CREDIT CARD APPROVAL PREDICTION

TEAM MEMBERS:
1.T.Chandana Sree - 20MID0158
2. Damini N-20MID0119
3. Sai Sanjana Nednuri - 20MID0046
4. Epuru Sai Tejaswini -20MID0136

1 INTRODUCTION
1.1 Overview

Credit cards, which provide ease of shopping and payment without carrying cash, have become an important part of our daily lives. However, getting approved for a credit card can be a daunting task for many. This is where credit card authorization comes into play. The Credit Card Approval Estimation Project is a solution designed to help financial institutions determine credit card approval probability. The project aims to accurately predict the approval of a credit card application using machine learning algorithms and credit history data.
The system takes into account many factors such as income, employment, credit history and other variables to assess the creditworthiness of the applicant. This project builds machine-learning models with real-time accuracy using Python.

1.2 Purpose
The aim of the project is to provide financial institutions with powerful tools to make credit card approval decisions. Using this program, organisations can achieve the following:
1. Effectiveness
2. Risk Assessment
3. Advanced Decision Making

This project creates machine learning models with real-time accuracy using python.The Logistics model is a credit scoring method. The coefficients of each factor can be calculated as it is suitable for the logistic binary distribution function.

However, the goal is to build a better model, improve predictions, and save time by using and testing various algorithms such as Boosting, Random Forests, Multilayer Perceptrons, Decision Trees, and Support Vector Machines.

2 LITERATURE SURVEY
2.1 Existing problem
Existing approaches or methods to solve this problem

S.NO	Author	Journal Name	VOL.NO, JOURNAL.NO,	DESCRIPTION OF THE PAPER	DRAWBACK OF THEIR WORK

			PP.N O		
1	Lei Duan	Performance Evaluation and Practical Use of Supervised Data Mining Algorithms for Credit Card Approval	978-1-7281-7106-7/20/$31.00 ©2020 IEEE	In this article, they primarily utilized four models: logistic regression, decision tree, KNN, and neural network. With logistic regression, we not only assessed its performance but also examined the odds of belonging to a specific class for each variable, aiming to develop methods for applicants. For decision trees, they employed Gradient Boosting Machine, a powerful algorithm that optimizes the cost function and reveals the relative influence of variables.	Limited Model Comparison: The article focuses on the evaluation of supervised data mining algorithms for credit card approval but fails to provide a comprehensive comparison with a broader range of models, potentially limiting the understanding of the best-performing algorithms.
2	Maya Markova	Credit card approval model: An application of deep neural networks	AIP Conference Proceedings 2321, 030023 (2021); https://doi.org/10.1063/5.0040744	In this paper, the authors analyze a data sample of 690 credit card applications to build a predictor model using the Deep Learning Toolbox 14.0 in Matlab. They conduct a detailed analysis and correction of the input data and implement a deep neural network with different numbers of hidden layers. The binary classifier produced by the model can be used to divide data into two classes based on certain criteria. The paper highlights the importance of technological advancements in credit and deposit lines, credit card modelling, portfolio management, and financial risk assessment. The increasing number of credit card users and applications necessitates the development of automated models for	Firstly, they lack transparency, making it difficult to understand the decision-making process. Secondly, obtaining sufficient and diverse credit card approval data for training can be challenging.

				efficient processing. The authors focus on deep feed-forward networks or multilayer perceptrons as the preferred deep learning models for their study.	
3	Pathipati Yasasvia and S. Magesh Kumarb	Improve Accuracy in Prediction of Credit Card Approval Using Novel XGboost Compared with Random Forest	doi:10.3233/APC220083	This work compares the performance of the XGBoost algorithm and Random Forest (RF) for credit card approval prediction. The dataset consists of 19 attributes and a sample size of 48678. The XGBoost Classifier achieves an accuracy of 87.97% and a loss of 12.03%, outperforming RF, which achieves 82.86% accuracy and a 17.14% loss. Statistical analysis confirms the significant superiority of the XGBoost Classifier (p = 0.001) in predicting credit card approval.	Some potential drawbacks of using XGBoost or Random Forest for improving accuracy in credit card approval prediction is model complexity, longer training times, difficulty in interpreting the models' decision-making process, and potential overfitting or bias if not properly managed.
4	Mohammad Haseeb Dar, Neerendra Kumar	Using Machine Learning Methods to Forecast Credit Card Approvals	Volume 7, Issue 7, July – 2022 ISSN No:-2456-2165	In this paper, they used machine learning techniques to create a prediction system for automated credit card approvals, much like actual banks do.	Not much information is provided about the explanation and just showed the outputs.
5	R.H.DAVIS, D. B. EDELMAN, AND A. J. GAMMERMAN	Machine-learning algorithms for credit-card applications	IMA Journal of Mathematics Applied in Business & Industry (1992) 4, 43-51	The paper describes the motivation for using the machine-learning algorithms for credit-card assessment, describes the algorithms in detail, and compares the performance of these algorithms in terms of their accuracy.	Two computational models have been described and applied to a set of banking data. The present results are limited by the relatively small number of training examples and test set. The accuracy achieved by G&T was on the level of (~71%). As can be seen the 'proper Bayes' model has achieved a slightly higher level of accuracy than the

					'simple Bayes' model (~69%). The connectionist model was much slower in the learning part than G&T model, and achieved about 64% of accuracy.
6	Mohammed J. Islam, Q. M. Jonathan Wu, Majid Ahmadi, Maher A. Sid-Ahmed	Investigating the Performance of Naive- Bayes Classifiers and K-Nearest Neighbor Classifiers	0-7695-3038-9/07 $25.00 © 2007 IEEE DOI 10.1109/ICCIT.2007.148	The paper discusses the importance of probability theory in decision-making under uncertainty, focusing on classification tasks. It highlights the use of Bayes' rule to calculate class probabilities and rational classification for risk reduction. The paper introduces Bayesian theory, emphasizing the analysis of past data to predict the future. It presents two Bayesian learning methods: Naive Bayes and K-Nearest Neighbor (KNN) classifiers. The paper also explores how adjusting the value of K improves the performance of the KNN classifier.	Limited exploration of Bayesian theory,Lack of comparative analysis, Reliance on a single dataset, Lack of discussion on assumptions and limitations, Insufficient analysis of error rates.
7	Admel Huesejinovic	Application of Business Intelligence in decision making for credit card approval	. Vol 12 No.2 (2022) pp. 54-65	Researchers have taken credit card approval data from the UCI repository . They have applied chi-square test to find out the dependence of credit card approval on the features such as age, debt, prior default etc . Used the techniques of business intelligence for feature selection.Trained the model using logistic regression.86% accuracy was achieved . Results obtained have shown that prior default has highest influence on the credit card approval	Model was trained only with Logistic regression . Results were not compared with other models which may have given better accuracy. Only accuracy was used as a performance metric. Metrics like F1-score should be used for better evaluation

8	K. M. Azharul Hasan	CreditCardApproval Prediction byNon-negative Tensor Factorization	Conference Paper · February 2021 DOI: 10.1109/ICREST51555.2021.9331172	Researchers have used PARAFAC tensor factorization for predicting credit card approval . They have used the dataset from kaggle . 6 personal and financial features were taken and reduced to 3 features . Tensor factorization imposed less errors . Higher dimension prediction algorithm was used instead of handling 2-d matrix data .	PCA was used to do feature selection which is not an ideal approach when dealing with categorical features . obtained accuracy using tensors was not mentioned in the document .
9	Anupam Shukla	Design of Credit Approval System using Artificial Neural Network: A Case Study	Vol 4, Issue 6, June 2017 ISSN (Online) 2394-2320	Researchers have used artificial neural network as classification algorithm for credit card approval. Standard Australian credit card approval dataset is used. The dataset consists of 14 features and 1 Label column. An 88.67% accuracy was obtained.	Back propagation is the base of neural network. An error in the output layer propagates it back to the previous layers. Hence the training process is computationally expensive and time consuming.
10	Ipin Sugiyarto,Umi Faddillah,Bibit Sudarsono	Performance Comparison of Data Mining Algorithm to Predict Approval of Credit Card	Journal Publications & Informatics Engineering Research Volume 4, Number 1, October 2019	In order to forecast if a credit card application would be approved, this study examined a model utilising a neural network using the Principle Component Analysis (PCA) selection feature and optimised it using the Particle Swarm Optimise (PSO) algorithm. The finest outcomes came from a number of experiments. The findings of this study demonstrate that an accuracy of 80.33% may be achieved by using just one Neural Network technique. However, it has been demonstrated that the PCA + Neural Network + PSO hybrid technique may boost accuracy to 82.67%. Likewise, when the neural network was optimised	The limitations include the quality and representativeness of the dataset, the selection and relevance of features used, the possibility of overfitting and generalization issues, and the computational complexity of certain algorithms.

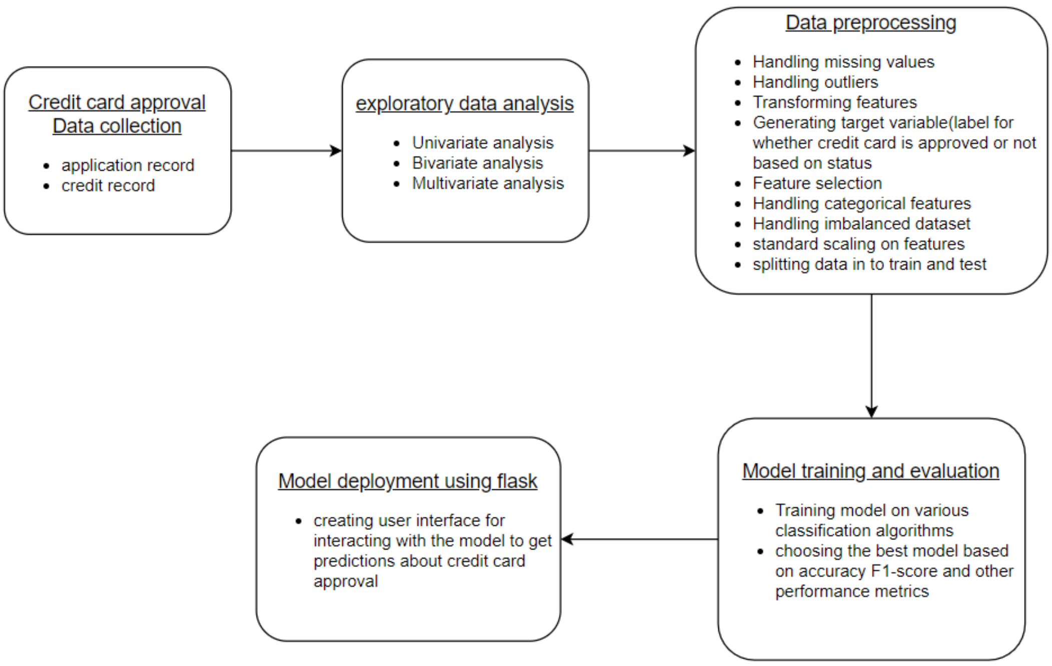
				using PSO and feature selection, the AUC NN value, which was 0.706, climbed to 0.74.	
11	1 Erkan Ilgun 2 Ensar Mekiš 3 Emina Mekiš	Application of Ann in Australian Credit Card Approval	European Researcher, 2014, Vol.(69), № 2-2	The study evaluates the potential of Artificial Neural Networks (ANNs) in the Australian financial sector for credit card approval. It uses learning algorithms(back propagation) and a historical dataset. Results show ANNs have higher accuracy and predictive skills than conventional techniques. Future research should focus on improving ANN models and addressing implementation challenges in the Australian financial sector.	The study's neural network model output layer had one neuron, limiting its ability to categorise results into two groups for credit approval. This limitation means deeper analysis is needed for applicants falling between "good" and "bad" credit categories..
12	Md. Golam Kibria and Mehmet Sevkli	Application of Deep Learning for Credit Card Approval: A Comparison with Two Machine Learning Techniques	International Journal of Machine Learning and Computing, Vol. 11, No. 4, July 2021	This study aims to develop a deep learning model for credit card approval using data from the University of California, Irvine. The model is evaluated against support vector machine and logistic regression models, and results show it performs more accurately and effectively than traditional machine learning methods. The deep learning model has the highest F1-measure score of.886, outperforming SVM and LR in terms of false positive rate. Additionally, SVM outperforms LR with a rate of 12.80%, while LR has a rate of 16.10%.	The research paper should consider the size of the dataset used for experimentation to ensure results are statistically significant and generalise well to larger credit card approval systems. A diverse and representative dataset can improve the model's performance.

2.2 Proposed solution

After analysing the previous works , the proposed solution handles imbalance in the dataset using SMOTE technique and trains the model using 8 classification algorithms i.e., Decision tree, Random forest,CatBoost , Xgboost, MLP classifier , SVM,Logistic regression and Adaboost classifier . Evaluates all the classifiers using evaluation metrics like accuracy,precision,recall and F1-score . Based on the performance, the best model is chosen for deployment .

3 THEORETICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software Designing

Hardware Requirements:

1. Computer or server: You will need a computer or server to host and run the Flask application.
2. Processor: A modern processor with multiple cores will ensure smooth execution of the application.
3. RAM: Sufficient RAM to handle the application's memory requirements. A minimum of 4GB is recommended, but it may vary depending on the size and complexity of the dataset and models used.
4. Storage: Adequate storage space to store the application code, datasets, and any other required files.

Software Requirements:

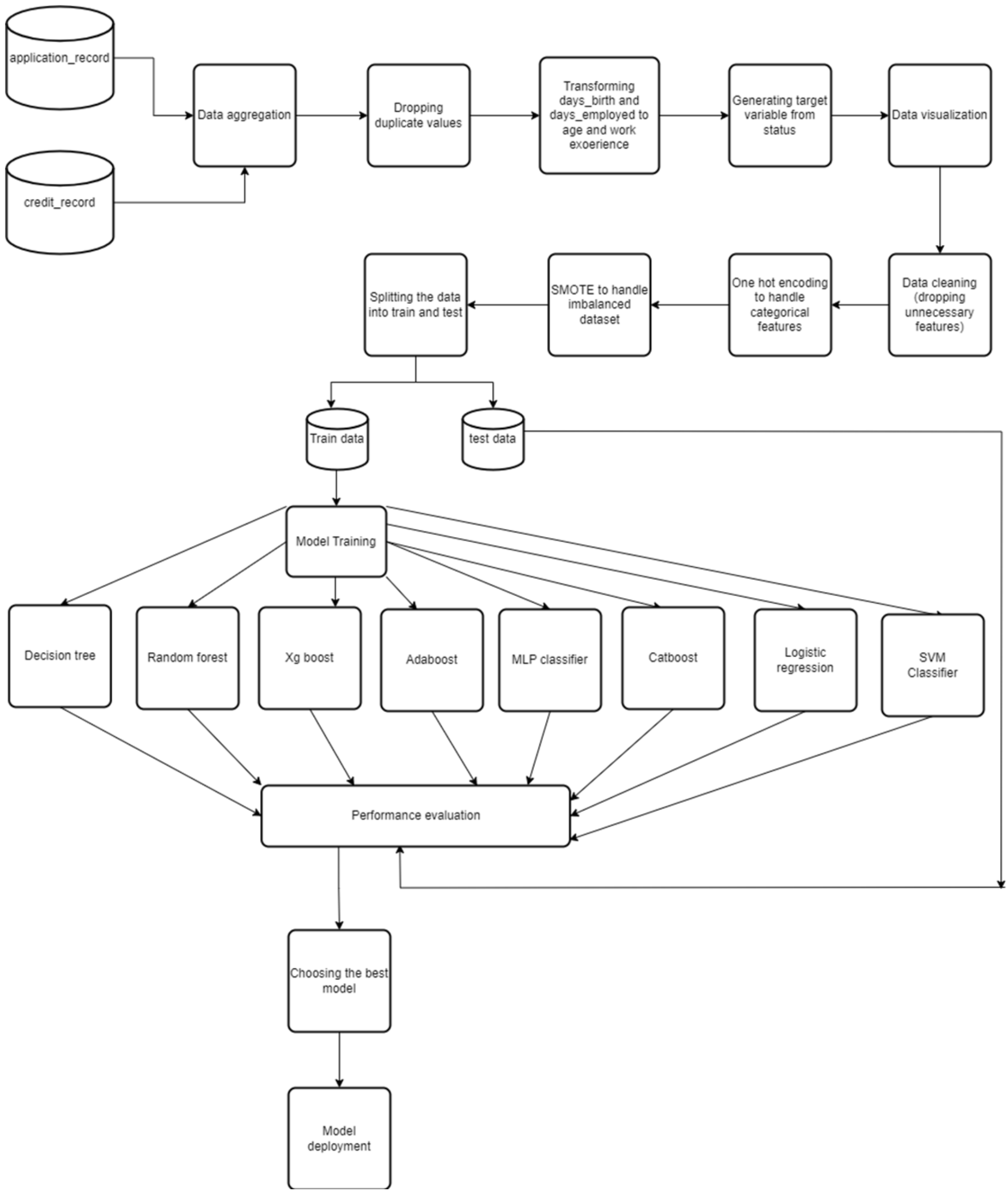
1. Python: Install the latest version of Python, as Flask and Jupyter Notebook are Python-based frameworks.
2. Flask: Flask is a lightweight web framework used for developing web applications in Python.
3. Jupyter Notebook: Jupyter Notebook is an interactive computing environment that allows you to create and share documents that contain code, visualizations, and explanations.
4. Libraries: Install the necessary Python libraries for data analysis, machine learning, and credit card approval system implementation. Some common libraries include pandas, numpy, sci-kit-learn, etc.

4 EXPERIMENTAL INVESTIGATIONS

During the analysis of the credit card approval system, several investigations were conducted while working with the credit card approval dataset. Firstly, the dataset was explored to understand the distribution and relationships between variables It was observed that the dataset given has a highly imbalanced target variable where 99.76% of the data has label 1 whereas only 0.26 % of the data has label 0. Occupation type feature has more than 30% missing values which may hinder the training process. Therefore, the occupation type feature was removed. From the correlation matrix, it was found that there was no correlation among independent variables 'FLAG_WORK_PHONE', 'FLAG_PHONE ', 'FLAG_EMAIL', and 'FLAG_MOBIL' do not contribute to credit approval. Therefore, these features are removed during feature selection. Since most of the features are categorical, random forest and decision trees can be used to get better results.

5 FLOWCHART

Diagram showing the control flow of the solution



6 RESULT

Algorithm	Accuracy	precision	Recall	F1-score
Decision tree classifier	99.67%	99.82%	99.51%	99.6%
Random forest classifier	99.87%	99.85%	99.89%	99.87%

Xgboost classifier	99.80%	99.87%	99.74%	99.8%
Catboost classifier	99.49%	99.78%	99.19%	99.49%
Logistic regression	85.67%	84.4	87.57%	85.9%
MLP classifier	96.96%	96.95%	96.60%	97.31%
Adaboost classifier	93.63%	93.63%	93.65%	93.64%
SVM classifier	95.88%	97.9%	93.75%	95.8%

Decision tree

```
n [47]: from sklearn import tree
        from sklearn.tree import DecisionTreeClassifier

n [48]: dtree = DecisionTreeClassifier(criterion='entropy',random_state=0)
        dtree = dtree.fit(x_train, y_train)

n [49]: d_pred = dtree.predict(x_test)

n [50]: from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score,confusion_matrix

        print(accuracy_score(y_test,d_pred))
        print(f1_score(y_test,d_pred))
        print(recall_score(y_test,d_pred))
        print(precision_score(y_test,d_pred))

        0.9967005773989552
        0.9966990647350082
        0.9951478531538954
        0.9982551198457159

n [51]: confusion_matrix(y_test,d_pred)

ut[51]: array([[10880,    19],
               [   53, 10870]], dtype=int64)
```

Random Forest Classifier

```
In [52]: from sklearn.ensemble import RandomForestClassifier
         rf = RandomForestClassifier(n_estimators=29,criterion='entropy',random_state=0)
         rf.fit(x_train,y_train)

Out[52]: RandomForestClassifier(criterion='entropy', n_estimators=29, random_state=0)

In [53]: y_pred = rf.predict(x_test)

In [54]:
         print(accuracy_score(y_test,y_pred))
         print(f1_score(y_test,y_pred))
         print(recall_score(y_test,y_pred))
         print(precision_score(y_test,y_pred))

         0.9987168912107048
         0.9987186527548966
         0.998992950654582
         0.9984445054442309

In [55]: confusion_matrix(y_test,y_pred)

Out[55]: array([[10882,    17],
                [   11, 10912]], dtype=int64)
```

Xgboost classifier

In [56]:
```python
# xgboost

import xgboost as xgb
xg= xgb.XGBClassifier(n_estimators=150,random_state=0)
xg.fit(x_train,y_train)
```

Out[56]:
```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=150, n_jobs=None, num_parallel_tree=None,
              predictor=None, random_state=0, ...)
```

In [57]:
```python
xg_pred = xg.predict(x_test)
```

In [58]:
```python
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score

print(accuracy_score(y_test,xg_pred))
print(f1_score(y_test,xg_pred))
print(recall_score(y_test,xg_pred))
print(precision_score(y_test,xg_pred))
```

```
0.9982586380716708
0.9982594356907294
0.997619701547194
0.998899990833257
```

In [59]:
```python
confusion_matrix(y_test,xg_pred)
```

Out[59]:
```
array([[10887,    12],
       [   26, 10897]], dtype=int64)
```

Catboost classifier

In [60]:
```python
# catboost

from catboost import CatBoostClassifier

clf = CatBoostClassifier(
    iterations=150,
    learning_rate=0.1,
)

clf.fit(x_train,y_train)
```

```
0:	learn: 0.6170986	total: 175ms	remaining: 26.1s
1:	learn: 0.5696106	total: 193ms	remaining: 14.3s
2:	learn: 0.5173375	total: 210ms	remaining: 10.3s
3:	learn: 0.4839869	total: 227ms	remaining: 8.3s
4:	learn: 0.4563183	total: 245ms	remaining: 7.09s
5:	learn: 0.4277710	total: 261ms	remaining: 6.25s
6:	learn: 0.4110717	total: 274ms	remaining: 5.6s
7:	learn: 0.3901431	total: 289ms	remaining: 5.14s
8:	learn: 0.3760916	total: 303ms	remaining: 4.75s
9:	learn: 0.3613465	total: 315ms	remaining: 4.42s
10:	learn: 0.3464118	total: 326ms	remaining: 4.12s
11:	learn: 0.3328103	total: 335ms	remaining: 3.86s
12:	learn: 0.3200114	total: 343ms	remaining: 3.62s
13:	learn: 0.3078279	total: 351ms	remaining: 3.41s
14:	learn: 0.2837048	total: 360ms	remaining: 3.24s
15:	learn: 0.2726751	total: 369ms	remaining: 3.09s
16:	learn: 0.2648212	total: 377ms	remaining: 2.94s
17:	learn: 0.2574197	total: 385ms	remaining: 2.82s
18:	learn: 0.2497048	total: 394ms	remaining: 2.71s
```

In [61]:
```python
cat_pred = clf.predict(x_test)
```

In [62]:
```python
print(accuracy_score(y_test,cat_pred))
print(f1_score(y_test,cat_pred))
print(recall_score(y_test,cat_pred))
print(precision_score(y_test,cat_pred))
```

```
0.9947300889011089
0.9947206537207914
0.991852055296164
0.9976058931860037
```

In [63]:
```python
confusion_matrix(y_test,cat_pred)
```

Out[63]:
```
array([[10873,    26],
       [   89, 10834]], dtype=int64)
```

Logistic Regression

In [64]:
```
# logistic regression

from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
lr.fit(x_train_scaled,y_train)
```

Out[64]:
```
LogisticRegression()
```

In [65]:
```
lr_pred = lr.predict(x_test_scaled)
```

In [66]:
```
print(accuracy_score(y_test,lr_pred))
print(f1_score(y_test,lr_pred))
print(recall_score(y_test,lr_pred))
print(precision_score(y_test,lr_pred))
```

```
0.85620016497113
0.8590929501571621
0.875766730751625
0.8430422138010046
```

In [67]:
```
confusion_matrix(y_test,lr_pred)
```

Out[67]:
```
array([[9118, 1781],
       [1357, 9566]], dtype=int64)
```

MLP Classifier

In [68]:
```
from sklearn.neural_network import MLPClassifier
clf2 = MLPClassifier(random_state=1, max_iter=10).fit(x_train_scaled, y_train)
```

In [69]:
```
mlp_pred = clf2.predict(x_test_scaled)
```

In [70]:
```
print(accuracy_score(y_test,mlp_pred))
print(f1_score(y_test,mlp_pred))
print(recall_score(y_test,mlp_pred))
print(precision_score(y_test,mlp_pred))
```

```
0.969663642195949
0.9695855922080309
0.9660349720772682
0.9731624089274186
```

In [71]:
```
confusion_matrix(y_test,mlp_pred)
```

Out[71]:
```
array([[10608,   291],
       [  371, 10552]], dtype=int64)
```

adaboost classifier

In [72]:
```
from sklearn.ensemble import AdaBoostClassifier
```

In [73]:
```
ada_clf = AdaBoostClassifier(n_estimators=175, random_state=0)
ada_clf.fit(x_train, y_train)
```

Out[73]:
```
AdaBoostClassifier(n_estimators=175, random_state=0)
```

In [74]:
```
ada_pred = ada_clf.predict(x_test)
```

In [75]:
```
print(accuracy_score(y_test,ada_pred))
print(f1_score(y_test,ada_pred))
print(recall_score(y_test,ada_pred))
print(precision_score(y_test,ada_pred))
```

```
0.9363944643020805
0.9364701574514829
0.9365558912386707
0.9363844393592677
```

In [76]:
```
confusion_matrix(y_test,ada_pred)
```

Out[76]:
```
array([[10204,   695],
       [  693, 10230]], dtype=int64)
```

SVM

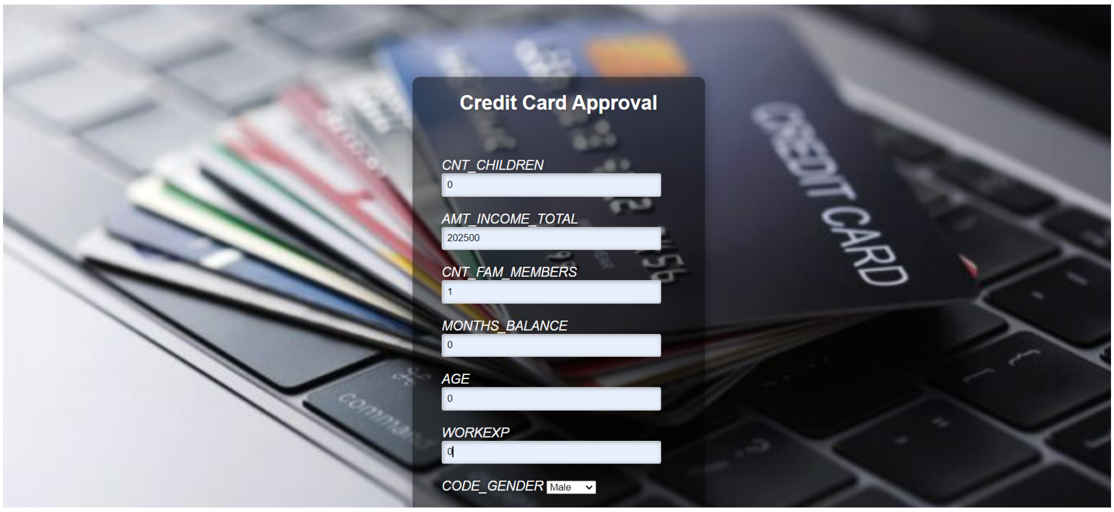
In [77]:
```python
from sklearn.svm import SVC
linear_svc=SVC(kernel="rbf")
```

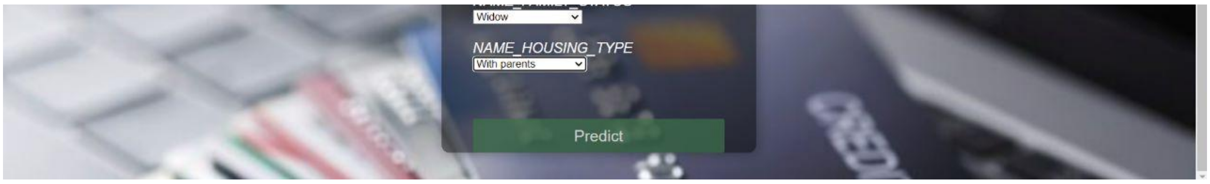
In [78]:
```python
msvc=linear_svc.fit(x_train_scaled,y_train)
svc_pred = msvc.predict(x_test_scaled)
```

In [79]:
```python
print(accuracy_score(y_test,svc_pred))
print(f1_score(y_test,svc_pred))
print(recall_score(y_test,svc_pred))
print(precision_score(y_test,svc_pred))
```

```
0.95889469342865
0.9580429393329903
0.9375629405840886
0.9794376434583014
```

AFTER RUNNING THE FILE:





7 ADVANTAGES & DISADVANTAGES

Advantages of Credit Approval:

- Credit approval offers convenience by allowing individuals to decide whether the applicants are eligible for credit card
- Random Forest is renowned for its superior propensity to correctly forecast outcomes. It creates a stable and trustworthy model by combining the predictions of various decision trees. This is especially helpful for credit card prediction, where it's critical to accurately classify fraudulent and legitimate transactions.
- Credit approval provides a safety net during emergencies when immediate funds are needed. It allows individuals to handle unforeseen expenses or cover temporary cash flow gaps.
- Machine learning algorithms can quickly process massive volumes of data. They can process large numbers of transactions in real time or almost real time, giving prompt results. On the other hand, classifying credit card transactions manually by people may take a lot longer, especially when dealing with a big volume of transactions.

Disadvantages of Credit Approval:

- The model is not 100% accurate hence may give way to dishonest users may also get the credit card approval
- Credit approval is the potential to accumulate debt. Overspending or failing to manage credit responsibly can lead to a cycle of debt, high-interest payments, and financial stress.
- For credit card acceptance prediction algorithms, outliers, particularly fraudulent transactions that greatly vary from typical trends, might be challenging. If outliers are not properly handled, the model can have trouble correctly identifying and categorising them. Increased false positives or false negatives (approving fraudulent transactions) might come from this, both of which have detrimental effects on both customers and financial institutions.

8 APPLICATIONS

Credit card approval prediction has several practical applications, including:

1. Risk assessment: To determine the risk involved in approving credit card applications, financial institutions, and credit card firms utilize prediction models. These models can calculate the risk that a candidate will miss a payment on a credit card by looking at numerous variables like income, credit history, employment status, and debt-to-income ratio. This makes it easier for lenders to decide whether to accept or refuse credit card applications.

2. Simplifying the application process: Automating predictive models can speed up and improve the efficiency of the credit card acceptance process. These models can instantly assess an applicant's creditworthiness and deliver an instant decision by utilizing historical data and machine learning techniques.

3. Fraud detection: Prediction models for credit card approval can be used to spot applications that might be fraudulent. These algorithms can identify high-risk applications for additional inquiry by examining patterns and anomalies in application data, such as conflicting information or suspicious behavior. This guards against fraud and safeguards both the credit card firm and its clients.

4. Customised offers and credit limits: Predictive models can help determine the credit card offers and credit limits that are most appropriate for applicants. These models can offer suitable credit card products and allocate reasonable credit limits based on the specific needs of applicants by taking into account several variables like income, credit history, and spending habits. The chance of card use rises thanks to this personalised strategy, which also improves consumer happiness.

5. Portfolio management: By analysing the risk profile of the current credit card client base, credit card approval prediction can also be employed in portfolio management. Predictive algorithms can identify consumers who may be more likely to default or become delinquent by examining past data and customer behaviour. Credit card firms can manage their portfolios proactively, take the required steps to reduce risks, and maximise their credit card business with the help of this information.

6. Marketing and customer acquisition: By identifying potential credit card applicants who are likely to be approved, predictive algorithms can help with focused marketing initiatives. These models can divide the market and find the most profitable consumer segments by examining various demographic, financial, and behavioural data.

9 CONCLUSION

In conclusion, we analysed previous works and addressed the issue of imbalance in the dataset through the utilisation of the SMOTE technique. We trained our model using eight classification algorithms, namely Decision tree, Random forest, CatBoost, Xgboost, MLP classifier, SVM, Logistic regression, and AdaBoost classifier. To evaluate the performance of these classifiers, we employed metrics such as accuracy, precision, recall, and F1-score. Based on the evaluation , it has been found that Random forest classifier was the best classifier which gave an accuracy of 99.87%. Therefore Random forest classifier was used for deployment.

10 FUTURE SCOPE

The model can further be improved by enhancing the dataset.The dataset provided is highly imbalanced.In this project only machine learning algorithms are used widely. By using neural network and tensors the model might result in higher accuracy. Establish partnerships and collaborations with financial institutions, such as banks and credit unions, to expand the project's reach and provide seamless integration with their existing systems which aids in updated datasets regularly.

11 BIBLIOGRAPHY

[1] Duan, Lei. "Performance evaluation and practical use of supervised data mining algorithms for credit card approval." In *2020 International Conference on Computing and Data Science (CDS)*, pp. 251-254. IEEE, 2020.

[2] Markova, Maya. "Credit card approval model: An application of deep neural networks." In *AIP Conference Proceedings*, vol. 2321, no. 1, p. 030023. AIP Publishing LLC, 2021.

[3] Yasasvia, Pathipati, and S. Magesh Kumarb. "Improve Accuracy in Prediction of Credit Card Approval Using Novel XGboost Compared with Random Forest." (2022).

[4] Dar, M. H., & Kumar, N. Using Machine Learning Methods to Forecast Credit Card Approvals.

[5] DAVIS, R. H., Edelman, D. B., & Gammerman, A. J. (1992). Machine-learning algorithms for credit-card applications. IMA Journal of Management Mathematics, 4(1), 43-51.

[6] Islam, M. J., Wu, Q. J., Ahmadi, M., & Sid-Ahmed, M. A. (2007, November). Investigating the performance of naive-bayes classifiers and k-nearest neighbour classifiers. In 2007 international conference on convergence information technology (ICCIT 2007) (pp. 1541-1546). IEEE.

[7] Husejinovic, Admel, Nermina Durmić, and Samed Jukić. "Application of Business Intelligence in Decision Making for Credit Card Approval." *Journal of Intelligence Studies in Business* 12.2 (2022): 54-64

[8] Zarnaz, Zaima, Dipannita Biswas, and KM Azharul Hasan. "Credit Card Approval Prediction by Non-negative Tensor Factorization." *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*. IEEE, 2021.

[9] Shukla, A., A. Mishra, and M. Gwalior. "Design of credit approval system using artificial neural network: a case study." *Int. J. Eng. Res. Comput. Sci. Eng* 4.1 (2017): 1-6.

[10] Ipin Sugiyarto,Umi Faddillah,Bibit Sudarsono"Performance Comparison of Data Mining Algorithm to Predict Approval of Credit Card"Journal Publications & Informatics Engineering Research,Volume 4, Number 1, October 2019
[11] Erkan Ilgun, Ensar Mekiš, Emina Mekiš."Application of Ann in Australian Credit Card Approval"European Researcher, 2014, Vol.(69), № 2-2
[12] Md. Golam Kibria and Mehmet Sevkli"Application of Deep Learning for Credit Card Approval: A Comparison with Two Machine Learning Techniques"International Journal of Machine Learning and Computing, Vol. 11, No. 4, July 2021

APPENDIX

A. Source Code

https://drive.google.com/drive/folders/1vQXYhp9n1VBhHmCGbb7BLQl4PrcwV10R?usp=sharing