Prudential Insurance Data Science Report

Project Description

The prudential data set is taken from the current Kaggle Competition. The detailed description about the data set can be found at https://www.kaggle.com/c/prudential-life-insurance-assessment

Features Description

Data set consists of 127 features and 59381 observations. The features are divided into broadly 4 categories namely, classification, discrete, dummy and continuous features. Response feature is the feature to be predicted and Id gives a unique id to each observation.

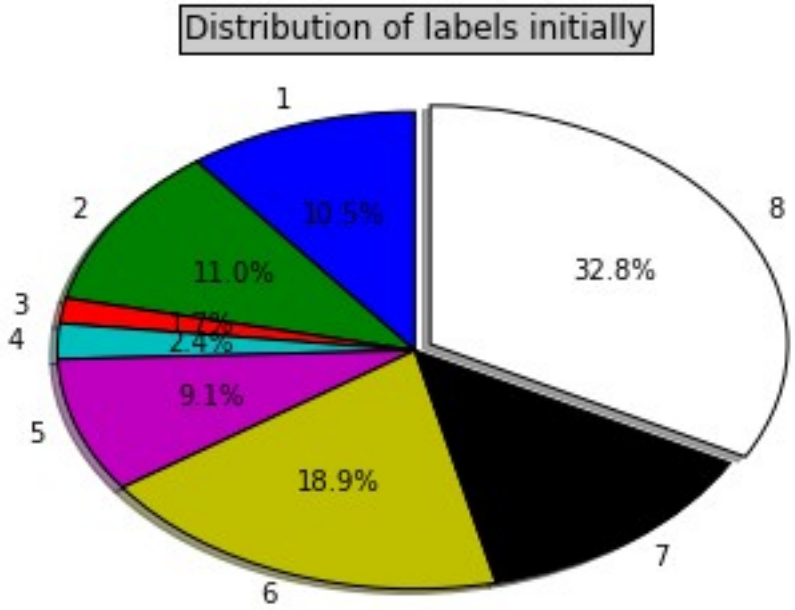
Below is a small description of the kinds of variables

Features	Description
Product_Info_1-7	A set of normalized variables relating to the product applied for
Ins_Age	Normalized age of applicant
Ht	Normalized height of applicant
Bt	Normalized weight of applicant
BMI	Normalized BMI of applicant
Employment_Info_1-6	A set of normalized variables relating to the employment history of the applicant.
Insured Info_1-6	A set of normalized variables providing information about the applicant.
Insurance_History_1-9	A set of normalized variables relating to the insurance history of the applicant.
Family_Hist_1-5	A set of normalized variables relating to the family history of the applicant.
Medical_History_1-41	A set of normalized variables relating to the medical history of the applicant.
Medical_Keyword_1-48	A set of dummy variables relating to the presence of/absence of a medical keyword being associated with the application.
Response	This is the target variable, an ordinal variable relating to the final decision associated with an application

Description of features of the DataSet

Label/Target Feature Description

Response variable has 8 categories. Distribution of response variable in each of these categories is show below in the pie chart.

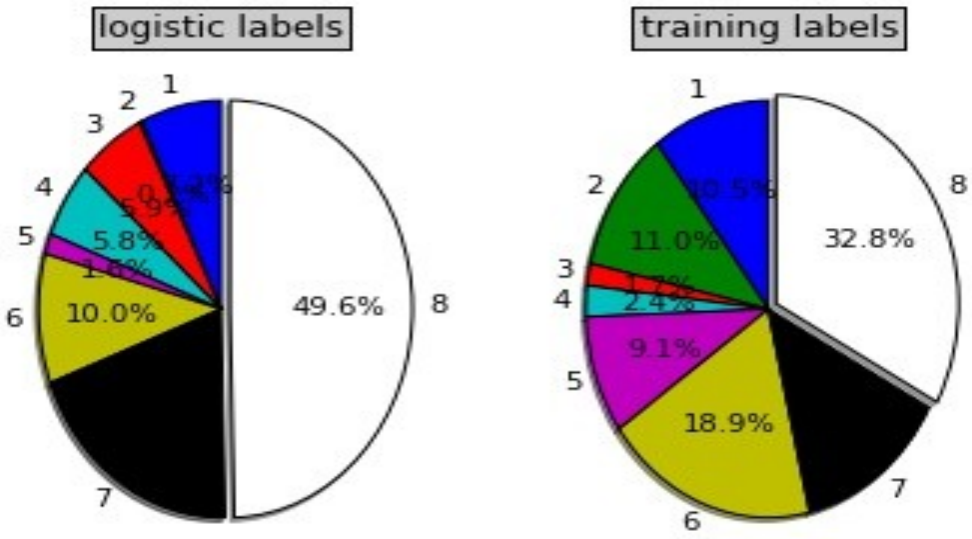


Summary of the Methods Tried

All these methods have been tried by taking 80% of the randomly selected training set for training and then predicting on remaining 20% of the set taking it as test set

Logistic Regression
Logistic Regression is one of the most prominent methods used in for classification problems. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. The outcome is a probability to be true for each of the possible outcomes/labels. The label with highest probability is choses as the final result.

The accuracy achieved with this method was **0.5037 or 50.37%** and the **f1Score was 0.4653**.

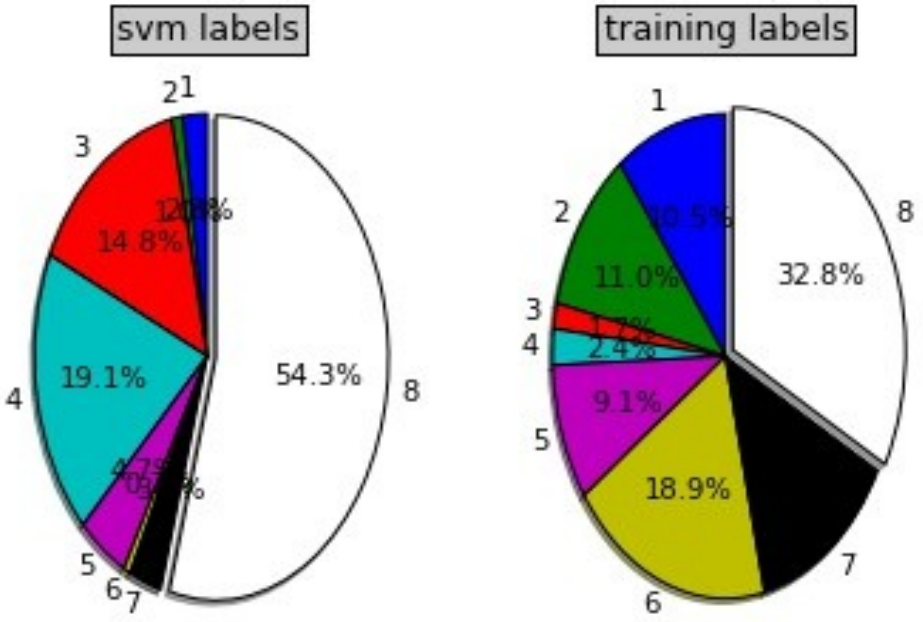


The above diagram shows the distribution of labels on the training data set vs distribution of labels predicted on the test data set as predicted by the logistic regression algorithm.

Support Vector Machines

In machine learning , support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non probabilistic classifier . An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

The accuracy achieved with this method was **0.4014** or **40.14%** and the **f1Score was 0.3254** on using SVM with a linear kernel.



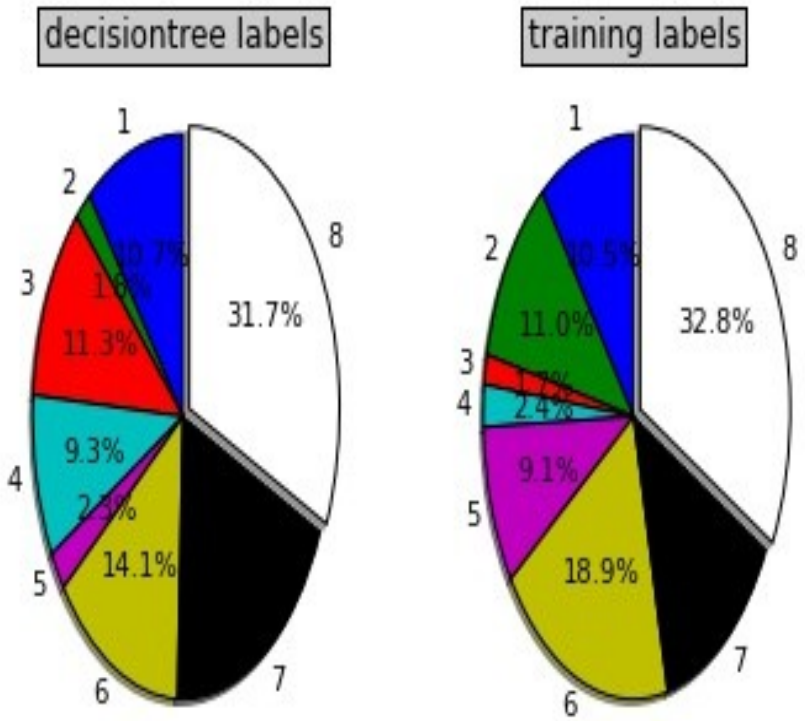
The above diagram shows the distribution of labels on the training data set vs distribution of labels predicted on the test data set as predicted by the SVM algorithm.

Decision Trees

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represents classification rules.

Each feature is ranked upon by using Information Gain as a parameter and the data set is then trained upon by classifying it on these features in order of the values of Information Gain for these features.

The accuracy achieved with this method was **0.4503** or **45.03%** and the **f1Score was 0.4508**.

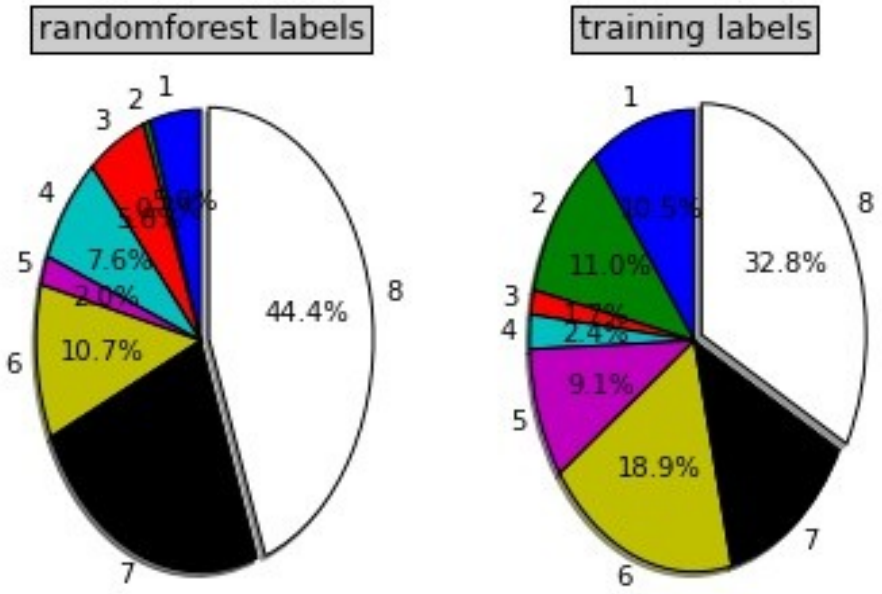


The above diagram shows the distribution of labels on the training data set vs distribution of labels predicted on the test data set as predicted by the decision tree algorithm.

Random Forest

Random forests is a notion of the general technique of random decision forests, that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

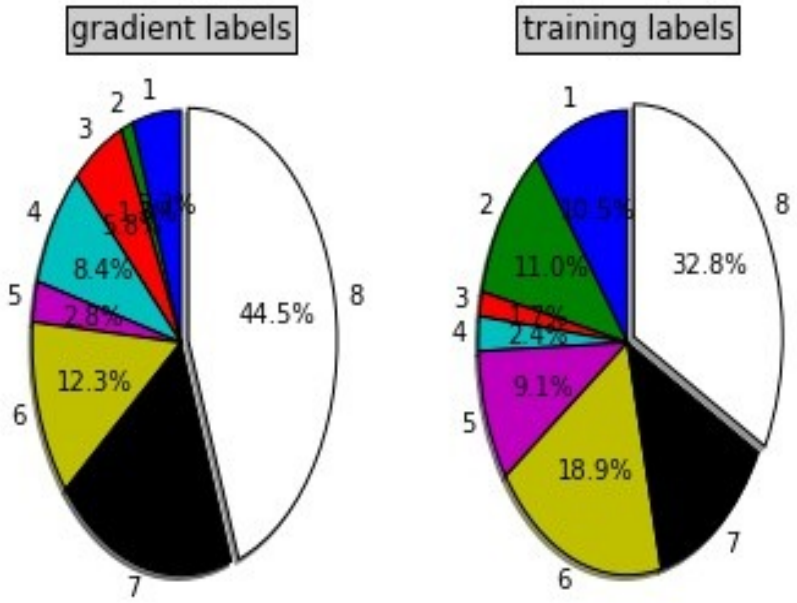
The accuracy achieved with this method was **0.5760** or **57.60%** and the **f1Score was 0.5465**



The above diagram shows the distribution of labels on the training data set vs distribution of labels predicted on the test data set as predicted by the random forest algorithm.

Gradient Boosting
Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

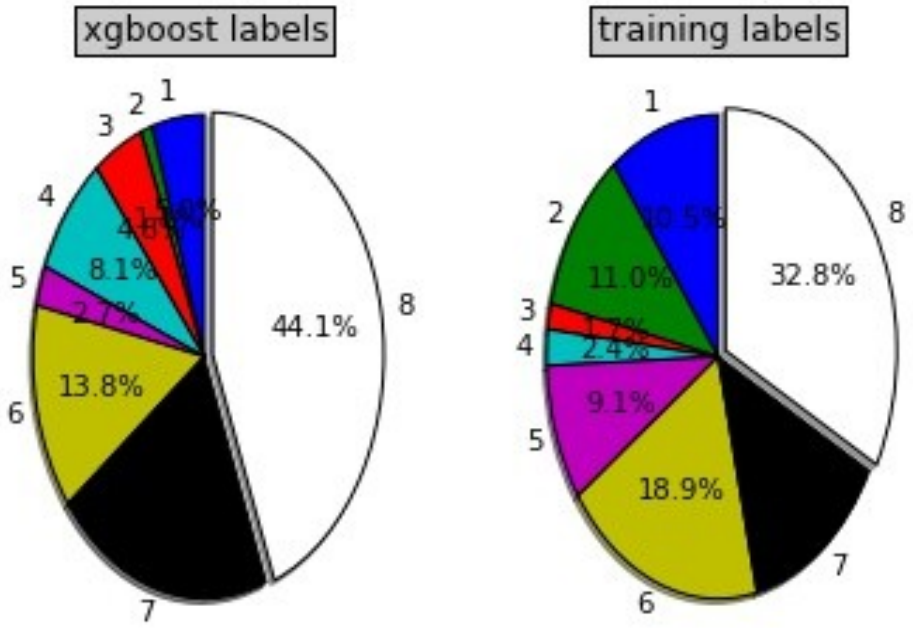
The accuracy achieved with this method was **0.5913** or **59.13%** and the **f1Score was 0.5643**



The above diagram shows the distribution of labels on the training data set vs distribution of labels predicted on the test data set as predicted by the gradient boosting algorithm.

XGBoost
XGBoost is short for “Extreme Gradient Boosting”, where the term “Gradient Boosting” is proposed in the paper *Greedy Function Approximation: A Gradient Boosting Machine*, by Friedman. XGBoost is based on this original model. The algorithm made its debut during the Higgs Boson competition at Kaggle and has been one of the most successful algorithms since then.

The accuracy achieved with this method was **0.5987** or **59.87%** and the **f1Score was 0.5706**



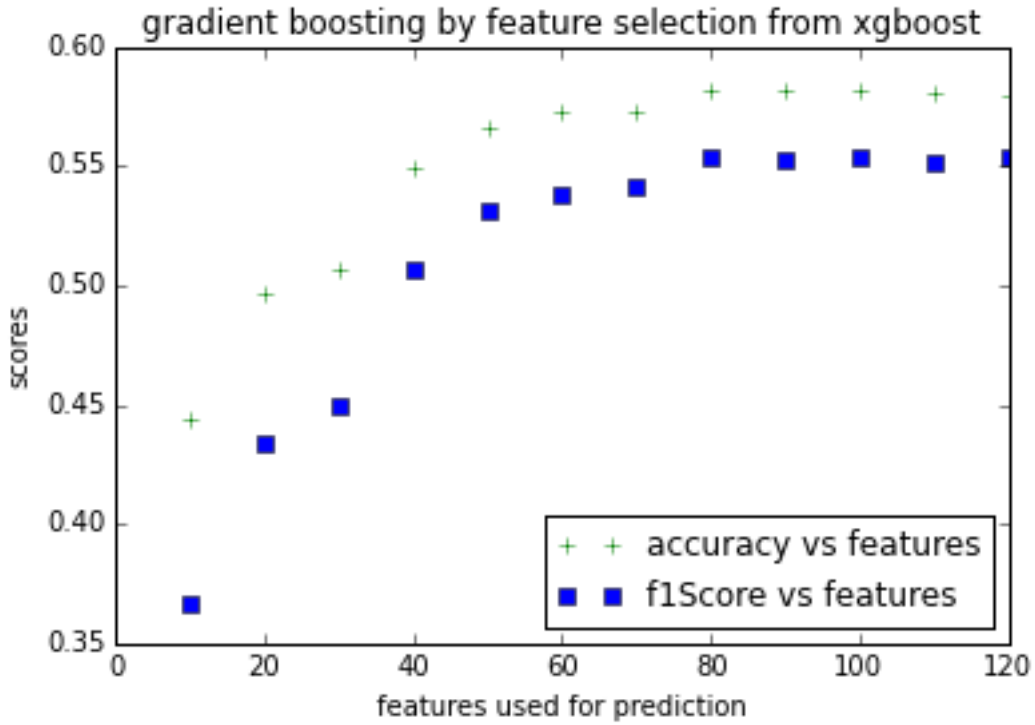
The above diagram shows the distribution of labels on the training data set vs distribution of labels predicted on the test data set as predicted by the XGBoost algorithm.

Since the f1Score by gradient boosting and xgboost were quite close to each other hence models from these algorithms were used to predict on the data set. Out of the two models the one by gradient boosting scored a better score of 0.56370 whereas the one by xgboost scored 0.54124 on the test set on Kaggle.

Improvements Tried

1. Feature Selection through XGBoost

XGBoost algorithm can also be used for feature selection since it provides a score to each of the feature being used for prediction. One of the obvious methods thus was to perform a feature ranking with the help of XGBoost and then using Top features for prediction using gradient boosting. Below attached is a plot to show the variation of accuracy and f1Score with the selected features.



As can be seen from the graph above calculated f1Score plateaus at a value of 0.556 and almost all features are used. Thus the idea of selecting features first through XGBoost and then using those features in gradient boosting did not yield any benegit over the previous method.

2. Prediction by leaving out the continuous variables

Another method which was tried was to predict the response variable by only taking into consideration the categroical , discrete and dummy variables. The intuition behind the method was that since the end task is to do multi-class classification, hence more value can be obtained from non-continuous variables. However, the method was not a success.

3. Feature selection through PCA

Another method tried was to perform a feature selection using PCA and then predicting using gradient boosting algorithm. The intuition behind the method was to if the best features can be obtained from PCA and then prediction is done using those features only it might lead to a better result. However, this method was also not a success.