kuthuru	801166362

Submitted to

Minwoo “Jake” Lee

Github link: https://github.com/bsebast2/AppliedMLProject.git

INTRODUCTION

In this project, we attempt to create a Regression model to predict when a company is in financial distress. Being able to predict companies close to financial distress will help investors make decisions to protect themselves, or invest more and help these companies prevent bankruptcy in advance because the collective number of failing companies can be regarded as an important indicator of the financial health and robustness of a country's economy.

Financial distress is a term in corporate finance used to indicate a situation when promises to creditors of a company are broken or cannot be reciprocated easily. If Financial Distress cannot be relieved, it can lead to bankruptcy, which is a severe risk to a company's relationships and credibility.

Financial Distress associated with companies is one of the biggest threats for business. Finding a method to identify corporate financial distress as early as possible is clearly a matter of considerable interest to investors, creditors, auditors and other stakeholders. There are several reasons why Financial distress is important for companies:

(1) Bankers and other lenders will tend to look upon a request for further finance from a financially distressed company with a prejudiced eye – taking a safety-first approach – and this can continue for many years after the crisis has passed
(2) The rate of bankruptcy (Indicator of Financial Distress) has increased during recent years
(3) Suppliers providing goods and services on credit are likely to reduce the generosity of their terms, or even stop supplying altogether, if they believe that there is an increased chance of the firm not being in existence in a few months' time
(4) In a financial distress situation, employees may become demotivated as they sense increased job insecurity and few prospects for advancement
(5) Management find that much of their time is spent "fire fighting" – dealing with day-to-day liquidity problems – and focusing on short-term cash flow rather than long-term shareholder wealth

Problem Statement

The problem that we have investigated is how to predict **Financial Distress** so that investors, creditors, auditors and other stakeholders can use the information to learn about the current state of a company's financial state.

We aim to use Machine Learning to predict if a company is in Financial Distress at a given point

in time. Our aim is to build a **Regression model** to predict when a company is in financial distress and therefore, has a high chance to go bankrupt.

Our project is different from the existing approaches, as we are using Regression models rather than Classification models to analyze financial distress. In our literature review, we have discovered that Classification is a popular technique for predicting Financial distress. We will also be using a stacking approach, which involves combining three of our highest scoring models and comparing the accuracy scores with the individual Regression models and existing Classification models in current literature.

Motivation and Challenges

As companies become bigger and more complex, it becomes difficult to study all the features without using Machine Learning. This large and complex feature set requires Machine Learning algorithms to help with prediction.

There are several motivations for creating models to predict Financial Distress:

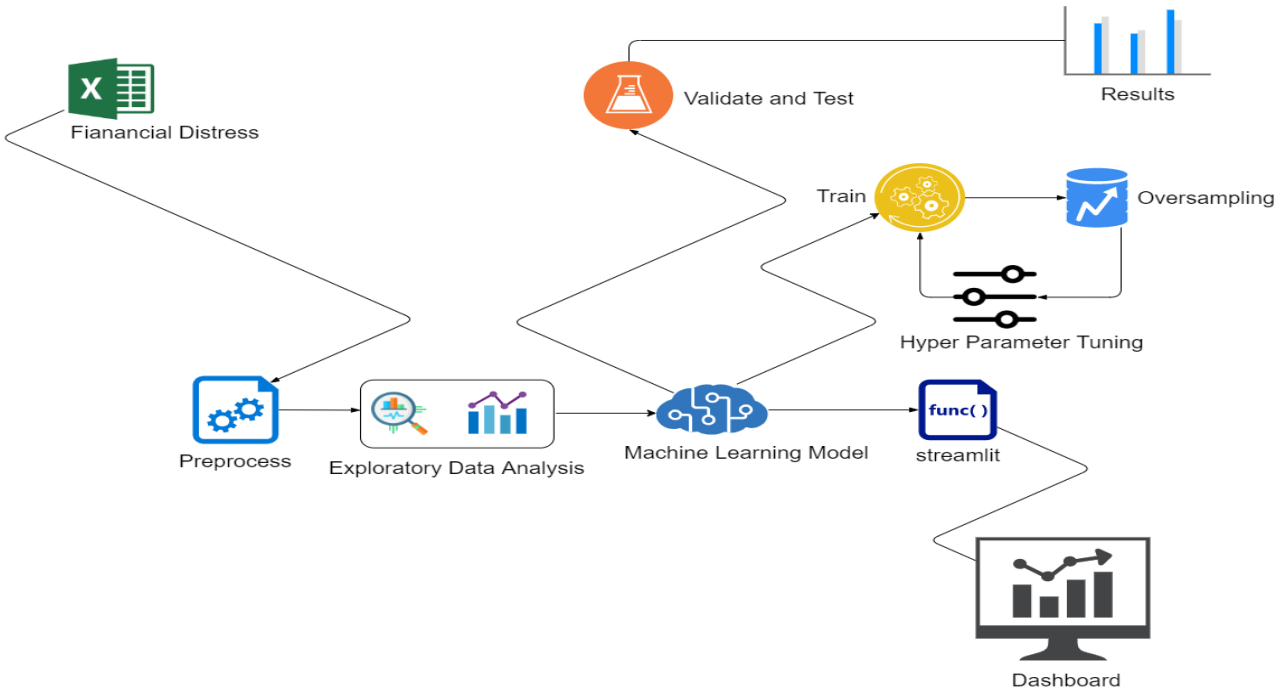
(1) Being able to predict companies which are about to go broke will help investors make decisions to protect themselves, or invest more

(2) It can also help the companies prevent going bankrupt in advance, and make long-term decisions

(3) A country's economic departments would benefit from the collective number of failing companies, which can signal a crisis and challenge the financial health and robustness of a country's economy

Some of the major challenges with current Machine Learning models are the lower prediction scores. Most of the models score on average around 85%. This is quite low, and we hope that we can meet this challenge with a novel approach: Stacking.

Concise Summary of Approach



(Fig 1 - Concise Summary of our Approach)

In the diagram above, we can see all the steps that we have taken to understand our dataset. We start with Preprocessing our dataset, which involves checking for outliers, imputing , scaling the data and feature extraction. We use scatter plots, histograms, and correlation matrices to look for any patterns. We then train and test several Machine Learning Regression models, including Linear Regression, Polynomial Regression, Decision Trees, Random Forest models, etc. We then present our results through a Streamlit Dashboard interface and compare the results to the current literature.

BACKGROUND/RELATED WORK

Deep enough survey of literature

(1) In general, the task of financial distress prediction is to predict whether the firm will go bankrupt or not, which is a binary classification problem

(2) Bankruptcy prediction has been a significant issue in finance and management science, which attracts the attention of researchers and practitioners

(3) For financial institutions, the ability to predict or forecast business failures is crucial, as incorrect decisions can have direct financial consequences

(4) Bankruptcy prediction and credit scoring are the two major research problems in the

accounting and finance domain

Literature Survey and results from some of the papers:

(1) Zhao et. al. (2014) built an automatic credit scoring system with high accuracy (87%) and efficiency using Multi-Layer Perceptron Neural Network (MLPNN), with the experiment on German credit data. Multilayer Perceptron (MLPNN) is the most widely used baseline classifier for the prediction of financial crises [1]
(2) Shin, Lee and Kim (2005) used SVM in corporate bankruptcy prediction and achieved better accuracy than back-propagation (SVM-84.0%, compared to BPN-55.4%)
(3) Nanni and Lumini (2009) conducted a series of experiments on financial datasets including Australian credit data, German credit data and Japanese credit data, and found that ensemble method led a better classification performance than stand-alone models in bankruptcy prediction and credit scoring [6]

Summary of approaches, pros and cons

These are some of the classification models previously used in Financial Distress prediction -

(1) Classical machine learning models :
Linear/Multivariate Discriminant Analysis (LDA/MDA)
Logistic Regression (LR)
Ensemble method
Neural Networks (NN), such as Multilayer Perceptron (MLP)
Support Vector Machines (SVM)
(2) Deep Learning Models:
Convolutional Neural Networks (CNN)
Deep Belief Networks (DBN)

Here is a table of the results used in [3]. It is important to note that the Ensemble Approach, in the last row, allowed for a higher accuracy. It can be seen that using an ensemble approach has an advantage.

Table 1. Comparative Accuracy of classifiers without Feature Selection

Sr. No	Classifier Used	Accuracy on Training and Test sets with cross classification					
		60:40		70:30		50:50	
		Training	Test	Training	Test	Training	Test
1.	LR	79.86	76.53	78.14	75.67	79.67	75.44
2.	SVM	99.66	73.84	99.57	72.00	99.59	71.73
3.	CART	79.86	71.15	74.00	69.67	75.15	70.96
4.	CHAID	80.88	68.95	79.00	69.67	79.47	69.79
5.	NN	75.8	71.39	78.00	67.00	73.92	68.81
6.	C5.1	83.93	71.39	87.14	70.33	83.57	68.81
7.	QUEST	76.48	71.88	81.14	70.67	77.62	69.20
8.	Ensembled	100	86.1	99.75	80.59	100	85.32

(Table 1 - Comparative Accuracy of Classifiers)

In general, the basis of the financial position and performance evaluation is the accounting data which is fundamentally considered as input variable of standard financial ratios in financial analysis [4] (Salehi & Abedini, 2009). Nevertheless, certain models, such as Ohlson's model, include an index of gross national product (Ghodrati & Moghaddam, 2012) [5].

Within the elaborated analysis of data, the authors examined a sample of twenty bankruptcy models which are frequently used in the Czech Republic. Within the models frequency of occurrences of each ratio have been defined in these models. Then, there was an established weighted average of coefficients, which are assigned to these indicators. Their overview is presented in Table 2. The below given in Table 2 are some of the main factors that may have a huge effect on the company that may go into financial distress which are given according to their frequency. This is a simple view of an idea based on a research paper [7] published in 2013.

Ratio	Frequency	Weighted Average of Ratio's Weight
Sales / Assets	11	0,70685
Retained Earnings / Assets	8	0,19000
EBIT / Assets	8	0,41347
Working Capital / Assets	8	0,21496
Current Assets / Current Liabilies (Current Ratio)	5	0,58575
Equity / Debts	5	0,06155
EAT / Assets (ROA)	4	0,39575
Debts / Assets	3	–0,09562
EBT / Current Liabilities	2	0,32895
EAT / Sales (ROS)	2	0,32893
EBT / Assets	2	0,21176

Source: authors' elaboration

(Table 2 - Weighted Average of Ratio's Weight)

Within the list of bankruptcy models in [7] (Bellovary, Giacomino & Akers, 2007), the most common indicators in the following order are:

1. Net income/Total Assets (in 54 models);
2. Current Ratio (in 51 models);
3. Working Capital/Total Assets (in 45 models);
4. Retained Earnings/Total Assets (in 42 models);
5. EBIT/Total Assets (in 35 models);
6. Sales/Total Assets (in 32 models), etc.

Pros: Most of the models in the history based on our research were done on classification. Classification tries to draw a line based on binary data whereas Regression tries to fit the data. Hence, for continuous values data, regression can get better results when financial distress is a continuous variable.

Cons: Classification allows us to see relationships between the things that may not be obvious when looking at them as a whole. Since the data is considered binary in classification, it draws a line and tries to predict the target values.

Relation to Our Approach

As seen previously, financial distress models have usually been classification models. Secondly,

the Ensemble approach yielded a higher score than the individual classifiers.

We want to use these two findings, and extend the study. We have created several Regression-based models to see if they produce a higher accuracy. We have also stacked the three best models, and compare them to our literature review. We hope to see an increase in accuracy, as regression models are more accurate in general but sensitive to outliers.

METHOD

We have already seen the importance of predicting Financial Distress for a Company and for the Investors and Suppliers as well and have also seen the existing approaches which use the Classification approach to predict Financial Distress. We have taken a different approach in predicting Financial Distress i.e., by using Regression. We first fit the data and then categorize the output into 0's and 1's where 0 indicates company is healthy and vice versa, and this would definitely result in higher accuracy. There are different steps involved in our Method: Data Preprocessing, Exploratory Data Analysis, Regression Algorithms / Models, Model Building, Evaluation.

Description of the Data

We took Data for this experiment from Kaggle, a well known Data resource for most Machine Learning Experimentation. The Dataset consists of a total of 86 columns in which the Financial Distress column is the Target variable. There are two named columns - Company and Time - and all other features are denoted as x1 to x83. These are some of the financial and non-financial characteristics of the sampled companies. These features belong to the previous time period, which should be used to predict whether the company will be financially distressed or not. Feature **x80** is a **categorical variable**. From the below figure (Fig 2), we can see that there are a total of 3672 rows. Each Company has data measured at 14 different time periods but if the company goes into Financial Distress at some time period then that particular company's data ends there. A company is said to be in Financial Distress if its Financial Distress value is less than or equal to -0.5; otherwise it is considered healthy.

	Company	Time	Financial Distress	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x1
0	1	1	0.010636	1.2810	0.022934	0.87454	1.21640	0.060940	0.188270	0.52510	0.018854	0.182790	0.006449	0.85822	2.005800e+00	0.125460	6.97060	4.65120	0.05010
1	1	2	-0.455970	1.2700	0.006454	0.82067	1.00490	-0.014080	0.181040	0.62288	0.006423	0.035991	0.001795	0.85152	-4.864400e-01	0.179330	4.57640	3.75210	-0.01401
2	1	3	-0.325390	1.0529	-0.059379	0.92242	0.72926	0.020476	0.044865	0.43292	-0.081423	-0.765400	-0.054324	0.89314	4.122000e-01	0.077578	11.89000	2.48840	0.02807
3	1	4	-0.566570	1.1131	-0.015229	0.85888	0.80974	0.076037	0.091033	0.67546	-0.018807	-0.107910	-0.065316	0.89581	9.949000e-01	0.141120	6.08620	1.63820	0.09390
4	2	1	1.357300	1.0623	0.107020	0.81460	0.83593	0.199960	0.047800	0.74200	0.128030	0.577250	0.094075	0.81549	3.014700e+00	0.185400	4.39380	1.61690	0.23921
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
3667	422	10	0.438020	2.2605	0.202890	0.16037	0.18588	0.175970	0.198400	2.22360	1.091500	0.241640	0.226860	0.35580	1.550000e+07	0.839630	0.19101	12.09200	0.94673
3668	422	11	0.482410	1.9615	0.216440	0.20095	0.21642	0.203590	0.189870	1.93820	1.000100	0.270870	0.213610	0.38734	1.920000e+07	0.799050	0.25149	2.63990	0.94073
3669	422	12	0.500770	1.7099	0.207970	0.26136	0.21399	0.193670	0.183890	1.68980	0.971860	0.281560	0.210970	0.44290	2.030000e+07	0.738640	0.35384	1.49010	0.90501
3670	422	13	0.611030	1.5590	0.185450	0.30728	0.19307	0.172140	0.170680	1.53890	0.960570	0.267720	0.203190	0.47601	1.099600e+02	0.692720	0.44358	1.38370	0.89163
3671	422	14	0.518650	1.6148	0.176760	0.36369	0.18442	0.169550	0.197860	1.58420	0.958450	0.277780	0.213850	0.51969	1.542500e+01	0.636310	0.57156	0.42332	0.91935

3672 rows × 86 columns

(Fig 2 - Visualizing the Dataset)

Data Preprocessing

We load the Data into a Pandas Dataframe and start Data Preprocessing and Data Analysis.There are different steps involved in Data Preprocessing:

(1) Checking for Null Values: There are no null values in the given Data set

(2) Total Companies in Distress: There are a total of 136 financially distressed companies against 286 healthy ones

(3) Dropping Columns: We drop the columns Company, Time, x80 which are not required and are redundant for our experiment

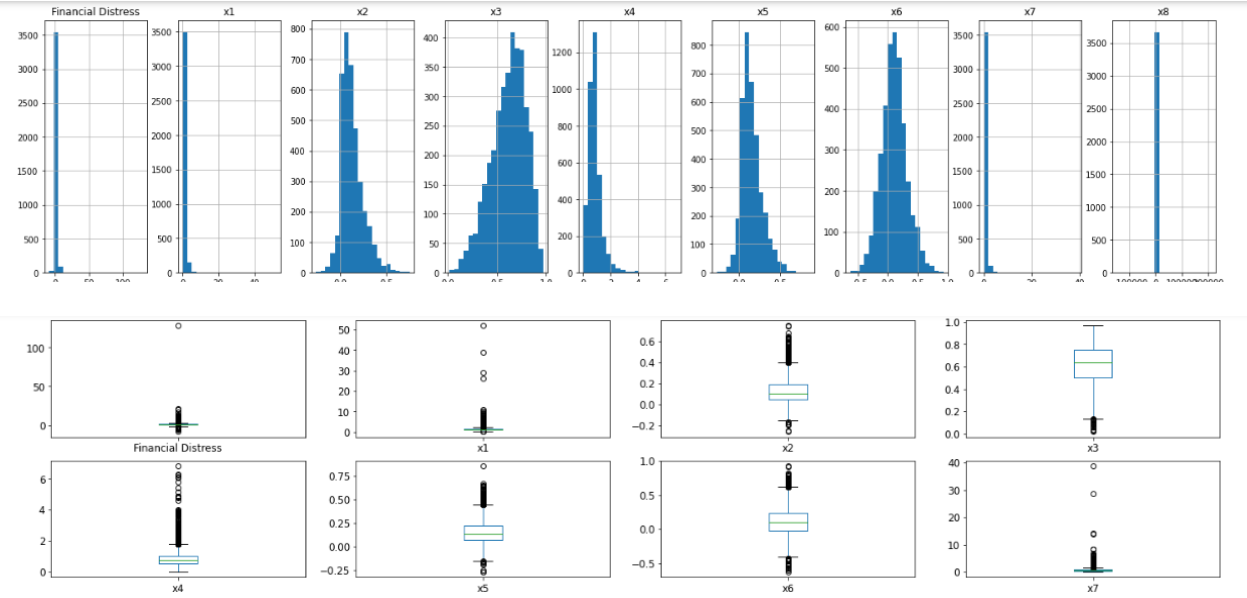
(4) Descriptive Statistics: From the below figure (Fig 3) we can see that there are some extreme values in some of the columns. Some are highly skewed and others are not

	Financial Distress	x1	x2	x3	x4	x5	x6	x7	x8	x9	...	x73
count	3672.000000	3672.000000	3672.000000	3672.000000	3.672000e+03	3672.000000	3672.000000	3672.000000	3672.000000	3672.000000	...	3672.000000
mean	1.040257	1.387820	0.129706	0.615769	8.681599e-01	0.154949	0.106717	0.784031	39.274361	0.332610	...	30.407166
std	2.652227	1.452926	0.120013	0.177904	5.719519e-01	0.124904	0.210555	1.033606	4305.688039	0.346135	...	3.714512
min	-8.631700	0.075170	-0.258080	0.016135	5.350000e-07	-0.269790	-0.627750	0.035160	-145000.000000	-3.611200	...	22.000000
25%	0.172275	0.952145	0.048701	0.501888	5.525575e-01	0.070001	-0.027754	0.436003	0.056185	0.157677	...	28.000000
50%	0.583805	1.183600	0.107530	0.638690	7.752450e-01	0.131830	0.104325	0.641875	0.135585	0.302610	...	30.000000
75%	1.351750	1.506475	0.188685	0.749425	1.039000e+00	0.219570	0.231230	0.896773	0.273423	0.484035	...	33.000000
max	128.400000	51.954000	0.749410	0.967900	6.835600e+00	0.858540	0.929550	38.836000	209000.000000	3.810200	...	36.750000
skewness	30.873600	20.058157	1.026241	-0.514097	3.214546e+00	0.871433	0.169445	21.014228	21.162111	0.712157	...	-0.158215
kurtosis	1451.206671	579.289645	1.815265	-0.232932	2.019422e+01	1.341499	0.351143	664.142698	1865.899414	19.439512	...	-0.337231

10 rows × 83 columns

(Fig 3 - Descriptive Statistics of the data)

(5) Histogram and Box plots: The below given figures of the plots (Fig 4) confirm there are Outliers that need to be replaced or removed

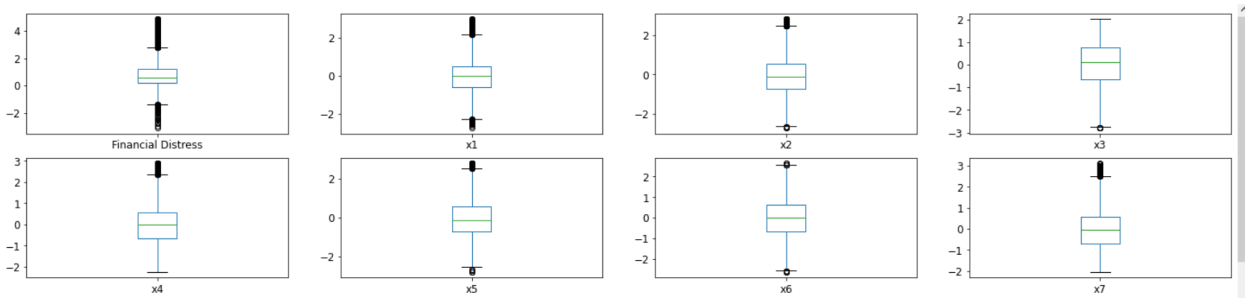


(Fig 4 - Plotting Histograms and Box plots for raw data)

(6) Dealing with Outliers: Although we have an option to remove outliers, we don't want to lose the data. So we came to the conclusion to impute them with the mean value of that column

(7) Imputing: We detected Outliers using the formula (Outlier if < Q1 - 1.5*IQR and > Q3 + 1.5*IQR), but this range removes some significant number of values in the Financial Distress column. So we considered (Outlier if < Q1 - 3*IQR and > Q3 + 3*IQR) just for the column Financial Distress. Now we substitute all the values that consist of outliers with the mean of that column

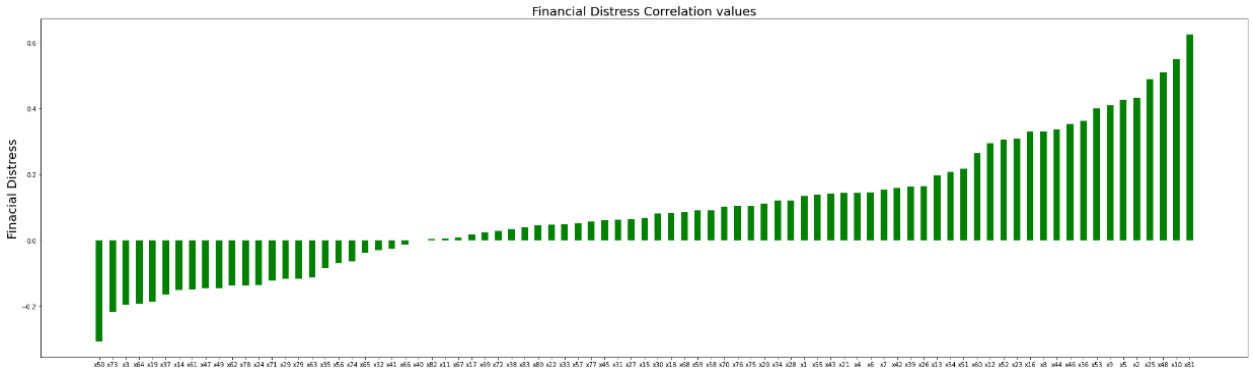
(8) Standardizing: Standardization of Data is a norm in a Machine Learning Pipeline. Since we have 82 features and the Target Variable (Financial Distress) of different ranges and also, while training a Machine Learning Model the features with higher range would have more weight, which we need to avoid, we Standardize all the columns except Target Variable since we need raw output of Predictive Model to classify values into 0's and 1's where 1 implies the company is in Financial Distress (<=-0.5) and 0 otherwise. The below figure (Fig 5) shows how Data looks after Data Preprocessing.



(Fig 5 - Box plot of Dataset after Preprocessing)

Exploratory Data Analysis

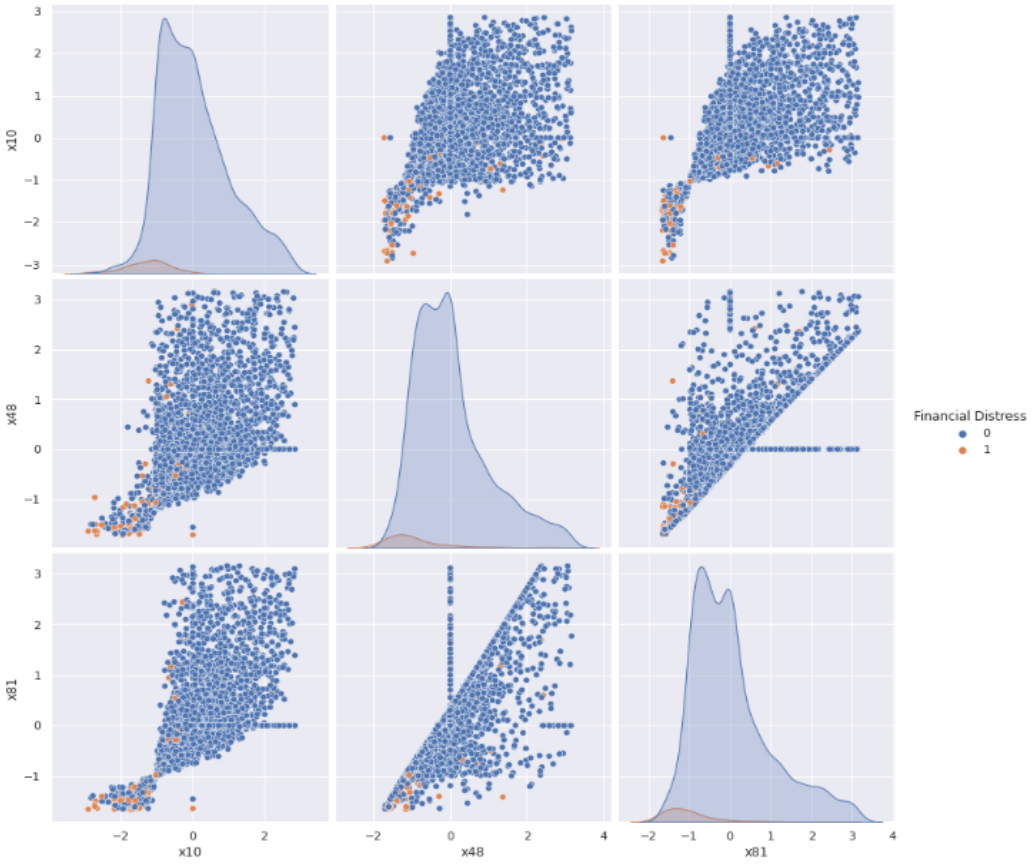
We do Data Visualization to better understand how Financial Distress depends on features. But we don't have feature names and it's difficult to plot Financial Distress against all the features, so we plot a Correlation Heatmap to see the highly correlated features. From the figure below (Fig 6), we can see correlation of all features with Financial Distress.



(Fig 6 - Correlation of the features with Financial Distress)

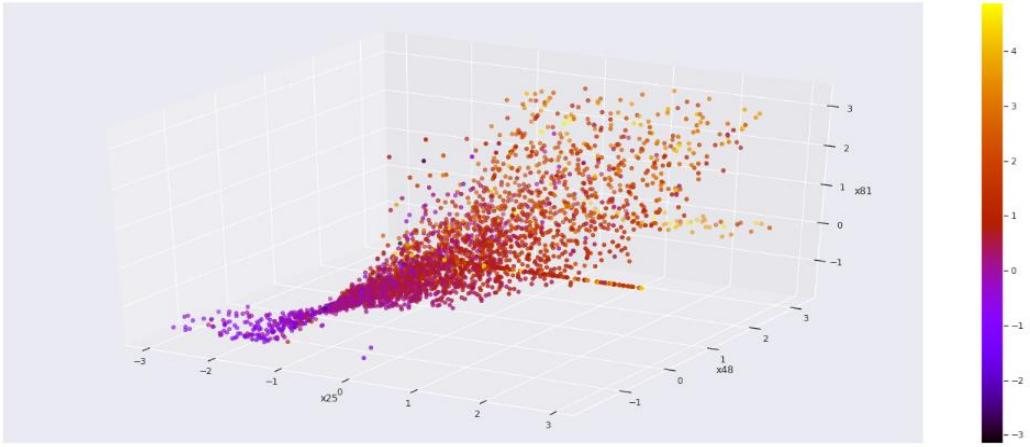
We consider all the features that have <-0.5 and > 0.5 correlation with the Target variable and we find there are only three of them (There are actually four, but for visualization purposes we considered three): x25, x48 and x81. We do the Visualizations against the Target variable using these three highly correlated columns.

But looking at correlation among the features themselves, the correlation seems to be high. Fig 7 shows the features that are highly correlated.



(Fig 7 - Scatter Plot between the highly Correlated Features)

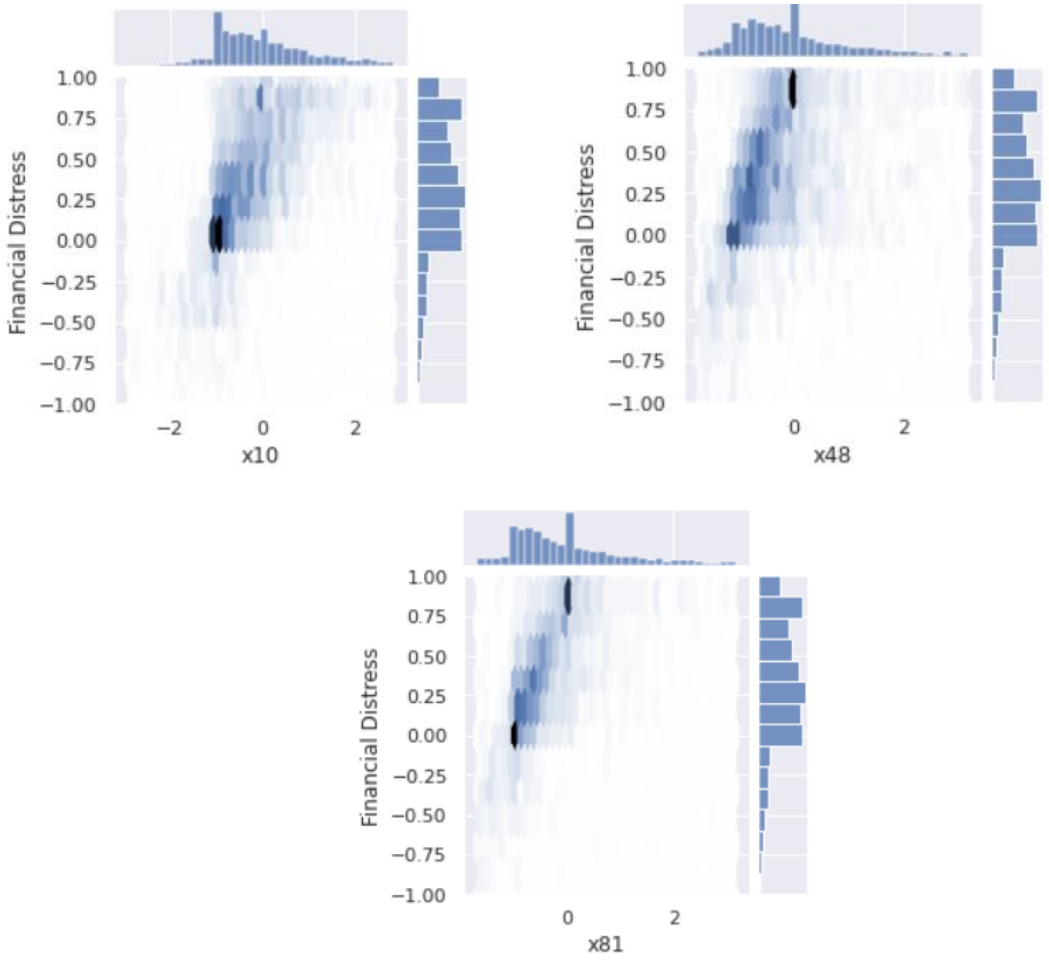
Now let's look at the 3D plot to see how Financial Distress changes with the three correlated features. Fig 8 shows how this is implemented.



(Fig 8 - 3D Scatterplot between the correlated features)

We can see how Financial Distress increases linearly with all three and it's safe to assume that they will have a higher weight when building regression models.

Another plot reveals in what areas Financial Distress values are denser against all the three features (Fig 9).



(Fig 9 - Plot of density between Financial distress and the highly correlated features)

All these plots have revealed some information about Target variables dependency and we must be aware of these dependencies when building a Regression Model.

Regression Algorithms / Models

The different Machine Algorithms we used in our experiment are:

(1) Linear Regression: Linear regression is a statistical analysis which depends on modelling a relationship between two kinds of variables, dependent (response) and independent (predictor). The main purpose of regression is to examine if the independent variables are successful in predicting the outcome variable and which independent variables are significant predictors of the outcome [8]. A linear regression line has an equation of the form

$$Y = a + bX,$$

where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when x = 0).

(2) Ridge Regression: Ridge regression is a popular parameter estimation method used to address the collinearity problem frequently arising in multiple linear regression. Algebraic properties of the ridge regression coefficients, if given, will elucidate the behaviour of a ridge trace for small values of the ridge parameter and for large values of the ridge parameter [9]. It can be written as

$$\underline{\sim B} = (R + kI)^{-1} X'\underline{Y}$$

(3) Lasso Regression: Lasso regression methods are widely used in domains with massive datasets, such as genomics, where efficient and fast algorithms are essential [10]. The lasso estimator uses the ℓ1 penalized least squares criterion to obtain a sparse solution to the following optimization problem:

$$\beta\hat{} \text{ (lasso)} = \arg\min(\beta) \, ||y - X\beta||_2{}^2 + \lambda||\beta||_1$$

(4) Elastic Net: The elastic net procedure is a form of regularized optimization for linear regression that provides a bridge between ridge regression and the lasso. The estimate that it produces can be viewed as a Bayesian posterior mode under a prior distribution implied by the form of the elastic net penalty [11]. It can be represented as:

$$\beta\hat{} = \arg\min(\beta) \, ||y - X\beta||^2 + \lambda_2||\beta||^2 + \lambda_1||\beta||_1$$

(5) Polynomial Regression: Polynomial regression is a special case of multiple regression, with only one independent variable X. One-variable polynomial regression model can be expressed as [12] :

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_k x_i^k, \text{ for } i = 1,2,\ldots,n$$

where *k* is the degree of the polynomial. The degree of the polynomial is the order of the model.

(6) Stochastic Gradient Descent: Stochastic gradient descent (SGD) performs a parameter update for each training example x_i and label yi. The Stochastic Gradient Descent (SGD) procedure is an extension of the Gradient Descent (GD) to stochastic optimization of f as follows [13]:

$$X_{t+1} = x_t - \eta_t \bigtriangledown \mathbf{f}_t(x_t),$$

where η_t is the learning rate. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster and can also be used to learn online.

(7) Decision Trees: A decision tree is a recursive structure for expressing classification rules. Decision tree regression is a type of tree-based structure used to predict the numeric outcomes of the dependent variable. A Decision tree uses a splitting criterion that minimizes the intra-subset variation in the class-values of instances that go down each branch. The attribute, which maximizes the expected error reduction, is chosen as the root node. Next, the tree is pruned back from each leaf. Finally, a smoothing procedure is used to compensate for the sharp discontinuities [14].

(8) Random Forests: Random forests or random decision forests are an ensemble learning method for that work by creating a multitude of decision trees at training time and outputting the class that is the mean/average prediction of the individual trees for regression. A random forest is a collection of decision tree predictors $h((x; \theta_k), k = 1 \ldots K$ where x represents the observed input (covariate) vector of length p with associated random vector X and the θk are independent and identically distributed (*iid*) random vectors [15].

(9) Neural Networks: Feed-forward neural networks are well known and popular tools to deal with non-linear regression models. We can describe MLP models as a parametric family of regression functions. Multilayer perceptrons (MLP) with one hidden layer have been used for a long time to deal with non-linear regression. If parameters of MLP models are not bounded, for Gaussian noise, the overfitting is strong, even if the number of data is large. The Transfer function is [16]:

$$y_i = f_{ij}(x_i w_{ij}) \text{ ,where}$$

y_i : Output of node j,

w_{ij}: Connection weight between node i and node j.

x_i: Input signal from the node i.

(10) K Nearest Neighbours: KNN is a kind of supervised learning method. Supervised learning infers a function(learner) from a training data T, which is a collection of training examples called samples. Each sample is a pair including an input vector(instance) and the desired output value. After learning from the training set, the learner seeks to correctly determine the output for unseen instances. KNN regressor is based on learning by comparing the given test instances with the

training set. ŷ, the prediction output y of x is the mean of the outputs of its k nearest neighbours in regression [17].

(11) Support Vector Machine: A support vector machine (SVM) is a type of model that is optimized so that prediction error and model complexity are simultaneously minimized. Consider a set of data points, $\{(x_1, y_1), \ldots, (x_k, y_k)\}$, such that $x_i \in R^n$ is an input, y_i is a target output, and k is the total number of examples [18]. The output of an SVM is either a linear function of the inputs, or a linear function of the kernel outputs. In a regression model, our main aim is to decide a decision boundary at 'a' distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

Model Building

We will now let's discuss the steps involved in Model Building:

(1) Splitting the Data: Since we have Preprocessed data, we now need to split the Data into Train, Test and Validation. We first split the data into Train (80%) and Test (20%) and then we split the Train again into Train (68%) and Validation (12%)

(2) Stratification: Since there are very few rows of data with values <=-0.5 in the Financial Distress column it's important that we split data in the same ratio i.e. ratio = values>0.5 / values<=-0.5 which goes into Train, Validation and Test. We can achieve this stratification by creating a column using a Target variable which consists of 0 and 1 where 0 indicates Financial Distress value >0.5 and 1 indicates value <=-0.5

(3) Oversampling: Since there are very few 1's (meaning Financial Distress value <=-0.5) i.e. 136 compared to 3536 0's we need to Oversample, but we just do this for Training data before we split into Validation. This improves the model's performance dramatically. We used a resreg library which oversample data of rare domain (bottom 4 percentile of data) by randomly taking data from the bottom 4 percentile of data. We approximately added 200 more samples to the rare domain

(4) Feature Selection: We used sklearn's SelectKBest for feature selection but contrary to the expectations, there was a decrease in Model's performance. This selection would be before splitting the data

(5) Model Training: We Train the Model with default or configured values for Hyper Parameters for each Model and we evaluate it on the Validation set

(6) Hyper Parameter Tuning: We tune different Hyper Parameters like alpha, eta0, l1_ratio, max_iterations, n_neighbors, kernel, etc. for respective Model's to see if there is any improvement in performance and we observe huge improvements in Model's performance

(7) Stacking: A decent approach for combining results from different models. We picked three best models of all we evaluated and the output of these models are passed into the final estimator which can be used for prediction

Evaluation

We have used different metrics as well as plots for evaluating the model's performance and they are listed below:

(1) R2 score: It's the goto metric for regression models. The closer the value is to 1, the better the model performs. 0 implies the model is no better than predicting each row as mean value

(2) Mean Squared Error: Any regression model can be evaluated based on MSE. Mean Squared error is better than Mean Absolute Error in the sense that it penalizes (since squaring of actual difference between actual and predicted value) for very small differences in actual and predicted values

(3) Max Error: This metric gives information about max error that the model had in predicting

(4) Accuracy Score: Although this is a regression model we classify the actual and model predicted values into 0's and 1's where 1 implies Financial Distress value <=-0.5 and vice versa. We need this metric in evaluating a model since our main goal is to predict if a company is in Financial Distress or not and tuned the Model according to this score as well

(5) True and Predicted value range: It is important that the regression model covers all the values in the range of True data

(6) Distribution Plot: This plot gives us the information about density of predicted and true values. Overlapping Actual, Predicted curves imply that the model is successful in predicting the values similar to actual values with same density

(7) Scatter Plot: In this plot we have predicted vs actual scatter plot and two horizontal, vertical lines at starting -0.5 on both axes. We use this plot in tuning the model such that very few or no scatter plot values are in the top - left corner of the plot. We prioritized predicting values to be <=-

0.5 for actual values since we shouldn't misclassify those that are 1's to 0's as it is important to predict a company which is in Distress

(8) Confusion Matrix: We also used confusion matrix to see correct and misclassifications. We tuned the Model's in a way that we have very few 1's misclassified with a tradeoff by allowing the model to misclassify 0's. We followed this approach since we prioritized predicting Financial Distress over fitting the model which has less error i.e. accuracy and recall over mean squared error

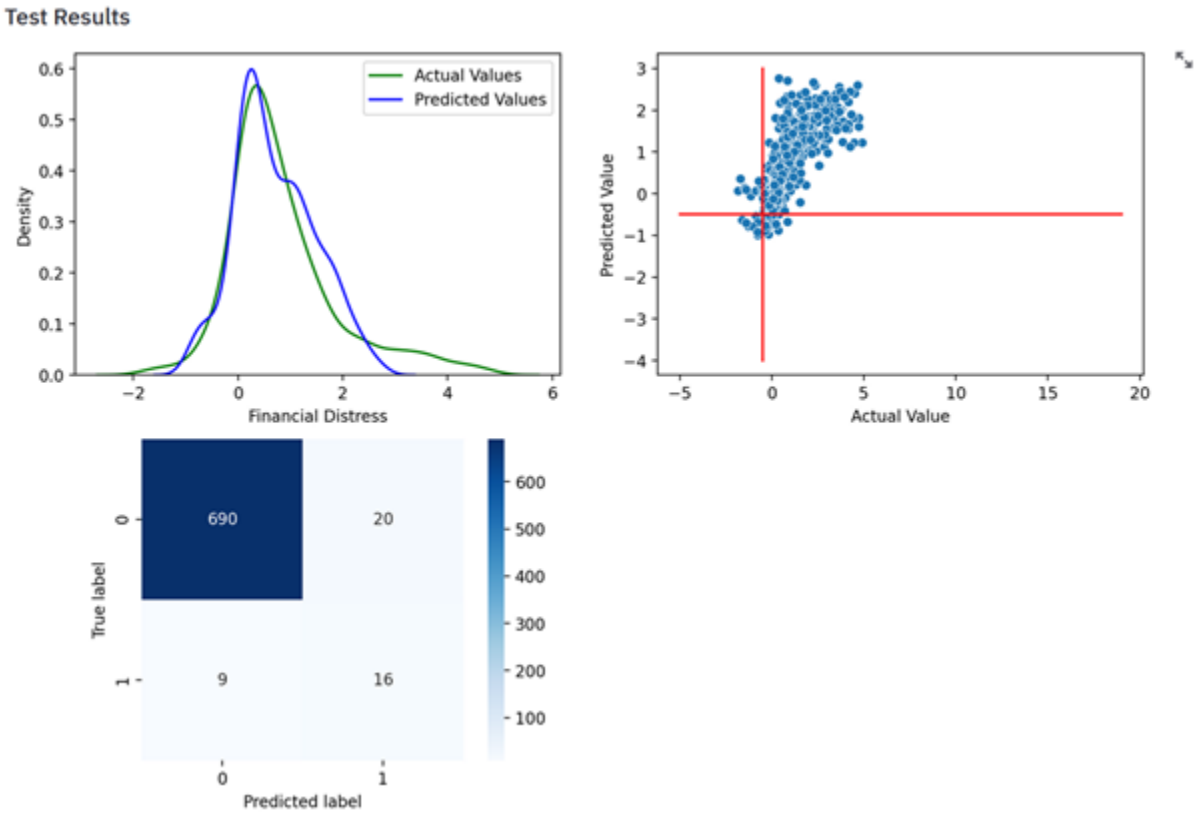
EXPERIMENTS

Explanation of experimental setup

We have used Jupyter notebook to perform Data Preprocessing, Exploratory Data Analysis, Model Training and Model Evaluation. We finally build a Streamlit Dashboard which is interactive and people can have a concise understanding of our experiment. There are different libraries used for this experiment: Pandas, Numpy, Sklearn, Matplotlib, Resreg, etc.

Test results of the proposed method

The proposed method got 96.5% accuracy which is pretty decent. The results for stacking are shown below in the figures (Fig 10a and 10b) taken from our Streamlit Dashboard designed for this project.



(Fig 10a - Output of Streamlit Dashboard)

True Values set range -1.8385 , 4.8878

Predicted Values set range -0.983505741641112 , 2.7638972226194998

Accuracy 96.05442176870748 %

R2 score 0.5524933436120212

Mean Squared error 0.506234342461822

Max Error 3.6626989013759004

Total Train Samples 3234

Total ratio of 0s and 1s in Train 7.125628140703518

Total Test Samples 735

Total ratio of 0s and 1s in Test 28.4

(Fig 10b - Final test results after stacking)

As mentioned above in the method section we have used 12 algorithms and used multiple metrics to find the best model. The below table shows the results for all the models for validation samples.

Models	True values min	True values max	Predicted Values min	Predicted Values max	Accuracy	R2 score	Mean Squared error	Max Error	Total Train Samples	Total ratio of 0s and 1s in Train	Total Test Samples	Total ratio of 0s and 1s in Test
SVR	-1.8385	4.8878	-2.803733	3.566377	94.29%	0.419966	0.656154	3.72318	2748	7.130178	735	28.4
KNN	-1.8385	4.8878	-0.923469	2.054935	96.46%	0.337163	0.749823	4.297567	2748	7.130178	735	28.4
ExtraTree	-1.8385	4.8878	-0.930836	2.43228	96.05%	0.551619	0.507224	3.435078	2748	7.130178	735	28.4
Decision Tree	-1.8385	4.8878	-1.013153	2.988353	96.19%	0.536029	0.524859	3.497223	2748	7.130178	735	28.4
Ridge	-1.8385	4.8878	-1.274709	3.669645	95.24%	0.531041	0.530501	3.457874	2748	7.130178	735	28.4
RandomForest	-1.8385	4.8878	-0.902166	3.143945	96.60%	0.555983	0.502287	3.633013	2748	7.130178	735	28.4
Linear Regression	-1.8385	4.8878	-1.268551	3.655805	95.37%	0.532781	0.528534	3.458356	2748	7.130178	735	28.4
LinearSVR	1.8385	4.8878	-2.358444	3.659183	91.70%	0.218394	0.884179	4.207342	2748	7.130178	735	28.4
SDGRegressor	-1.8385	4.8878	-0.796884	3.145718	96.73%	0.504138	0.560936	3.659088	2748	7.130178	735	28.4
Elasticnet	-1.8385	4.8878	-0.961928	3.119826	96.73%	0.543778	0.516094	3.541096	2748	7.130178	735	28.4
MLPRegressor	-1.8385	4.8878	-1.285981	3.508033	95.24%	0.527418	0.534601	3.518441	2748	7.130178	735	28.4
Lasso	-1.8385	4.8878	-1.273047	3.651262	95.10%	0.530163	0.531495	3.462333	2748	7.130178	735	28.4
Polynomial	-1.8385	4.8878	-1.053153	3.37372	95.92%	0.530163	0.531495	3.462333	2748	7.130178	735	28.4

(Table 3 - Results for all implemented models for validation samples)

Deep analysis/discussion about the results

From table 3, we can observe that many models have performed good with better decent accuracies. However, the target is to find if the company is in financial distress or not but the problem is, there are a very smaller number of 1's (in Financial Distress) than the number of 0's (Not in financial distress). The plan is to track three of the best performing algorithms and try to use stacking to see if the overall results are improved.

Models used for Stacking

The dataset we are using is huge and in order to avoid data overfitting, training, validation and testing samples from the datasets were ensured to be mutually exclusive. The above table (Table 3) has the results for the validation samples.

The first best performing model is Random Forest Regressor with 96.6% accuracy. It has accuracy and best R2 score combined best compared to any other algorithms. Max number of trees is set to 3 after hyperparameter tuning. This is because it is an Ensemble method. Random forest is a strong

modelling technique i.e, it has a large number of relatively uncorrelated trees that outperform a single Decision Tree. Its predicted values range is almost equal to the True values range and the max error is also less than any other models.

The second-best performing model is the Elastic net. It has accuracy of 96.7% and also the R2 score is pretty good. Elastic net is a popular type of regularized linear regression that combines two popular penalties, specifically the L1 and L2 penalty functions. Out of all models predicting the number of 1's (company in financial distress) which are very low, it has the best accuracy even though the training data is very limited.

The third best performing model is the MLPRegressor with 95.24% accuracy. MLPRegressor trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters. The hidden layers are selected as 45 for the neural network model. As it is already mentioned above in the literature survey, the MLPNN is already used and has secured accuracy of 87% whereas our regressor achieved 95% accuracy.

The models are selected not only based on the metrics like accuracy, R2 score but the priority is also to check for the number of correctly predicted 1's as the available data is less for training.

Other Models

Extra tree, Decision Tree Regressor and SDGRegressor are the next top models with almost the same accuracy. Regression tree refers to an algorithm where the target variable is and the algorithm is used to predict it's value. They mostly depend on the features which have continuous values in the data and are well suited for regression. Minimum of 100 estimators and max leaf nodes are used while tuning the tree models to predict for the test data.

The KNN and Linear Regressor are not the best performing models because the r2 score is very bad for the models and the predicted values ranges also do not cover the entire range from the true values range and it's a huge disadvantage for any model.

Other models do not perform great when compared to the above models. SVR, polynomial, Ridge has good accuracy but predicting the 1's is very low even though a lot of tuning and cross validation is done before checking them with the test data. LinearSVR has the worst accuracy of all the models.

After selecting the models for the final testing stacking is done and the results are given above in the test results column.

Stacking the Top performing models

Although the accuracies are pretty decent, we tried a stacking method to see if the model performance improves. Stacking regression is an ensemble learning technique to combine multiple regression models via a meta-regression process. The individual regression models are trained based on the complete training set; then, the meta-regressor is fitted based on the outputs -- meta-features -- of the individual regression models in the ensemble. Basically, it takes all the models given as estimators and tries to get a better accurate model if the selected models are performing well. The selected models are MLPRegressor, Random Forest Regressor and Elastic Net. The results for stacking are shown above (Table 3) in the screenshot taken from our streamlit dashboard for this project.

The first plot in the screenshot (Fig 10a) is the distplot which plots both actual values and predicted values for better visualization. Second is the scatter plot for both actual and predicted. It can show the distribution of values in 2D. The confusion matrix gives clear data. We were able to get a good prediction of the number of 1’s when used stacking rather than any individual model. Also, the predicted number of 0’s is obviously better.

Amount of effort your team has made

Task	Bentic	Sharath	Navneeth	Sumanth
Literature Survey	Yes	Yes	Yes	Yes
Project Topic Survey	Yes	Yes	Yes	Yes
Data Selection	Yes	Yes	Yes	Yes
Data Proposal	Yes	Yes	Yes	Yes
Data Preprocessing	Yes	Yes	Yes	Yes
Feature Selection	Yes	Yes	Yes	Yes
Data Visualization	Yes	Yes	Yes	Yes
Model Selection	Yes	Yes	Yes	Yes
MidTerm Progress Report	Yes	Yes	Yes	Yes
Model Training and evaluation, Hyperparameter Tuning and Model Testing of Linear, Ridge, Lasso and Elastic Net			Yes	
Model Training and evaluation, Hyperparameter Tuning and Model Testing of Polynomial, Lasso, Ridge and Elastic Net Regression	Yes			
Model Training and evaluation, Hyperparameter Tuning and Model Testing of KNN, Neural Networks and SVM		Yes		
Model Training and evaluation, Hyperparameter Tuning and Model Testing of Decision Trees, Random Forest and AdaBoost				Yes
Model Stacking and Streamlit Dashboard	Yes	Yes	Yes	Yes
Project Presentation	Yes	Yes	Yes	Yes
Final Report	Yes	Yes	Yes	Yes

CONCLUSION

Concluding Remarks

Companies may have to terminate operations for diverse reasons. The financial failure (financial distress) of a company affects various participants such as investors, owners, employees, creditors, clients and even the relevant authorities. Within this context, predicting financial failure attracts the interest of researchers. The regression model can give a better accuracy compared to the previous classification models mentioned in the literature survey. Further research can also be done for optimal tuning and better representation of the results for more data and can help to secure the companies.

Thoughts about the project

We worked on the regression model and this is our first experience working on regression. We picked the financial distress problem because it was very interesting and it's an important real life problem. Every one of us in the team had worked on various regression models which gave in depth knowledge and the math behind each model. This helped us to learn more about feature selection and hyperparameter tuning which is very crucial for any model. We have achieved better accuracy when using a stacking model. We got good hands on experience on python and working better with data which improved our data analysing skills.

What we have learnt

Working on this project gave everyone in the team in depth knowledge and the math behind each model. This helped us to learn more about feature selection and hyperparameter tuning which is very crucial for any model. We have achieved better accuracy when using a stacking model.

Challenges and how to overcome them

One of the biggest challenges was the presence of unclean data which we overcame by imputing, and standardizing the data. Another challenge was data visualization since the features had no names and we overcame it by finding the correlation between the features. Another challenge was finding the best models which we overcame by picking the best three models. The last challenge was finding the final estimator for stacking which we solved by picking one model after extensive testing. model.

Future Work

Although we achieved great results we still have to work out Feature Selection and Dimensionality

Reduction since we are getting bad results but there is high correlation among variables and these methods should have worked. Maybe a Time Series Analysis would even yield better results and accurate predictions.

RESPONSE TO THE FEEDBACK

List of Comments by the Instructor in midterm report

Comment 1: It would be better to clarify what exactly you want to predict.

Response: The target variable is **Financial Distress.**

Comment 2: Need better-organized presentation of your approach

Response: We listened to your feedback and modified the way we presented and organized the content well for both Presentation and Report.

REFERENCES

[1] Yi Qu a,b,c, Pei Quan b,c, Minglong Leid , Yong Shi a,b,c,e, "Review of bankruptcy prediction using machine learning and deep learning techniques", 7th International Conference on Information Technology and Quantitative Management (ITQM 2019)

[2] Grice, J.S., Dugan, M.T. The Limitations of Bankruptcy Prediction Models: Some Cautions for the Researcher. *Review of Quantitative Finance and Accounting* 17, 151–166 (2001)

[3] Ashraf, S.; G. S. Félix, E.; Serrasqueiro, Z. Do Traditional Financial Distress Prediction Models Predict the Early Warning Signs of Financial Distress? *J. Risk Financial Manag.* 2019, *12*, 55

[4] Salehi, Mahdi & Abedini, Bizhan. (2009). Financial Distress Prediction in Emerging Market: Empirical Evidences from Iran. Business Intelligence Journal.

[5] Ghodrati, Hadis & Moghaddam, A.. (2012). A study of the accuracy of bankruptcy prediction models: Altman, Shirata, Ohlson, Zmijewsky, CA score, Fulmer, Springate, Farajzadeh genetic, and Mckee genetic models for the companies of the stock exchange of tehran. American Journal

of Scientific Research. 59. 55-67.

[6] Credit Scoring Using Ensemble of Various Classifiers on Reduced Feature Set, Shashi Dahiya1 S.S Handa2 N.P Singh3, Industrija, Vol.43, No.4, 2015

[7] Omelka, Jiří & Beranová, Michaela & Tabas, Jakub. (2013). Comparison of the models of financial distress prediction. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis. 61. 2587-2592. 10.11118/actaun201361072587.

[8] Kologlu, Yunus & Birinci, Hasan & Ilgaz, Sevde & Ozyilmaz, Burhan. (2018). A Multiple Linear Regression Approach For Estimating the Market Value of Football Players in Forward Position.

[9] McDonald, Gary. (2009). Ridge regression. Wiley Interdisciplinary Reviews: Computational Statistics. 1. 93 - 100. 10.1002/wics.14.

[10] Ogutu, J.O., Schulz-Streeck, T. & Piepho, HP. Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. *BMC Proc* **6,** S10 (2012). https://doi.org/10.1186/1753-6561-6-S2-S10

[11] Hans, Chris. "Elastic Net Regression Modeling With the Orthant Normal Prior." *Journal of the American Statistical Association* 106 (2011): 1383 - 1393.

[12] Eva Ostertagová, Modelling using Polynomial Regression, Procedia Engineering, Vol 48, 2012, pp 500-506, ISSN 1877-7058, https://doi.org/10.1016/j.proeng.2012.09.545.

[13] Loshchilov, I. and Hutter, F., "SGDR: Stochastic Gradient Descent with Warm Restarts" 2016.

[14] Santosh Singh Rathore and Sandeep Kumar. 2016. A Decision Tree Regression based Approach for the Number of Software Faults Prediction. 41, 1 (January 2016), 1–6. DOI: https://doi.org/10.1145/2853073.2853083

[15] Segal, Mark. (2003). Machine Learning Benchmarks and Random Forest Regression. Technical Report, Center for Bioinformatics & Molecular Biostatistics, University of California, San Francisco.

[16] H. A. A. Aqlan, S. Ahmed and A. Danti, "Death prediction and analysis using web mining

techniques," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, pp. 1-5, doi: 10.1109/ICACCS.2017.8014715.

[17] Yunsheng Song, Jiye Liang, Jing Lu, Xingwang Zhao, An efficient instance selection algorithm for k nearest neighbor regression, Neurocomputing, Vol 251, 2017, pp 26-34, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2017.04.018.

[18] Flake, G.W., Lawrence, S. Efficient SVM Regression Training with SMO. Machine Learning 46, 271–290 (2002). https://doi.org/10.1023/A:1012474916001