TASK 1

The dataset chooses is a Loan default Prediction Dataset
https://www.kaggle.com/roshansharma/loan-default-prediction

Summaries the existing contributions

In this customers are the considered who have their data stored in the dataset of the bank which include their personal and the financial detail. It shows how much time the customers are paying the EMI and will finally pay full loan amount or not.

In the existing the contribution (Kernels) the data is loaded in python and then finding of null values is done then whatever the null values were found out it was replaced. After that there is attribute in that as CNS Description is not that clear so made some change so its instances look clear. Similar is done for date as the Disbursal date and Date of birth is not require jus the age will be a easy way through then there were some visualization s which should what type of data is really present in that dataset. The training dataset is evaluated and many factors are removed. Different kernels used different approaches to do some cleaning till here

Some used loops some made the clear data one by one primary and secondary are two factors which are in these so both of them were also cleaned and then their distribution was seen through plots. In other this attribute was not even considered and is dropped out so mane the dataset less complex.

In this some kernels are only showing the preprocessing part in which for future work was considered to predict the load default.

After the preprocessing the kernel has applied normalization, standardization and min-max classifier then the splitting is one on which cross validation is applied but different coders have different folds . In one of those only splitting is done then string or categorical data is converted in to numerical on which different models were applied as Logistic Regression, Decision Tree Classifier, K Neighbors Classifier, Ada Boost Classifier, Bagging Classifier, Random Forest Classifier, XGB Classifier etc and by this it was tested on the test dataset and the final loan default file was formed for customers. For different kernels different models were considered as best.

When the best model was found out they applied then on test and found out the new loan default list which was formed by the highest accuracy model.

Formulating a plan

From the above mentioned contribution on Loan default prediction we can see that there were very necessary thing that some kernels did as plotting everything, for sure it is a better option to understand the attributes well but this is not that much required and time consuming also.

And we can see some attributes were not required t be processed but then also they were processed and they were dropped also so it was a gap in that.

After cleaning the data processing involved standardizing and normalizing which really skews our data and make it better and faster so that models can be applied on that.

Cross validation was applied with putting some thought in it and then the models which were applied on that were precise giving a reason why they are to be applied.

When the best model was selected it was applied on the dataset (test)to achieve a new loan list which was not that accurate and had some glitches in getting values in that, as the data was 0,1 but it was showing result in points or all 0 or all 1.

By looking into all this a plan was made in which some specific actions were included and added to have a better result.

Solution

For developing a solution packages were imported as nump, panda, seaborn, Matplotlib, sklearn-tree, linear model, ensemble, metrics and model selection. Then data was loaded into the jupyter notebook using panda as.csv file. Training and the test dataset were viewed and what all attribute and instances are there were seen. By using rain .info data type of the data was viewed then if there are any null values or not was found out, in the employment type there were null values which were covered by putting the employment as “unemployed”

A correlation heatmap was coded on train data to view the relation between attributes .As date of both was in the form of date and it is not accepted while cross validation so we changed it into age and further will drop this column.

Further avg acct age and credit history length were also in the frm of yr-mn which was converted all in months.so that a clear form of data can be seen and easy to evaluate.

PERFORM_CNS.SCORE.DESCRIPTION is the attribute which makes a problem when we are splitting the data so we came back to the preprocessing part and started encoding it as all the data was mixed and complex so it was changed into numerical form which stated it as easier way.

We can see that Disbursal date was also in the form of date which we have to convert so encoded that to extract the age of disbursal.

After all this final data frame was creating which had involved dropping some columns which were unnecessary and occupying scope which does not had anything to do with the results also

In one frame the data to be predicted is extracted out and the data was refined for the splitting.
The label encoder was applied which converted categorical data into numerical and dummies were also created for the attributes which had sting or not a pure numerical data.

The data frame was now split into training and the test data which states that `(test_size=0.5,stratify=y,random_state=90)` these basis were on which data was split. In statistics and machine learning we usually split our data into two subsets: training data and testing data (and sometimes to three: train, validate and test), and fit our model on the train data, in order to make predictions on the test data. When we do that, one of two thing might happen: we overfit our model or we underfit our model. We don't want any of these things to happen, because they affect the predictability of our model—we might be using a model that has lower accuracy and/or is un-generalized.

After splitting the length of our train data was 6609 then 5 fold cross validation was put up as a loop which will be further used when data accuracy is to be tested. A loop was again created for precision, recall and AUC score which makes the code look clearly what is really happening and this does not make the model application complex.

Considered models in this were Stochastic Gradient decent, Decision Tree, Random Forest, Logistic regression on which bagging was applied.

Stochastic Gradient decent also known as incremental gradient descent, is an iterative method for optimizing a differentiable objective function, a stochastic approximation of gradient descent optimization. This is the reason it is considered to be applied on this dataset.

Decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions, which will help in telling what it is good for or what other factors should be considered.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. As it is a better model we can say that this model has to be applied to get the accuracy and then its comparison is seen.

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. As the output table is in the form of binary so this model is applied.

Bagging/Boosting helps to decrease the variance and increased the robustness of the model. Combinations of multiple classifiers decrease variance, especially in the case of unstable classifiers, and may produce a more reliable classification than a single classifier.

Then model are applied and their accuracy is obtained after which the precision, recall and auc score is calculated which was represented in graphs which gave a very nice visual representation if the values and the changes.

When all the models were tested they were compared which showed that Random forest is the best model for loan data which was further visualized and at the conclusion we can easily tell the best model by seeing the graph.

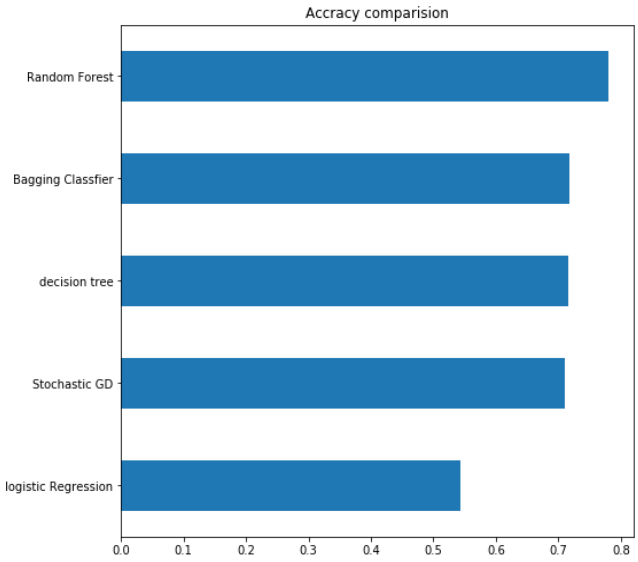
local evaluation and analysis

The models that were compared were Stochastic Gradient decent, Decision Tree, Random Forest, Logistic regression on which bagging was applied. Gradient decent have the accuracy of .71and the score of .33. For Decision tree accuracy was .716 and score .22.for Logistic Regression accuracy was obtained as .54 and with the score of .37 on which bagging was applied which gave the result of.718 and the score of .30 and Random forest was applied which gave the highest accuracy of .78% and score oh .33

I accuracy was seen Random forest is obtained as the best result (model)but on the basis of score Decision is considered as the best.

But for our data as it is categorical we will consider Random forest to achieve the Loan default predicted list.

Reflect on the results



As we can see from the graph given above our best model is Random Forest and with not that much of a difference in DT and SGD as the second best. When this result is compared to the kernel we are obtained very different results and considering everything. The environment, the data, preprocessing steps, cross validation, Splitting as well as the model chosen to be applied.

Some had ADA boost as the best some had bagging and some even has logistic which is considered as the worst model of all in my experiment.

Preprocession plays the most important part of the having the better results model get applied easily but analyzing and processing is the hardest part.

The kernel formed had some gaps also which were fulfilled by me and other kernels. In this experiment conduct there can be more sklearn preprocessing which will help us receive a better result.