Loan Approval Prediction

Fundamentals of Data Science, 2021-2022 project

Albiona Guri, Alexandru Popa, Elvina Lika, Lorenzo Del Signore, Luca Stravato

Abstract

The main objective of this project is to automate the loan qualifying procedure. Predict loan approval is a decision-making process for determining whether an applicant is eligible for a loan or not, based on applicant information.

Introduction

With the enhancement in the banking sector lots of people are applying for bank loans but the bank has its limited assets which it has to grant to limited people only, so finding out to whom the loan can be granted will be a safer option for the bank. In order to automate the loan qualifying procedure we implemented a machine learning algorithm in order to predict the loan approval.

Related works

We inspired our work from a kaggle challenge[1] and took as reference a response to the dataset we used, analyzing his choices.

Proposed method explained

Since the problem is a binary classification we chose to use different machine learning models, one for each team member.

- Logistic Regression
- Support Vector Classification (SVC)
- Gaussian Naive Bayes (GaussianNB)
- K-Nearest Neighbors (KNeighborsClassifier)
- Linear Discriminant Analysis (LinearDiscriminantAnalysis)

Dataset and benchmark

The dataset used is taken from the kaggle challenge mentioned above.. The interesting part of this project is that the dataset is highly imbalanced. Resulting in 68% of the majority class and only 32% of the minority class.

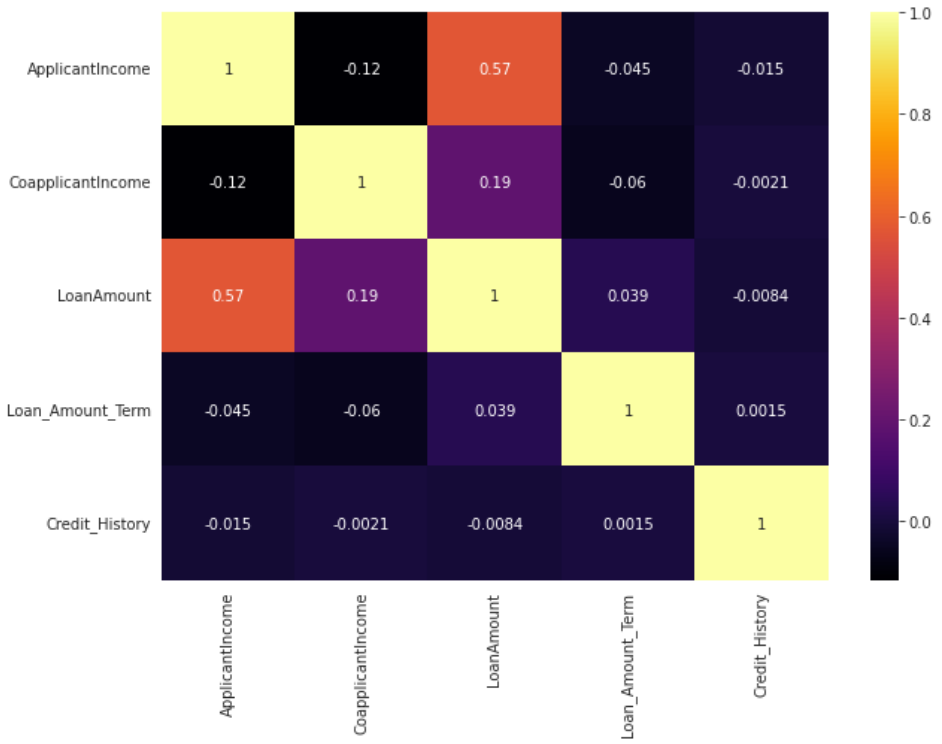


Figure 1. Correlation between features in the dataset.

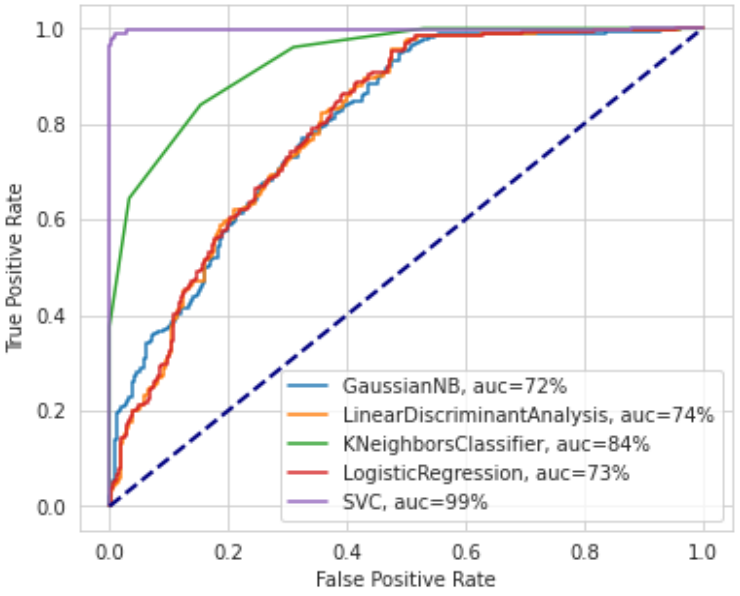
To prepare the data for training we have performed these feature engineering steps:

- We split the dataset into training and test set before any data preprocessing approach and applied all pre-processing techniques to train and test set independently in order to avoid data leakage(overfitting).
- Remove columns that are not relevant for prediction like Loan_ID. This feature is not important for prediction since it is unique in each and every row.
- For categorical and discrete features we used mode to fill the missing values. For continuous features we used median if there are outliers and mean if there aren't outliers as it is sensitive to them.
- Since there are few categories in categorical features we applied One-Hot encoding to convert them into numerical ones.
- We applied StandardScaler feature scaling technique since our features are in different scales and some machine learning models which we trained assume that the features are in the same range.
- The dataset is highly imbalanced(68% - 31%) thus we applied the SMOTE oversampling technique to approach a 50/50 balance of the minority and majority class. Usually, in imbalanced datasets the model is biased towards the minority class. We should give equal priority to each class. We must not balance the test set as it is supposed to be completely unseen for our model.
- We applied the PCA technique to describe the data using the most important principal components. In this way we removed noisy information from the dataset. The higher the number of the principal component, the higher the variance is. We should keep in mind that we do not usually want to reach 100% explained variance, because the dataset may contain noisy data, outliers, etc., which we want to avoid. To find the right amount of principal components, we tuned different values, checked how the model performed and chose the one that had the best performance.

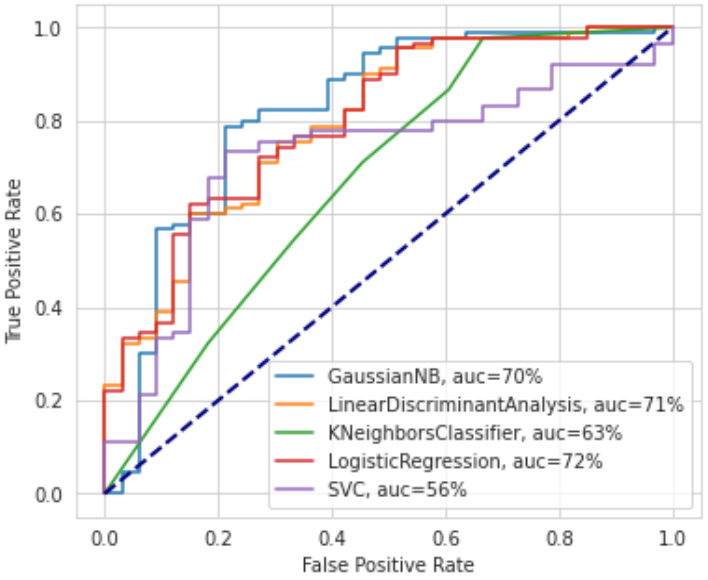
Experimental results

ROC curve

Training Data



Testing Data



Benchmark

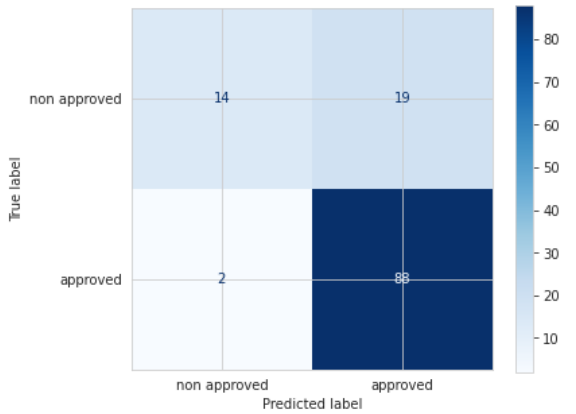
Training Data

-	precision		recall		f1-score		roc auc	cross val
(class)	**0**	**1**	**0**	**1**	**0**	**1**	**-**	**-**
GaussianNB	96%	64%	45%	98%	62%	78%	72%	69%
LinearDiscriminantAnalysis	93%	67%	52%	96%	67%	79%	84%	76%
KNeighbors Classifier	84%	85%	85%	84%	84%	84%	84%	64%
Logistic Regression	85%	67%	55%	91%	67%	77%	73%	81%
SVC	98%	99%	99%	98%	99%	99%	99%	69%

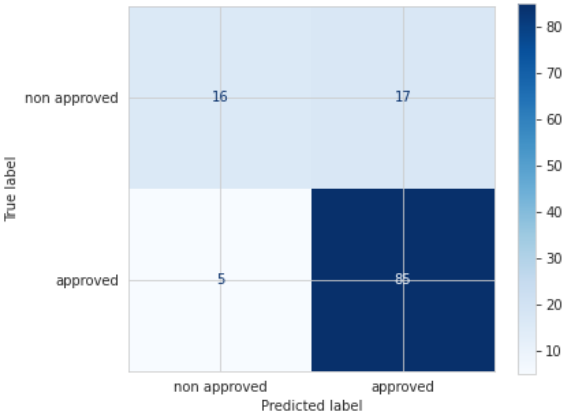
Testing Data

-	precision		recall		f1-score		roc auc	cross val
(class)	**0**	**1**	**0**	**1**	**0**	**1**	**-**	**-**
GaussianNB	88%	82%	42%	98%	57%	89%	70%	69%
LinearDiscriminantAnalysis	76%	83%	48%	94%	59%	89%	71%	76%
KNeighbors Classifier	41%	81%	55%	71%	47%	76%	63%	64%
Logistics Regression	64%	84%	55%	89%	59%	87%	72%	81%
SVC	39%	76%	27%	84%	32%	80%	56%	69%

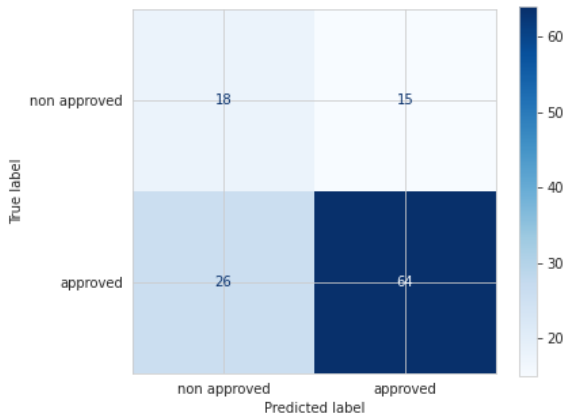
Confusion Matrix



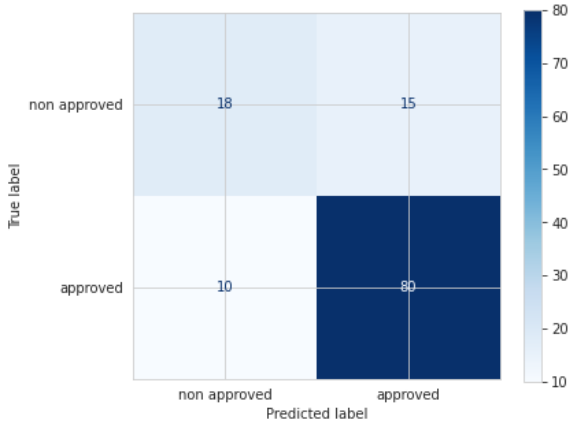
GaussianNB



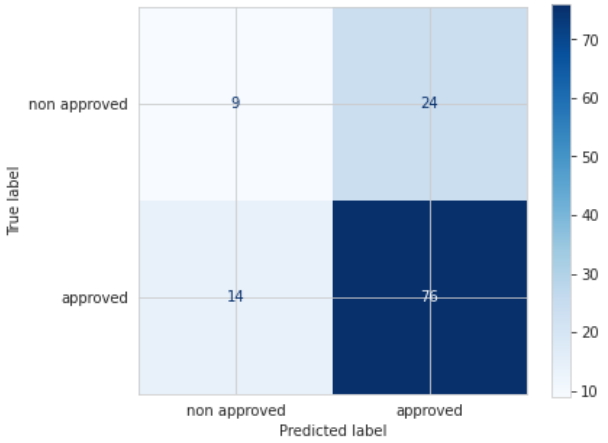
LinearDiscriminantAnalysis



KNeighborsClassifier



Logistic Regression



SVC

Hyper tunning

We did parameter hyper tuning for all our models and got two groups of models:

The first group

Includes the Support Vector Machine and K Nearest Neighbor Models.
The parameters needed to be tuned for each model were:

1. SVC → kernel, C, gamma
    a. We got best results for { 'C' : 100, 'gamma' : 1, 'kernel' : 'rbf' }
2. KNN → we did a combination for the number of principal components together with the number of neighbors.
    a. We got best results for 11 components of PCA and k_neighbours: 14

The second group

Includes GaussianNB, LinearDiscriminantAnalysis and LogisticRegression. The parameters needed to be tuned for each model were:

1. LinearDiscriminantAnalysis → solver, tol, shrinkage
    a. We got best results for { 'solver': 'svd', tol: 0.0 }
2. LogisticRegression → C, solver
    a. We got best results for { 'C': 1.0, 'solver': 'liblinear'}
3. GaussianNB → var_smoothing
    a. We got best results for { var_smoothing: 0.9202966808760416 }

Conclusions and future work

On Kaggle they have used Accuracy as the decision-making metric which is not the right metrics that should be used in case of imbalanced data. In comparison with the project from Kaggle, we chose AUC score because we should find a model which performs well in both minority and majority class and not just only in majority class.

For the first group of models
The SVC and KNN, after changing our pre-processing phases and after tuning a range of different values and selecting the combination with the highest AUC_score, we got lower results than the original project.

We discovered that one of the reasons that these models do not perform well is that SVC and KNN models do not perform well on unbalanced data. The performance on the testing set was worse than the other models. Our finding is that these models overfit in unbalanced data.

In conclusion, by taking into account the facts that we chose Precision as the main metric, that these models do not perform well on unbalanced data and that our dataset is unbalanced we can say that SVC and KNN are not the right models for our problem.

For the second group of models
Logistic Regression, LinearDiscriminantAnalysis and GaussianNB we got almost the same values for the performance metrics.

For our problem we decided the best model would be Logistic Regression since we got a high precision, recall, f1-score for each class. Also, the most important metric, in an unbalanced dataset, we got the highest AUC score.

Our problem is about a binary classification problem. The goal is to correctly classify people who should not get approved for a loan. That is why our main focus is the F1-score and AUC score.

Contributions

All team members contributed by analyzing the dataset(which was done using feature visualization) and by working on the preprocessing phase. After that we all worked together on having a final pre-processing phase to be used on all models.
As more individual work, each of us had one model to analyze further on and execute the hyper tuning of the parameters for their specific machine learning model, as seen below:

- Logistic Regression: Luca Stravato - 1755633
- Support Vector Classification: Elvina Lika - 2024059
- Gaussian Naive Bayes: Alexandru Popa - 1840967
- K-Nearest Neighbors: Albiona Guri - 1992979
- Linear Discriminant Analysis: Lorenzo Del Signore - 1605952

The final phase was discussing together for each personal finding and deciding on the best model for our problem.

References

1. https://www.kaggle.com/caesarmario/86-eligibility-prediction-w-various-ml-models
2. BAO, Fuguang, et al. Effect Improved for High-Dimensional and Unbalanced Data Anomaly Detection Model Based on KNN-SMOTE-LSTM. Complexity, 2020, 2020.
3. ILYAS, Sadaf, et al. Predicting the future transaction from large and imbalanced banking dataset. Int. J. Adv. Comput. Sci. Appl., 2020, 11.1: 273-286.