CO 544 - Machine Learning and Data Mining

Project (Group 09)

Machine Learning and Data Mining to solve a classification problem

Members	Reg No
M.M.M. Aslam	E/15/021
I.U. Ekanayake	E/15/092
S. Suhail	E/15/348

Project Overview

Data-set

The Credit approval is one of the critical things for a bank to handle since most of the applicants are not in the approval end or the rejection end, where borderline applicants should have to be evaluated properly. Since a pitch from the applicant or a discussion could lead to a miss judgement depending on the experience of the evaluation officer.

In real world there are few different levels of credit cards and approving each at given level plays a critical role of motivate the customers to spend more, and retain in with the company. The initial rejection without a justification would give a bad image about the company or the bank and in long run it affects the name of the bank which is an intangible asset.Therefor, for this project the the credit approval data-set from UCI(cite) database has been used.

Methodology

Predicting the Successful approval or not could be the final task but to achieve that from the raw data set there was a step by step approach.

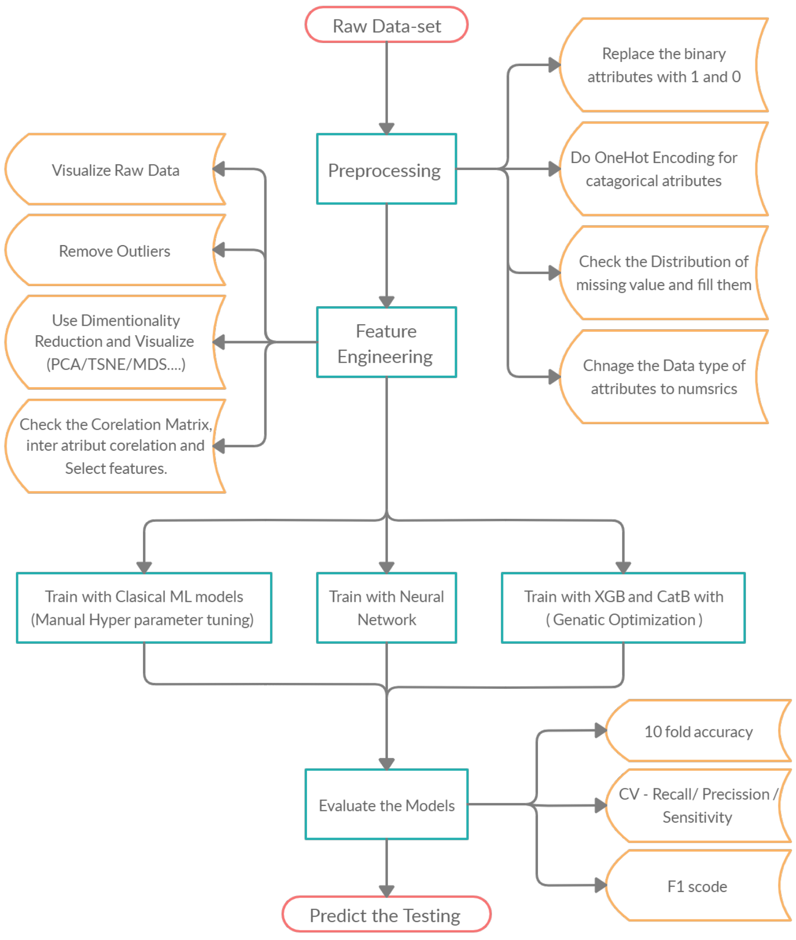
Raw Data-set
Preprocessing
Replace the binary attributes with 1 and 0
Do OneHot Encoding for catagorical atributes
Check the Distribution of missing value and fill them
Chnage the Data type of attributes to numsrics
Feature Engineering
Visualize Raw Data
Remove Outliers
Use Dimentionality Reduction and Visualize (PCA/TSNE/MDS....)
Check the Corelation Matrix, inter atribut corelation and Select features.
Train with Clasical ML models (Manual Hyper parameter tuning)
Train with Neural Network
Train with XGB and CatB with (Genatic Optimization)
Evaluate the Models
10 fold accuracy
CV - Recall/ Precission / Sensitivity
F1 scode
Predict the Testing

Figure 1: Methodology.

Tools and Software

Initially PowerBI and matplotlib in python were used to visualize the data .In-addition the main platform used to create the models were python with (SKlearn, Tensorflow, Pandas, Keras, Numpy) and weka while statistical analysis handled by R.

Data Preprocessing

Encoding Data

The attributes in a data-set have different type of entries (Floats, Numbers, Categorical, binary data) to train them all attributes should change to numeric. Therefor, the binary attributes were encoded as (1 and 0) and multi-categorical attributes were encoded using "ONE-HOT ENCODING". This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value. If the integer encode uses to train then it tries to miss-lead the model and create a relationship between the variation of integers which doesn't exist at all in some cases. After One-Hot Encoding the 15 attributes were distributed to 38 attributes.

Ex: A4: (u, y, l, t) => A4_u, A4_y, A4_l, A4_t

Missing Value Handling

One of the most important things in prepossessing is handling the missing values, where first have to have an clear understanding about how the missing values have been spread throughout the data set.

In this step, missing values have to be handled based on their distributions to achieve a reasonable accuracy. Here, to confirm the randomness of missing values,Little's MCAR test was performed. If the potential bias due to missing data depends on the mechanism causing the data to be missing, and the analytical methods applied to amend the missingness. [16]The chi-square test of MCAR for multivariate quantitative data proposed, which tests whether there exists significant difference between the means of different missing-value patterns.

Table 1: Little's MCAR results

Name	Value
Chi.Square	706.30
degree of freedom	185
P.value	0
Missing patterns	8

Based on the Little's MCAR test results as shown in Table 1, the missing values were considered to be completely random as the "p" value equals zero. Accordingly, substituting missing values with a constant will drop the accuracy and bias the prediction process. Therefore,the K Nearest Neighbor Imputer algorithm was used in this work to fill the missing values.

Feature Engineering

Remove Outliers

Outliers are unusual values in your data set, and they can distort statistical analyses and violate their assumptions therefor, the outliers have properly identified and removed from the data set. Since in categorical data its impossible to find outliers the continues numerical data has been used for this.

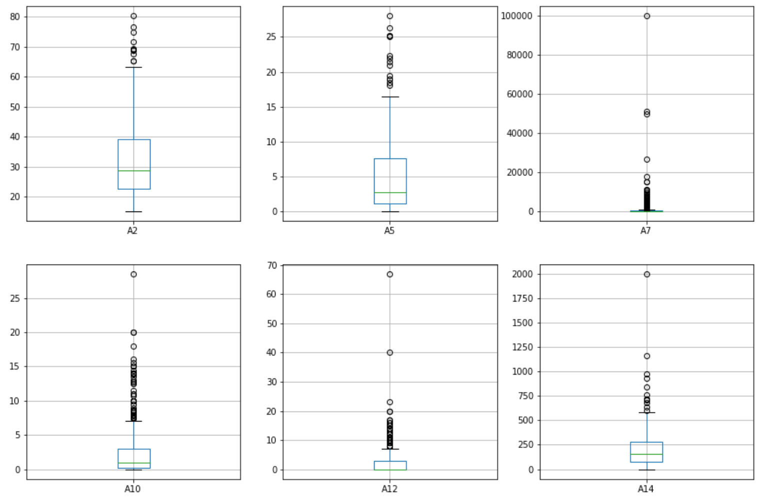

Figure 2: Outliers identified by a box-plot.

Visualization

Initially in visualization the distribution of the 15 base attributes with the Class=”A16” has been considered. As it shows in the Figure 3 most of attributes are well distributed among 2 classes. when consider ”A8” and ”A11”. those attributes have a clear bias which helps a lot in classification.

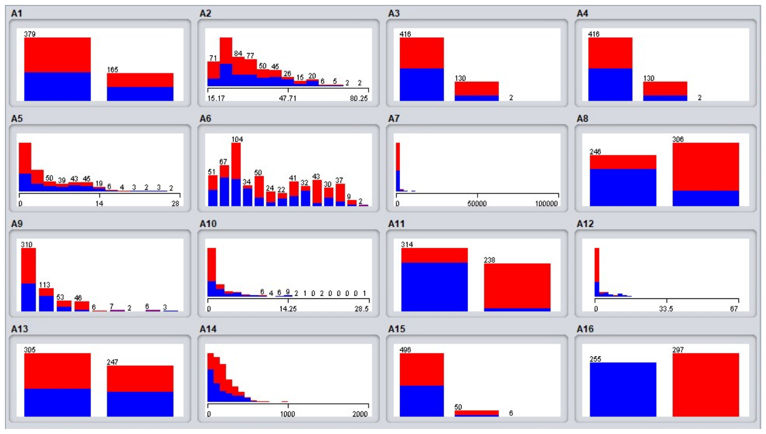

Figure 3: Visualization of 16 features.

Visualize with dimensional reduction

Here we have considered, Multidimensional scaling (MDS), Spectral embedding for non-linear dimensionality reduction, Locally Linear Embedding (LLE), Isomap Embedding, TSNE and PCA. The figure 4 clearly hows the distribution among the highest variant features of each manifold.

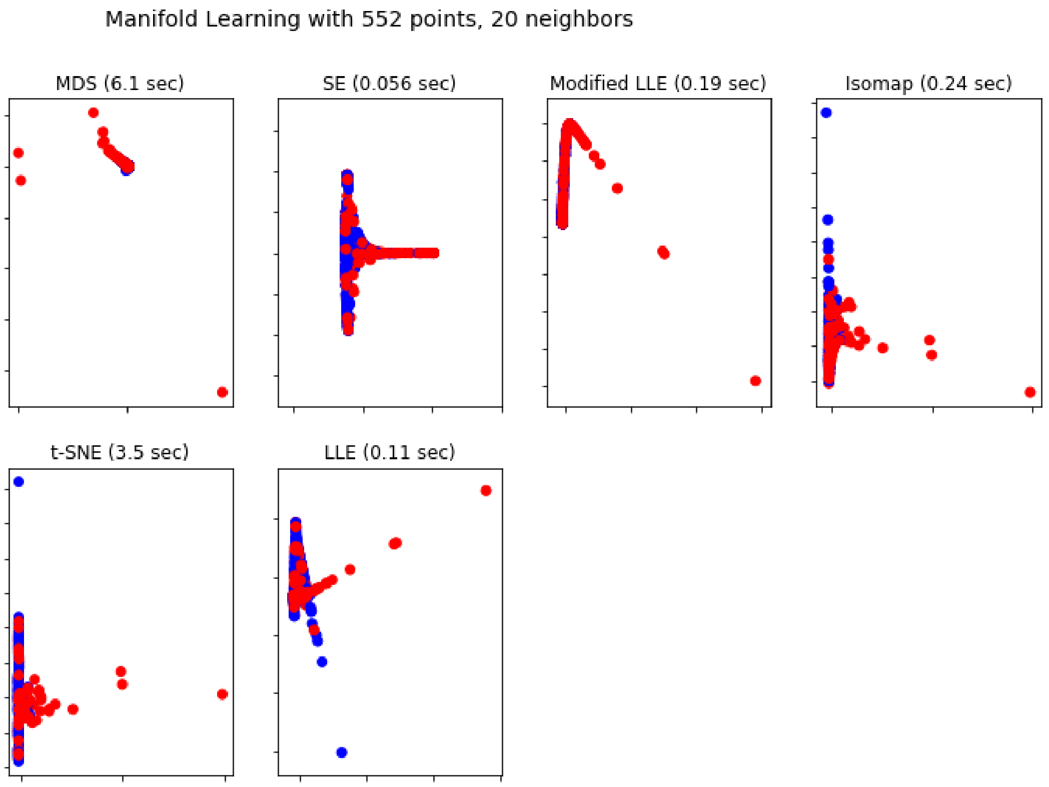

Figure 4: Manifold Visualization.

Apply PCA

Thereafter, PCA has applied to the data set and created 8 variables with that. the highest distributed variables are plotted down below in Figure 5.

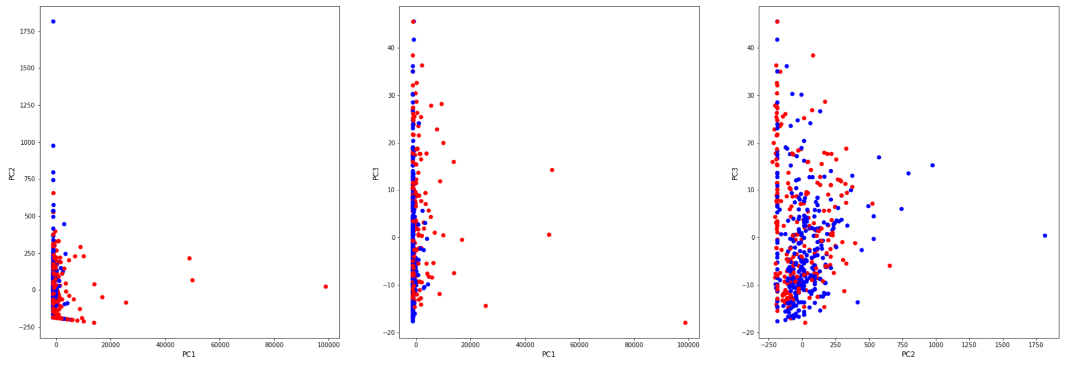

Figure 5: PCA attributes Variation.

The correlation for the class ”A16” and for each PCA attributes has shown below in a heat map. When applying the PCA the one-hot encoded data has been used to reduce the error.

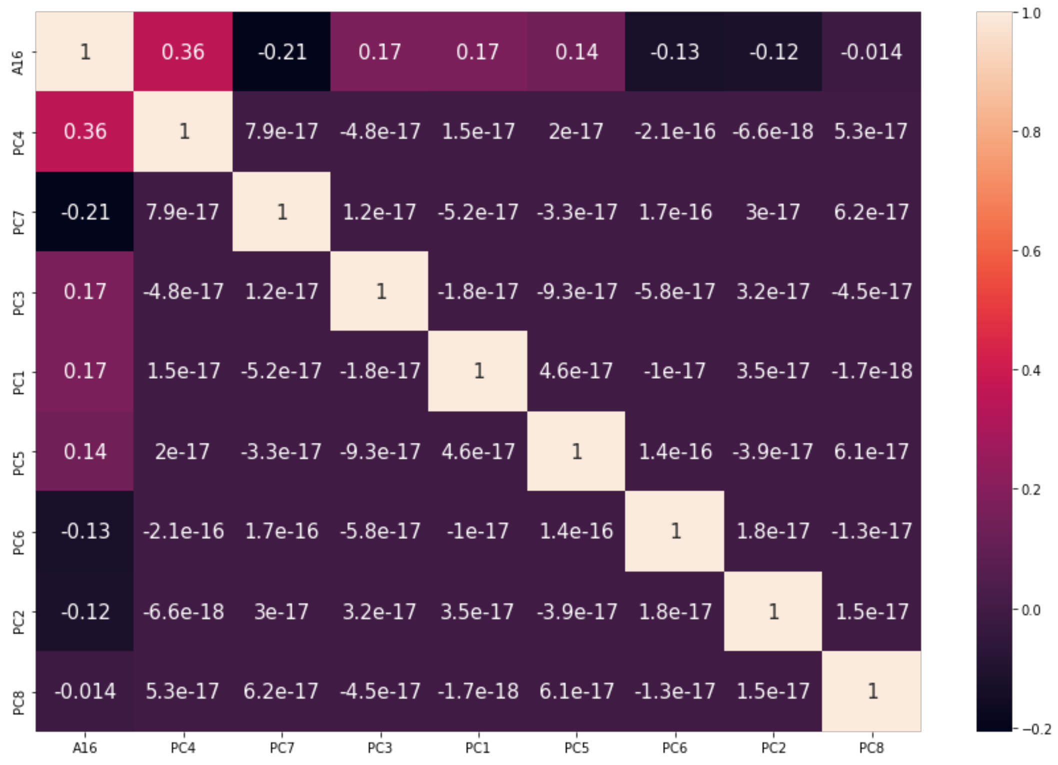

Figure 6: Heat Map of PCA attributes correlation.

Apply TSNE

After applying PCA it was unclear that now the data has been spread throughout the data set which led to apply TSNE with 76 nearest neighbors, for 552 instances which gave a mean sigma of 14.58 and KL divergence in 50th iteration with early exaggeration of 42.81 and after 100th iteration it dropped to 0.3049.

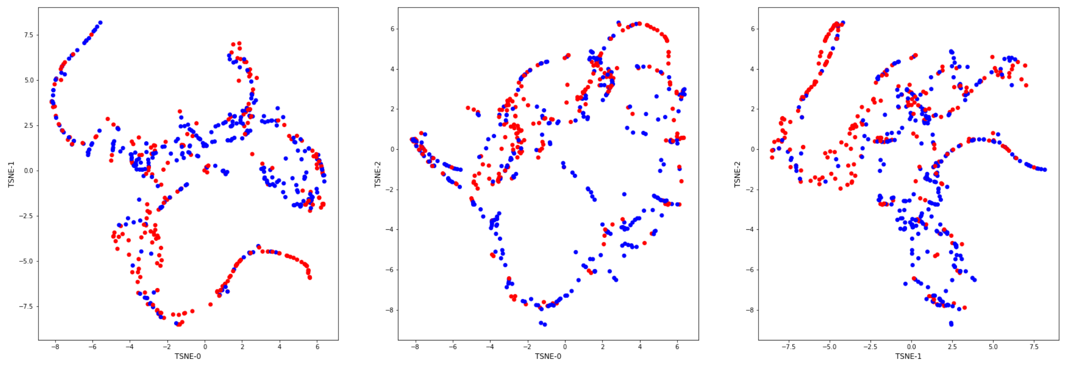

Figure 7: TSNE attributes Variation.

Feature Selection

When it comes to feature selection there were 2 data-sets, one with the one-hot encoded original data set, and the data-set created from PCA. Therefor, for different algorithms we used both data-sets and same as in Figure 6 the Correlation for the original data-set also have considered which has shown down below

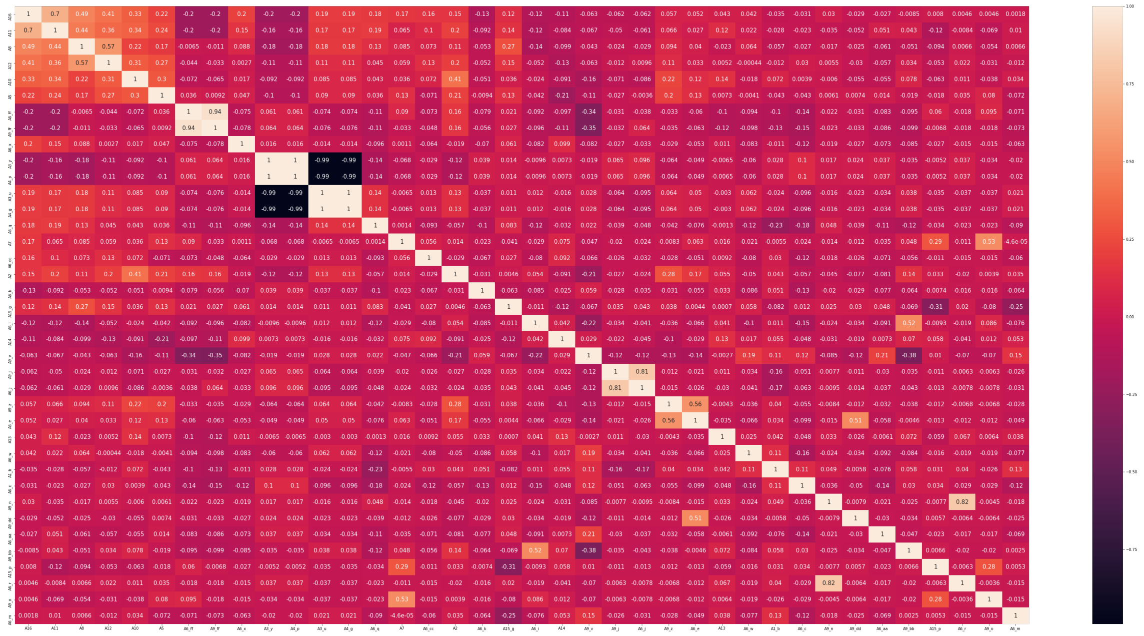

Figure 8: Heat Map of original encoded attributes correlation.

Then, by considering the above matrix and the domain knowledge about each feature from the UCI repository the features to learn has been selected. Moreover, for the algorithms like Gaussian NB and SVC with RBF kernel the PCA data set was used. which makes it quick to learn since almost all the information has concentrated on 8 attributes other than learning from 38 attributes.

Model Training

Classical Models

Initially 11 different classification models were training a Logistic Regression, KNN Regression, SVC with a Linear kernel, SVC with RBF kernel, Gaussian NB, a Decision Tree Classifier, Random Forest Classifier, XGB Classifier, Extra Trees Classifier and an Ada Boost Classifier and Cat Boost Classifier. Then the data set has been divided into 3 parts as 70% Training data, 15% cross validation data and 15% test data randomly. Thereafter,by checking co-relation matrix and by domain understating top features were selected from 2 sets from classical feature creation and PCA. Then the models were optimized for the Training data set and has done the initial accuracy and error test from the cross-validation set and tested. After, by repeating the same process with changing the cross-validation set and training set above models were trained for 5 times by fine tuning the hyper parameters.

Table 2: Accuracy of each algorithm with PCA-Dataset

Algorithm	Training Accuracy	cv Accuracy	Testing Accuracy
Decision Tree Classifier	87.65%	85.31%	83.83%
Random Forest Classifier	95.88%	82.57%	81.61%
XGB Classifier	85.12%	87.12%	86.02%
Extra Trees Classifier	96.83%	84.21%	81.61%
Ada Boost Classifier	84.17%	81.54%	80.88%
KNN	84.49%	87.21%	86.02%
Cat Boost Classifie	86.7%	87.10%	85.99%
SVC Linear	81.01%	83.66%	83.86%
Logistic Regression	83.07%	85.85%	86.02%
SVC RBF	84.81%	84.90%	83.08%
Gaussian NB	81.05%	82.34%	83.83%

As it shown in the above table Decision Tree Classifier, XGB Classifier, KNN Classifier, Cat Boost Classifier and SCV with RBF kernel are working well and Random Forest Classifier and Extra tree classifiers are over fitted to the data set. Therefor by pruning the trees its possible to have higher accuracy. These Algorithms were kept to compare with the rest of the training models.

Table 3: Accuracy of each algorithm with Full Data-set

	Training Accuracy	Testing Accuracy	Testing Precision	Testing Recall	Testing F1
Logistic Regression	84.20%	84.94%	90.48%	75.00%	82.01%
K Neighbors	77.98%	67.47%	66.18%	59.21%	62.50%
SVC Linear	89.63%	84.34%	86.76%	77.63%	81.94%
SVC RBF	98.19%	55.42%	54.55%	15.79%	24.49%
Gaussian NB	83.68%	81.33%	90.91%	65.79%	76.34%
Decision Tree	91.19%	86.75%	87.50%	82.89%	85.14%
Random Forest	98.45%	85.54%	90.63%	76.32%	82.86%
XGB Classifier	86.53%	89.16%	90.28%	85.53%	87.84%
Extra Trees Classifier	100.00%	84.94%	91.80%	73.68%	81.75%
Ada Boost Classifierr	95.34%	82.53%	84.06%	76.32%	80.00%
Neural Network	90.50%	89.52%	76.36%	89.36%	82.35%
Cat Boost	88.30%	86.90%	81.02%	88.13%	84.31%

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as "success", High precision relates to the low false positive rate. We have got the highest of 0.9180 precision in Extra Tree Classifier.

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - Success. We have got highest of 0.8936 precision in the Neural Network.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

$$F1\ Score = 2*(Recall * Precision) / (Recall + Precision)$$

When consider the models trained, the highest F1 score has been obtained from XGB Classifier of 0.8784.

Neural Network

The Multi-class Neural Network module to create a neural network model that can be used to predict a target that has multiple values. A neural network is a set of interconnected layers, in which the inputs lead to outputs by a series of weighted edges and nodes. The weights on the edges are learned when training the neural network on the input data. The direction of the graph proceeds from the inputs through the hidden layer, with all nodes of the graph connected by the weighted edges to nodes in the next layer. To compute the output of the network for any given input, a value is calculated for each node in the hidden layers and in the output layer. For each node, the value is set by calculating the weighted sum of the values of the nodes in the previous layer and applying an activation function to that weighted sum.

Applying neural network is one of the hardest part since the hyper parameters should have to be tuned but understanding the effect from those for each architecture of the network made it harder.

Initially we tried 2 different models, One is artificial neural network, with 3 hidden layers for the original data set. The number of neurons for each hidden layer was tuned by training the model over and over again but to confirm the optimal value for each model of the network we had to tune every model separately and had to make-sure it goes for the global optimal of it self other than stuck in some local optima.

When Consider the full encoded data set it has been trained by using Keras as the platform to design the neural network. Ultimately the Testing scores are,

Table 4: Confusion matrix for Artificial neural network with full data-set

Name	Training Values	Testing Value
Accuracy	90.50%	89.52%
Precision	79.98%	76.36%
Recall	92.13%	89.36%
F1 score	85.62%	82.35%

when consider the PCA data set its’ accuracy and other measures we low compared to the original data set which implies that the informations were low in PCA data-set compared to the original data-set.

when consider the PCA data set, its accuracy are high in training and low in testing.Moreover, other measures were low compared to the original data set which implies that the informations were low in PCA data-set compared to the original data-set which makes it impossible to achieve a higher F1 score or an accuracy from PCA data set compared to the original one.

Train XGB and CatB with Genetic Optimization

The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population ”evolves” toward an optimal solution. You can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. The genetic algorithm can address problems of mixed integer programming, where some components are restricted to be integer-valued.

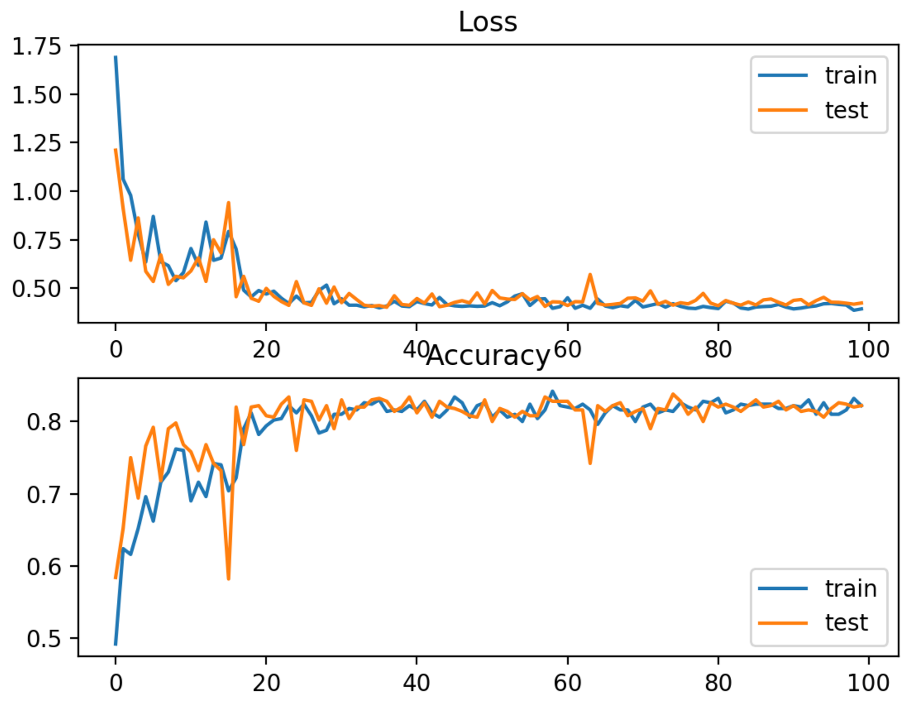

Figure 9: Full data set Accuracy and Loss variation over epochs .

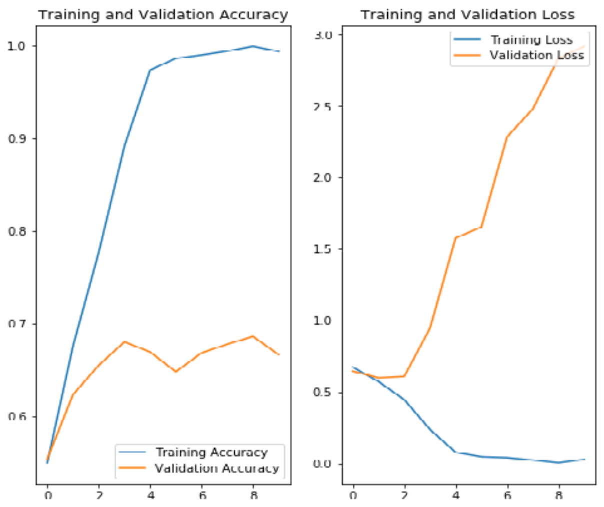

Figure 10: PCA data set Accuracy and Loss variation over epochs .

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

1. The genetic algorithm uses three main types of rules at each step to create the next generation from the current population: Selection rules select the individuals, called parents, that contribute to the population at the next generation.
2. Crossover rules combine two parents to form children for the next generation.

3. Mutation rules apply random changes to individual parents to form children.

Implementation of Genetic Algorithm

The algorithm we implemented consist of 4 different steps.

1. Initialization of hyper-parameters
2. selection hyper-parameters for each generation.
3. Crossover.
4. Mutation of generations

Initially 6 parameters were selected to optimize which are, earning_rate, n_estimators, max_depth, min_child_weight, colsample_bytree, and gamma. Thereafter, those parameters were randomly initialized and set the limits which it can varies.

The in the second step we used 10 fold accuracy to evaluate the model and check the fitness of the model, Then based on the fitness level the parent has been selected.

There are various methods to define crossover in the case of genetic algorithms, such as single-point, k-point crossover and uniform crossover etc. In here the uniform crossover has been considered which select select parameters for the child independently from the parent.

finally in mutation, change parent parameters in random amounts and which will make it unpredictable and make perfect for the algorithm, but the there will be a limit for the change of a parameter.

Apply Algorithm

When applying the algorithm we have considered all the prier prepossessing steps we have considered before applying other algorithms.

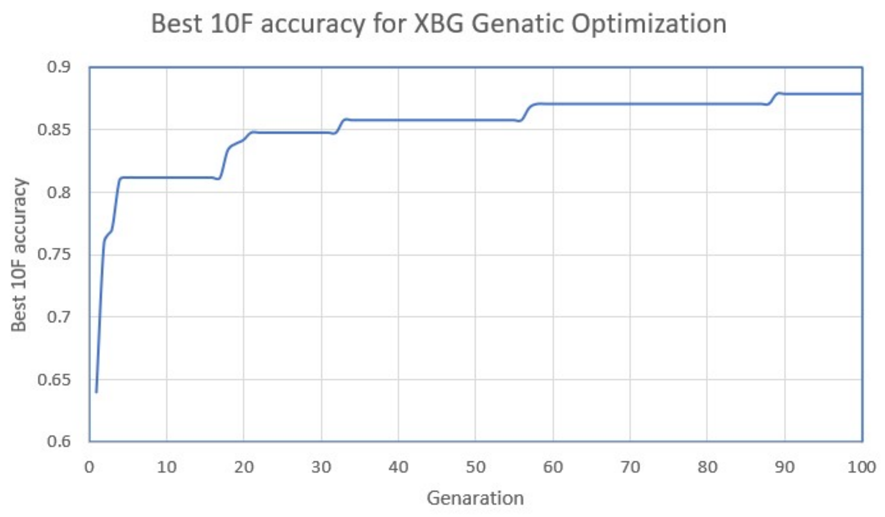

Figure 11: Maximum accuracy obtained until each generation.

when consider the above figure 11, it shows that the best model changes and improves over generations.

Table 5: XGBoost with full data-set

Name	10 Fold Values XGB
Accuracy	87.90%
Precision	80.02%
Recall	89.13%
F1 score	84.31%

Evaluation

Finally, after implementing all the algorithms and obtained the results there was an option to select the best algorithm which is suitable for this data set. For that we have considered the F1 score,10 fold accuracy, Recall and procession of each algorithm and selected the highest. Where XGB with granitic optimization and the neural network trained with full data set came hand in hand.

**Consider the Table-2 and Table-3

Conclusion

In conclusion, initially the raw data has been taken and passed through the prepossessing which consist of encoding the categorical data and replacing the missing values. Thereafter, data has been visualized in many different ways using dimensionality reduction methods, removed outliers and checked the correlation matrix and checked the features with the domain understanding. Next, 11 different models were trained using Classical Machine Learning models, A neural network has been trained with the full data set and the PCAed data set and finally two models (XGB classifier and CatB classifier) have trained using genetic Optimization for 100 generations. Finally based on the F1 score and accuracy the models were selected which are XGB with granitic optimization and the neural network trained with full data set.

References

[1] Cassel, M. and Lima, F., 2006, July. Evaluating one-hot encoding finite state machines for SEU reliability in SRAM-based FPGAs. In 12th IEEE International On-Line Testing Symposium (IOLTS'06) (pp. 6-pp). IEEE.

[2] Zou, H., Hastie, T. and Tibshirani, R., 2006. Sparse principal component analysis. Journal of computational and graphical statistics, 15(2), pp.265-286.

[3] Hesterman, J.Y., Caucci, L., Kupinski, M.A., Barrett, H.H. and Furenlid, L.R., 2010. Maximum-likelihood estimation with a contracting-grid search algorithm. IEEE transactions on nuclear science, 57(3), pp.1077-1084.

[4] Maaten, L.V.D. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), pp.2579-2605.

[5] Wickelmaier, F., 2003. An introduction to MDS. Sound Quality Research Unit, Aalborg University, Denmark, 46(5), pp.1-26.

[6] Hsu, H.H., Yang, A.C. and Lu, M.D., 2011. KNN-DTW based missing value imputation for microarray time series data. Journal of computers, 6(3), pp.418-425.

[7] Little, R.J., 1988. A test of missing completely at random for multivariate data with missing values. Journal of the American statistical Association, 83(404), pp.1198-1202.

[8] Núñez Castro, H., González Abril, L. and Angulo Bahón, C., 2011. A post-processing strategy for SVM learning from unbalanced data. In 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2011), p 195-200. Ciaco.

[9] Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[10] Dorogush, A.V., Ershov, V. and Gulin, A., 2018. CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363.

[11] Khashman, A., 2009. A neural network model for credit risk evaluation. International Journal of Neural Systems, 19(04), pp.285-294.

[12] Lefner, K. and Zabran, S., Trans Union LLC, 2010. Credit approval monitoring system and method. U.S. Patent 7,835,983.

[13] Whitley, D., 1994. A genetic algorithm tutorial. Statistics and computing, 4(2), pp.65-85.

[14] Harik, G.R., Lobo, F.G. and Goldberg, D.E., 1999. The compact genetic algorithm. IEEE transactions on evolutionary computation, 3(4), pp.287-297.

[15] Kohavi, R., 1995, August. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).

[16] Raschka, S., 2018. Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808.