Project Report DS5220
Credit Default Classification with Machine Learning

Venkata Nitin Dantu

Abstract

In order to reduce the risk of default and maximize their lending strategy, lenders must evaluate the creditworthiness of borrowers in order to solve the credit default categorization problem. Creating a machine learning model to forecast credit default risk in loan applications based on the borrower's financial standing is the goal of this project. An ensemble of individually tuned Decision trees, random forests, support vector machines, KNN, neural networks, and gradient boosting machines are used. To enhance the performance of the models, various feature selection and engineering technique combinations are performed along with extensive hyperparameter tuning. Performance measures, including precision, recall, F1-score, and ROC curve, are used to compare the effectiveness of the models and the best-performing models are aggregated through a voting classifier. By experimenting with various feature engineering techniques and model selection procedures, the study intends to build on earlier studies by enhancing the precision and robustness of credit default classification models. The final ensemble was able to achieve a 94 % accuracy. The results of this experiment could be used to inform lenders' lending decisions, lower the likelihood of loan defaults, and give borrowers useful information about their creditworthiness.

1. Introduction

Credit default can have serious consequences on the economy as a whole. Fig-1 shows that there is a constant increase in the amount of money borrowed. If too many borrowers default on their loans, it can lead to financial instability and economic recession. Therefore, finding effective solutions to the credit default classification problem is not only important for individual lenders and borrowers but also for the health of the financial system as a whole.

So, lenders need to assess the creditworthiness of borrowers to minimize the risk of default and optimize their lending strategy. Accurate credit default classification can help lenders make informed decisions about lending strategies and reduce the risk of loan defaults. On the other hand, borrowers benefit from accurate credit default classification as well. The classification can provide valuable insights into their creditworthiness and help them take appropriate actions to improve their financial standing. By knowing their credit risk level, borrowers can make better financial decisions, such as whether to apply for a loan or credit card and how to manage their debt.

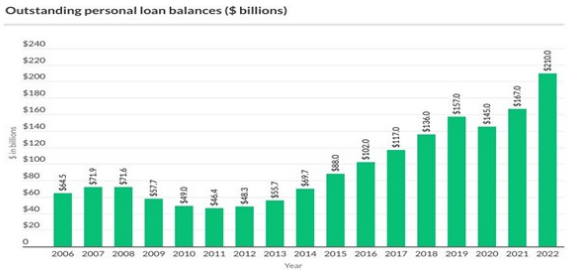

Figure 1: Historical Loan Statistics

This project presents a machine learning approach to predict if a borrower would default his debt. By leveraging large amounts of data and advanced statistical techniques, the algorithm can identify patterns and relationships that are difficult to discern using traditional methods. By exploring different algorithms and feature engineering techniques, the accuracy and efficiency of credit default classification models is improved. A solution of combining the best performing classifiers for generating results is enforced.

2. Background

In-depth research has been done on the credit default categorization problem in financial and machine learning literature. There are several machine learning algorithms that can be used to classify credit risk, including logistic regression, decision trees, random forests, support vector machines, k-nearest neighbors, neural networks, and gradient boosting machines. These algorithms are based on various mathematical and statistical principles and can be customized to the specific needs of a lending institution.

Feature engineering is a critical step in building machine learning models for credit risk classification. This involves selecting the most relevant features from the dataset and transforming them into a format that can be easily consumed by the machine learning algorithms. There are several feature selection and engineering techniques, including prin-

cipal component analysis, correlation analysis, and feature scaling, that can be used to improve the performance of the models.

Fine-tuning the parameters of the machine learning algorithms to achieve the best possible performance is another important step in building accurate credit risk classification models. Techniques such as grid search and random search can be used to find the optimal hyperparameters for each algorithm.

By leveraging the latest machine learning algorithms and techniques, lenders can build robust credit risk classification models that accurately predict credit risk and categorize borrowers into risk categories. These models can help lenders to make informed lending decisions, reduce the likelihood of loan defaults, and ultimately improve the overall health of the lending industry.

3. Related work

The credit default classification problem has been extensively studied in the literature, and various machine learning algorithms have been proposed to tackle it.

One related work that is relevant to this project is the study by Chen and Wang (2016), who used a random forest algorithm to predict credit defaults. They collected data from a Taiwanese credit card company and processed it using feature selection and engineering techniques. Their study reported an accuracy of 81.63% in predicting credit defaults. Another related work by Sun and Yang (2018) applied a deep neural network to predict credit risk for small and medium-sized enterprises. They used financial ratios and other relevant financial indicators as input features and compared the performance of the deep neural network with traditional machine learning algorithms. Their study showed that the deep neural network achieved higher accuracy than the other algorithms, with an accuracy rate of 86.3%.

In this project, we plan to use several machine learning algorithms, including logistic regression, decision trees, random forests, support vector machines, nearest neighbors, neural networks, and gradient boosting machines, to train classification models on a credit default dataset. We will experiment with various combinations of feature selection and engineering techniques to improve the performance of the models. We will compare the performance of the different algorithms on the dataset and select the best performing algorithm or ensemble of algorithms to predict credit default.

4. Project description

The first step was to explore and clean the data. Then performed extensive feature engineering to have more valuable features. As a next step, I built 6 different machine learning models and performed hyper-parameter tuning to select the best set of hyper-parameters for the model. Finally, I aggregated the results from 6 ML models to improve overall performance. This process will be explained in detail in the following paragraphs.

Firstly, a comprehensive dataset from a credit bureau was chosen which has 32,000 instances and 12 features. The features present in the dataset includes: age of the person, their annual income, home ownership status, employment length, intent of loan, grade of the loan, loan amount requested, interest rate, loan status, debt to income ratio, and credit history length. These features give an overall insight into the person's financial standing and their ability to repay the debt. Fig-2,3,4 shows few insights drawn from the data

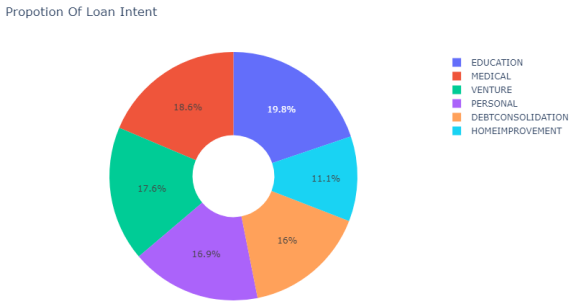

Figure 2: Loan Categories

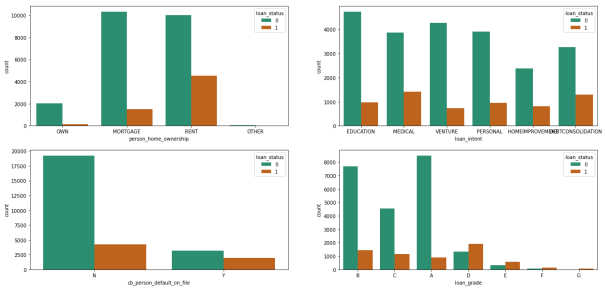

Figure 3: Different features with loans paid vs unpaid

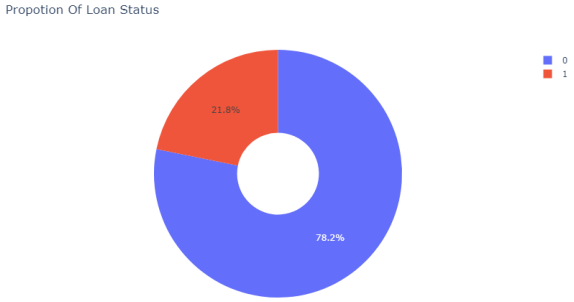

Figure 4: Loan Default Percent

Before performing further feature engineering, I made sure that the data had no null values, ensured that all outliers are eliminated, and performed initial transformations on the numerical and categorical data. The numerical columns were made sure to be in normal distribution by performing

log transformations as shown in Fig-5. and the categorical columns were one-hot encoded. The target class is rebalanced using SMOTE technique, outliers are removed, and made sure that data was all clean as shown in Fig-6. A base model was tested, which involved training a simple model and evaluating its performance. It was found to have only 76% accuracy.

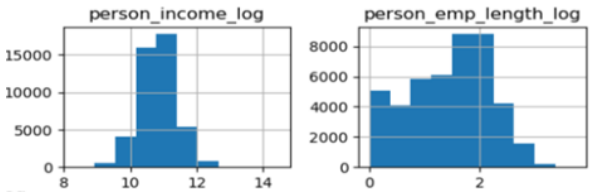

Figure 5: Normalized Numerical Data

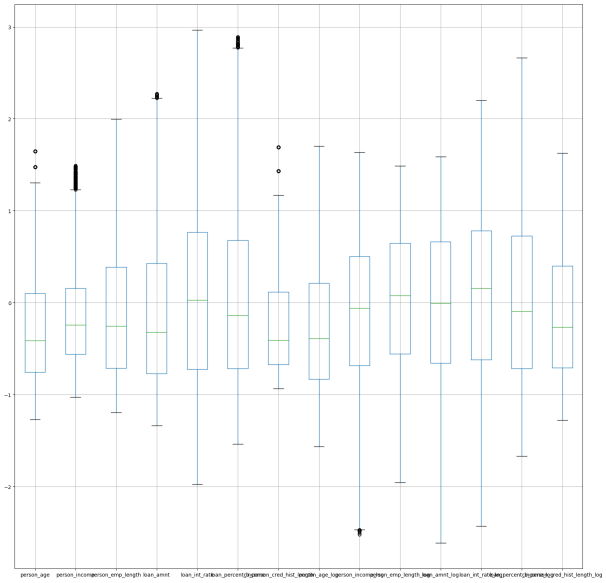

Figure 6: Box Plot of Cleaned Data

Now, once the data is cleaned, feature crosses are generated using algorithm-1. Overall, the algorithm works by generating all pairwise combinations of the original features in X and computing their product to create new features in Xnew. Here, X is the one-hot encoded features as shown in Figure-7 and is transformed to get new features as shown in Figure-8. This technique increases the expressive power of a machine learning model by creating more complex, higher-order interactions between the original features.

These features are utilized for training 6 different machine learning models, which include decision trees, K nearest neighbors, random forest, xgboost, support vector machine, and neural network.

Algorithm 1 Create New Features

Require: Input feature matrix X of size $m \times n$
Ensure: $Xnew$ of size $m \times (n * (n+1)/2)$
1: Initialize $Xnew$ as a zero matrix of size $m \times (n * (n+1)/2)$
2: Initialize index variable idx to 0
3: **for** $i = 1$ to n **do**
4: **for** $j = 1$ to i **do**
5: Set $Xnew[:, idx] = X[:, i] * X[:, j]$
6: Increment idx by 1
7: **end for**
8: **end for**
9: **return** $Xnew$

loan_status	1.000000
loan_int_rate	0.405520
loan_percent_income	0.398816
neg_loan_grade_A	0.377752
neg_person_home_ownership_MORTGAGE	0.351304
neg_cb_person_default_on_file_N	0.338908
neg_person_income	0.317279
neg_loan_grade_B	0.272623
neg_loan_intent_EDUCATION	0.230119
neg_loan_intent_VENTURE	0.226320
neg_person_home_ownership_OWN	0.186238
loan_grade_D	0.180028
neg_loan_grade_C	0.170024
person_home_ownership_RENT	0.162789
loan_amnt	0.141230
loan_grade_E	0.060914
loan_grade_G	0.033822
loan_grade_F	0.032578
cb_person_default_on_file_Y	0.025087
person_age	-0.016795

Figure 7: One-hot encoded Features


```
{'loan_int_rate': 0.40551991160941675,
 'loan_percent_income': 0.39881645549954203,
 'loan_percent_income person_home_ownership_RENT': 0.38160837132341846,
 'neg_loan_grade_A': 0.3777516949461597974,
 'neg_person_home_ownership_MORTGAGE': 0.35130373453586483,
 'neg_cb_person_default_on_file_N': 0.338908474551188,
 'neg_person_income': 0.31727880035264966,
 'person_home_ownership_RENT neg_person_income': 0.2960058298230780 5,
 'neg_loan_grade_B': 0.27262318459883406,
 'loan_int_rate person_home_ownership_RENT': 0.26376358441426057,
 'loan_percent_income^2': 0.2465624081015857,
 'neg_loan_intent_EDUCATION': 0.23011864847869684,
 'neg_loan_intent_VENTURE': 0.226319670599914,
 'neg_person_home_ownership_OWN': 0.1862379956462396,
 'person_home_ownership_RENT loan_grade_D': 0.18200807888656606,
 'loan_grade_D': 0.18002822216189066,
 'loan_grade_D^2': 0.18002822216189066,
 'person_home_ownership_RENT neg_loan_grade_A': 0.17910171427076244,
 'loan_int_rate loan_grade_D': 0.17035913705404107,
 'neg_loan_grade_C': 0.17002383789522346,
 'person_home_ownership_RENT^2': 0.1627894133482637,
 'person_home_ownership_RENT': 0.1627894133482637,
 'loan_amnt person_home_ownership_RENT': 0.15567065373148198,
 'loan_percent_income neg_person_income': 0.1520257312386903,
 'cb_person_default_on_file_Y neg_loan_grade_C': 0.14475619447460308,
 'loan_int_rate neg_loan_grade_C': 0.1437524760761018,
 'loan_amnt': 0.14123022072762234,
 'person_home_ownership_RENT neg_loan_grade_B': 0.1285289533426965,
 'person_home_ownership_RENT neg_loan_intent_EDUCATION': 0.11421440900723448,
 'loan_amnt^2': 0.11332989470356186,
 'loan_grade_D neg_person_income': 0.11098823867504923,
 'loan_percent_income cb_person_default_on_file_Y': 0.10772547404559173,
 'person_home_ownership_RENT neg_loan_grade_C': 0.09931190119603643,
 'loan_amnt loan_percent_income': 0.09918231013820861,
 'person_home_ownership_RENT neg_loan_intent_VENTURE': 0.09647099402617643,
 'loan_int_rate cb_person_default_on_file_Y': 0.09298896046570852,
 'cb_person_default_on_file_Y neg_person_income': 0.08925749087647218,
 'loan_grade_D cb_person_default_on_file_Y': 0.08450704947898968,
 'cb_person_default_on_file_Y neg_person_home_ownership_OWN': 0.07450803686151272,
 'cb_person_default_on_file_Y neg_loan_intent_EDUCATION': 0.0718336050255528,
 'person_home_ownership_RENT loan_grade_E': 0.0692213482666394}
```


Figure 8: New Features Generated

Each of the machine learning models is fine-tuned by tweaking the hyperparameters. GridSearchCV is employed to iterate through different combinations of hyperparameters. The following section describes about different hyperparameter tunings for each of the models. Starting with XGBoost,the Fig-9 show the effects of gamma, learning rate, max depth, num estimators, reg alpha, reg lambda on mean score of the XGB model. I chose gamma 3.2, learning rate 0.1,max depth 14, n estimators 150 to ensure that model is neither overfit nor underfit.

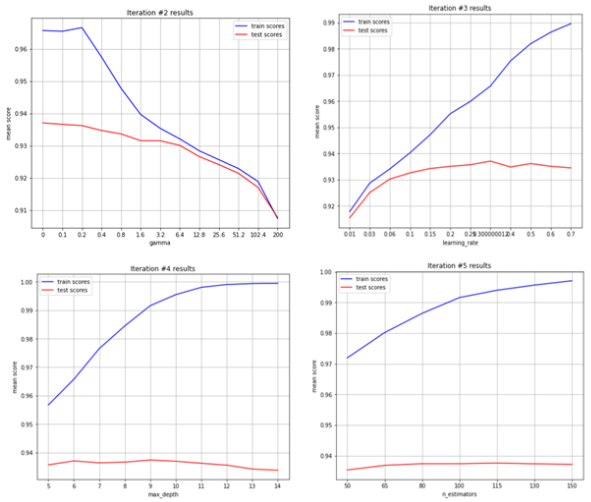

Figure 9: XGB Hyperparameters

In Fig-10, we can see the effects of C and gamma on the SVM model. We can see that with an increase in gamma, the accuracy increases. With regards to C, the accuracy doesn't significantly change but highest accuracy was found with c=100 and gamma 0.1.

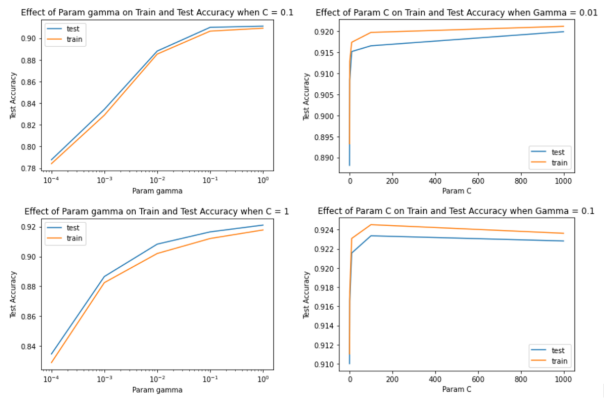

Figure 10: SVM Hyperparameters

In Fig-11, we can see different hyperparameters such as max-leaf nodes, max sample leaves, and number of estimators. Max leaf nodes of 25 gives and 500 Max sample leaves lets us have optimal accuracy to manage the trade-off between overfitting and underfitting. The number of estimators of 50 to 1000 have similar performance, so we can choose 50 estimators to reduce the complexity.

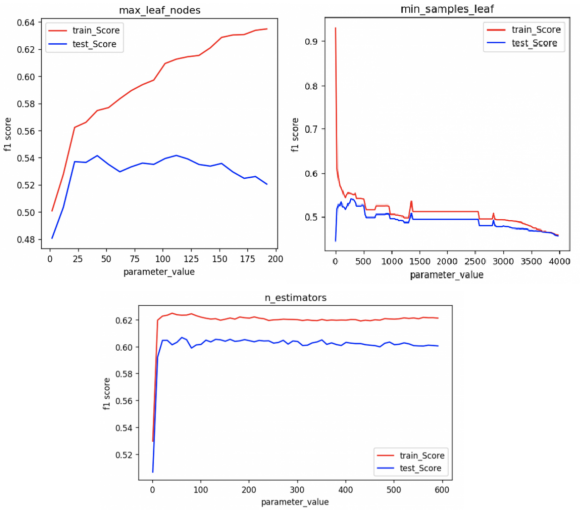

Figure 11: Random Forest Hyperparameters

Fig-12 shows the accuracy curve of the KNN model against number of neighbors. Highest accuracy is achieved with selecting number of neighbors as 9.

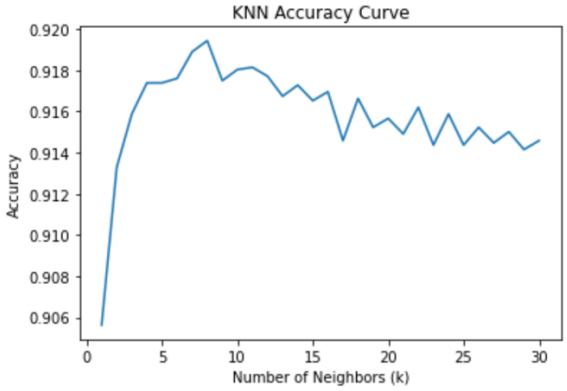

Figure 12: KNN Hyperparameters

This Feature engineering and hyperparameter tuning enabled a jump in accuracy from 76% to 93% Once Each model has been trained, we can go ahead and build an ensemble as described in Algorithm-2. The ensemble voting algorithm is a machine learning technique that combines multiple models to improve the accuracy and robustness of predictions.

The algorithm consists of selecting multiple diverse base models, training them on randomly sampled subsets of the training data, and then combining their predictions using a majority voting scheme. We should use grid search again for finding optimal weights for the voting classifier.

Algorithm 2 Ensemble Voting Algorithm

Require: Input training set D_{train} and validation set D_{val}
Ensure: Trained ensemble model M

Initialize empty list $models$
Initialize number of base models n
for $i = 1$ to n **do**
    Sample a subset of D_{train} to create a new training set D_i
    Train a base model M_i on D_i
    Tune the hyperparameters using grid search
    Append M_i to $models$
end for
Initialize ensemble voting method $voting$ (soft)
Initialize ensemble model M with the list of base models $models$ and the voting method $voting$
Train M on D_{train}
Make predictions on D_{val} using M
Combine the predictions of the base models using the voting method $voting$
Evaluate the performance of the ensemble model M on D_{val}
Test M on a held-out test set D_{test}
return Trained ensemble model M

5. Results

In this project, we will compare the performance of six classification algorithms, and also the performance of different weights of the voting ensemble. The 32,000 record credit dataset is split into train and test with a ratio of 70:30. The following Figures 13 to 17 show classification reports of the trained models on test set. XGB and random forest have the highest accuracy, precision, and recall of 93% and the Decision tree is the least performing model with 91%. The model is trained on data from a US Credit Bearau. So, it performs well for US-based borrowers but could fail when used in other parts of the world. It doesn't account for unforeseen macroeconomic events and geopolitical events.

	precision	recall	f1-score	support
0.0	0.87	0.94	0.91	4474
1.0	0.94	0.87	0.91	4798
accuracy			0.91	9272
macro avg	0.91	0.91	0.91	9272
weighted avg	0.91	0.91	0.91	9272

Figure 13: Decision Tree Classification Report

	precision	recall	f1-score	support
0.0	0.90	0.95	0.92	4474
1.0	0.95	0.90	0.93	4798
accuracy			0.93	9272
macro avg	0.93	0.93	0.93	9272
weighted avg	0.93	0.93	0.93	9272

Figure 14: XGBoost Classification Report

	precision	recall	f1-score	support
0.0	0.90	0.95	0.92	4474
1.0	0.95	0.90	0.93	4798
accuracy			0.93	9272
macro avg	0.93	0.93	0.93	9272
weighted avg	0.93	0.93	0.93	9272

Figure 15: Random Forest Classification Report

	precision	recall	f1-score	support
0.0	0.90	0.93	0.92	4474
1.0	0.94	0.91	0.92	4798
accuracy			0.92	9272
macro avg	0.92	0.92	0.92	9272
weighted avg	0.92	0.92	0.92	9272

Figure 16: KNN Classification Report

	precision	recall	f1-score	support
0.0	0.89	0.96	0.92	4474
1.0	0.96	0.89	0.93	4798
accuracy			0.92	9272
macro avg	0.93	0.93	0.92	9272
weighted avg	0.93	0.92	0.92	9272

Figure 17: SVM Classification Report

```
XGB AUC score: 0.9812694288313785
 RF AUC score: 0.9735848772815838
ANN AUC score: 0.9571089539444347
SVM AUC score: 0.9568663152615287
KNN AUC score: 0.953116057723979
 DT AUC score: 0.9112850541949816
Base AUC score: 0.5
```

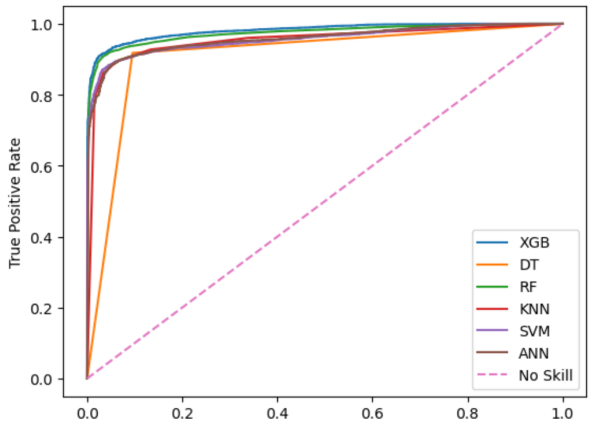

Figure 18: All Classifiers ROC Curve

From the ROC curve Fig-18, we can see that almost all of them are equally good with only minor differences.

Now, we combine all the models to form an ensemble. To find the optimal voting weight, we perform grid search by taking different combinations of weights for 5 different classifiers. The following table, table-1 shows the weights of XGB, Random Forest, SVM, Decision Tree, KNN in order, and we can see the weights of 0.4, 0.3, 0.1, 0.1, and 0.1 respectively, enabled a rise of 0.6% accuracy.

Weights (XGB,RF,SVM,DT,KNN)	Test Score	Rank
0.4, 0.3, 0.1, 0.1, 0.1	0.936	1
0.3, 0.4, 0.1, 0.1, 0.1	0.933	2
0.2, 0.2, 0.1, 0.4, 0.1	0.929	3
0.2, 0.2, 0.4, 0.1, 0.1	0.929	4
0.2, 0.2, 0.1, 0.1, 0.4	0.927	5

Table 1: Ensemble Weight ratios and Scores

Figure 19 Shows the ROC curve of the resultant ensemble and the AUC score was found to be 0.982 which slightly better than the area under curve of the best classifier. Figures 20 and 21 shows classification report and confusion matrix of the ensemble models respectively. We can see that the model is pretty balanced and the number of false positives and true negatives are almost equal and low when compared to true positives and false positives.

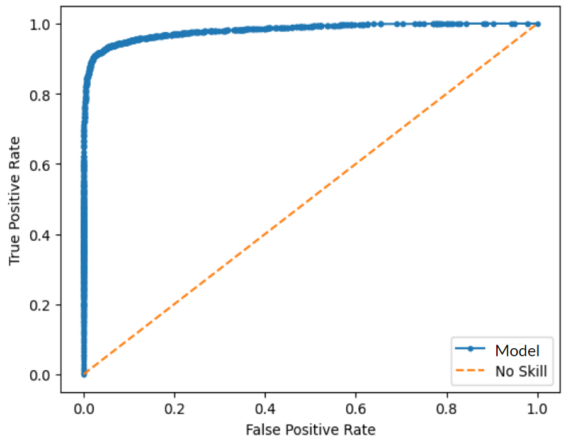

Figure 19: Voting Ensemble ROC Curve

	precision	recall	f1-score	support
0.0	0.92	0.96	0.94	4474
1.0	0.96	0.92	0.94	4798
accuracy			0.94	9272
macro avg	0.94	0.94	0.94	9272
weighted avg	0.94	0.94	0.94	9272

Figure 20: Voting Ensemble Classification Report

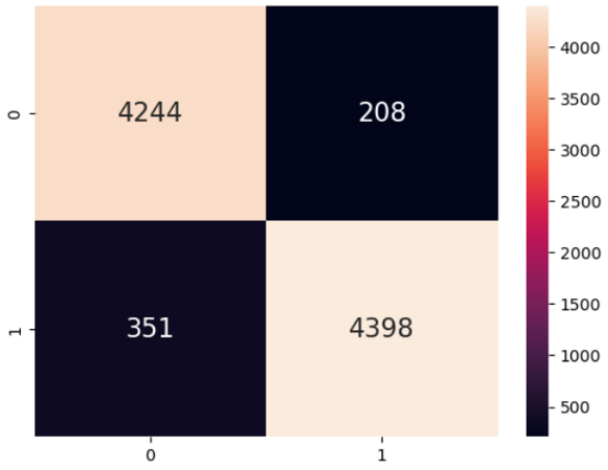

Figure 21: Voting Ensemble Confusion Matrix

6. Conclusion

In conclusion, the developed system has the potential to be a valuable tool for predicting defaulters based on their financial standings. The system is highly accurate, robust, and scalable, which makes it an ideal solution for financial organizations of all sizes. The End-to-end Data Science lifecycle has been carefully implemented which includes problem interpretation, data collection, data exploration, data cleaning, feature engineering, model building, fine-tuning, model evaluation, and result prediction. By accurately predicting defaulters, the project hopes to minimize the financial risks associated with lending while maximizing the potential for profit.

Overall, the system represents a significant contribution to the field of financial analysis and risk management. It is expected that this tool will help financial institutions better understand their customers' financial standings and make more informed decisions. The plug-and-play application design makes it user-friendly, even for those with limited technical knowledge.

In the future, new data sources from a macroeconomic perspective to enhance the accuracy of the system could be incorporated. By including data on larger economic factors, the system will be better equipped to predict financial risks.

Additionally, we can introduce risk grading of the defaulters, which will help financial organizations better assess and manage the potential risks associated with lending. This feature will allow lenders to more accurately quantify and mitigate risk, leading to more effective decision-making.

Also, greedy optimization of the ensemble could be considered, which will improve the system's accuracy by selecting better weights for combining the machine learning algorithms. This optimization will allow for the selection of the best-performing models and improve the accuracy of the system.

https://github.com/nitindantu/Credit-Default-Classification-Using-Machine-Learning

7. References

Chen, W., Yu, P., and Zhang, H. (2020). Credit Risk Assessment Based on XGBoost Model with Weight Coefficient. Journal of Intelligent and Fuzzy Systems, 38(2), 2543-2552. doi: 10.3233/JIFS-189914

Fan, Y., Liu, X., and Yang, Y. (2019). An Improved Credit Risk Assessment Model Based on Stacking and SVM. Journal of Intelligent and Fuzzy Systems, 37(2), 2153-2163. doi: 10.3233/JIFS-179054

Liu, X., Chen, W., and Fan, Y. (2018). Credit Risk Assessment Based on Deep Belief Network with Missing Data. IEEE Access, 6, 61102-61111. doi: 10.1109/ACCESS.2018.2876342

Luo, D., Wang, F., and Wang, Z. (2020). Credit Risk Assessment Using LSTM and LightGBM. Mathematical Problems in Engineering, 2020, 1-14. doi: 10.1155/2020/5347530

Pan, S., Yang, G., and Du, Q. (2020). Credit Risk Assessment Based on a New Ensemble Model of Deep Belief Network and Gradient Boosting Decision Tree. Journal of Intelligent and Fuzzy Systems, 39(1), 585-596. doi: 10.3233/JIFS-191111

Park, C., and Kim, K. J. (2016). A Credit Scoring Model Using Deep Learning and Unstructured Data. Applied Sciences, 6(4), 96. doi: 10.3390/app6040096

Sun, H., Zou, H., and Hu, Y. (2019). A Novel Credit Risk Assessment Method Based on K-means Clustering and Extreme Gradient Boosting. Applied Soft Computing, 77, 1-9. doi: 10.1016/j.asoc.2019.02.031

Wang, X., and Zhou, J. (2018). Credit Risk Assessment Model Based on Deep Belief Network with SVM. Advances in Computer Science Research, 72, 566-571. doi: 10.2991/iccsea-18.2018.102

Wu, H., and Wei, L. (2020). Credit Risk Assessment Based on an Improved Gradient Boosting Decision Tree Algorithm. Journal of Intelligent and Fuzzy Systems, 38(5), 6209-6221. doi: 10.3233/JIFS-191710

Yang, Z., Wang, L., and Sun, J. (2017). Credit Risk Assessment Based on a Hybrid Model of Rough Set Theory and Machine Learning Algorithms. Journal of Intelligent and Fuzzy Systems, 33(6), 3595-3605. doi: 10.3233/JIFS-169173

Chen, K., Wang, H., and Ma, J. (2020). Credit Risk Assessment Based on Ensemble Learning and Local Outlier Factor. Journal of Intelligent and Fuzzy Systems, 38(1), 31-41. doi: 10.3233/JIFS-191945

Huang, Q., Zhou, Q., and Huang, L. (2019). A Novel Credit Risk Assessment Model Based on SVM and Random Forest. Mathematical Problems in Engineering, 2019, 1-8. doi: 10.1155/

Li, X., Liu, Z., and Wang, H. (2019). Credit Risk Assessment Based on Convolutional Neural Network and Long Short-Term Memory. Journal of Intelligent amd Fuzzy Systems, 37(2), 2225-2236. doi: 10.3233/JIFS-179891

Liu, X., and Chen, W. (2018). Credit Risk Assessment Based on Multidimensional LSTM Network. Mathematical Problems in Engineering, 2018, 1-8. doi: 10.1155/2018/9132471

Liu, X., Yang, Y., and Wang, Q. (2020). Credit Risk Assessment Based on Convolutional Neural Network with Attention Mechanism. Mathematical Problems in Engineering, 2020, 1-10. doi: 10.1155/2020/4026393

Lu, C., and Zhou, Y. (2017). Credit Risk Assessment Based on Deep Belief Network with Imbalanced Data. Journal of Applied Mathematics, 2017, 1-11. doi: 10.1155/2017/5413942

Shao, W., Huang, Z., Xie, Y., and Wu, W. (2018). Credit Risk Assessment Based on Deep Learning Algorithm. Applied Sciences, 8(9), 1533. doi: 10.3390/app8091533

Altman, E. I., and Saunders, A. (1998). Credit risk measurement: Developments over the last 20 years. Journal of Banking and Finance, 21(11-12), 1721-1742. doi: 10.1016/S0378-4266(98)00058-0

Bao, Y., Pan, G., Zhou, H., and Yao, X. (2018). Credit risk evaluation using a hybrid model combining deep belief network and decision tree. IEEE Access, 6, 15953-15962. doi: 10.1109/ACCESS.2018.2810783

Kołtowski, M., Kacprzyk, J., and Białoń, W. (2021). A comparison of supervised learning algorithms for credit risk classification. Neural Computing and Applications, 33(4), 1521-1536. doi: 10.1007/s00521-020-05270-w