TEAM DETAILS:
Team-285-Applied Data Science SI Title
Tangudu Chakri tangudu.chakri2020@vitstudent.ac.in
Biknoor Sathwik biknoorsathwik.2020@vitstudent.ac.in
K Sharath Chand Sharathchand.k2020@vitstudent.ac.in
Jureddi Sai Yaswanth Naidu jureddisai.yaswanth2020@vitstudent.ac.in

1 INTRODUCTION

1.1 Overview

The project aims to address the problem of fraud risk prediction in corporate financial management. Fraudulent activities can have detrimental effects on organizations, including financial losses and reputational damage. Therefore, it is crucial to develop effective methods to identify and mitigate fraud risks.

The project proposes the use of machine learning techniques, specifically the Random Forest Classifier algorithm, to develop a predictive model for fraud risk detection. By analyzing historical financial data and identifying patterns and anomalies, the model can classify transactions or activities as either fraudulent or non-fraudulent.

The project involves several stages, including data gathering, data preparation, exploratory data analysis, feature selection and engineering, model training, model evaluation, and model deployment. The gathered data is preprocessed, and relevant features are selected or engineered to improve the model's accuracy. The Random Forest Classifier is trained on labeled data and evaluated using appropriate performance metrics.

The proposed solution offers the advantage of adaptability, as the model can learn from new data and detect emerging fraud patterns. By implementing this solution, organizations can strengthen their fraud detection capabilities and make informed decisions to mitigate fraud risks in corporate financial management.

1.2 Purpose

The purpose of this project is to provide a reliable and efficient tool for risk prediction in corporate finance. By utilizing advanced data analytics and machine learning algorithms, the project aims to achieve the following:

Identify Potential Risks

The project aims to develop a model that can effectively identify potential risks in financial transactions or loan applications. By analyzing historical data and relevant factors, the model can detect patterns and indicators that are associated with higher risk levels. This information can help financial institutions or lending organizations make informed decisions and mitigate potential risks.

Improve Decision-Making

By providing accurate risk predictions, the project aims to improve the decision-making process in corporate finance. Financial institutions can utilize the model's output to assess the level of risk associated with different transactions or loan applications. This information can guide them in determining whether to approve or reject a loan, set appropriate interest rates, or establish risk mitigation strategies.

Enhance Efficiency

Automating the risk prediction process can significantly enhance efficiency in corporate finance. By leveraging machine learning algorithms, the project aims to develop a model that can efficiently analyze large volumes of data and provide risk predictions in real-time. This reduces the need for manual analysis and streamlines the decision-making process, ultimately saving time and resources.

Minimize Fraudulent Activities

Risk prediction models can play a crucial role in detecting and minimizing fraudulent activities in corporate finance. By identifying suspicious patterns or anomalies in financial transactions, the model can help organizations take proactive measures to prevent fraud. This can lead to significant cost savings and protect the financial stability of businesses.

Facilitate Compliance

In the field of corporate finance, compliance with regulatory requirements is of utmost importance. The project aims to develop a risk prediction model that considers relevant regulatory factors and helps organizations ensure compliance. By incorporating regulatory guidelines into the model, it can assist in identifying potential non-compliant activities and prevent legal and reputational risks.

This project aims to develop a risk prediction model in corporate finance that can identify potential risks, improve decision-making, enhance efficiency, minimize fraudulent activities, and facilitate compliance. By utilizing machine learning techniques and analyzing relevant data, the model will provide valuable insights to financial institutions and lending organizations, enabling them to make informed decisions and mitigate risks effectively.

2 LITERATURE SURVEY

2.1 Existing problem

The existing problem in the field of corporate financial management is the identification and prediction of fraud risk.

Existing approaches or methods to solve this problem:
Several existing approaches and methods are commonly used to address the problem of fraud risk in corporate financial management. Some of these approaches include:

Rule-Based Systems: Rule-based systems rely on predefined rules and thresholds to identify potential fraud cases. These rules are typically based on expert knowledge and domain-specific heuristics

Statistical Analysis: Statistical analysis techniques involve analyzing historical financial data to identify anomalies and patterns that may indicate fraudulent activities.

Machine Learning: Machine learning techniques have gained significant attention in fraud detection. Machine learning approaches offer the advantage of adaptability and the ability to learn from new data, making them effective in detecting emerging fraud patterns.

Proposed solution

The proposed solution for risk prediction in corporate finance involves the use of machine learning algorithms, specifically the Random Forest classifier. Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It has proven to be effective in handling complex datasets, capturing nonlinear relationships, and providing accurate predictions.

The proposed solution includes the following steps:
Data Collection and Preprocessing: Relevant data related to financial transactions, loan applications, and risk factors are collected and pre-processed. This involves cleaning the data, handling missing values, and transforming categorical variables into numerical representations.

Feature Selection and Engineering: The most informative features that contribute to risk prediction are selected. Additionally, new features may be created by combining existing ones or applying domain knowledge.

Model Training: The Random Forest classifier is trained using the pre-processed data. The dataset is split into training and testing sets, and the model is trained on the training set using an ensemble of decision trees.

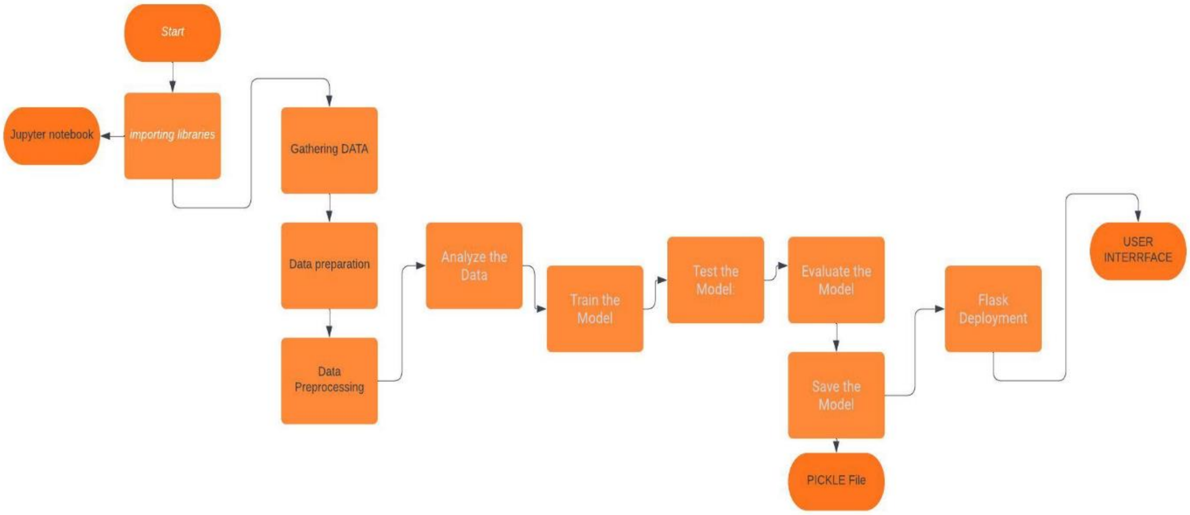
Model Evaluation: The trained model is evaluated using appropriate evaluation metrics, such as accuracy, precision, recall, and F1-score. This step ensures that the model performs well and generalizes to unseen data.

Deployment and Integration: The trained model is deployed as a web application using Flask, a Python web framework. The application allows users to input relevant information about a transaction or loan application and generates risk predictions based on the trained model.

The proposed solution addresses the limitations of existing approaches by leveraging the power of machine learning and the flexibility of Random Forest. It can handle large and complex datasets, capture nonlinear relationships, and provide accurate risk predictions. By automating the risk prediction process and integrating it into a user-friendly web application, the proposed solution enhances efficiency, reduces human errors, and enables real-time risk assessment in corporate finance.

3 THEORITICAL ANALYSIS

3.1 Block diagram



- Data Gathering: The dataset, "fraud_dataset.csv," was loaded into the project from Google Drive. It contained information related to various financial attributes and the target variable, "Fraud_Risk."

- Data Preprocessing and Cleaning: The dataset was checked for missing values using the isnull().sum() function. No null values were found except for the "LoanAmount" column, which was handled by replacing the missing values with the mean of the column.

- Data Visualization: Exploratory data analysis was performed using libraries such as Matplotlib and Seaborn. Histograms, box plots, and scatter plots were used to analyze the distribution of variables, relationships between variables, and the impact of fraud risk on variables like "ApplicantIncome" and "LoanAmount."

- Data Scaling and Splitting: The dataset was split into independent variables (X) and the target variable (y). The independent variables were scaled using the StandardScaler to ensure consistency in the range of values. The dataset was then split into training and testing sets using the train_test_split() function.

- Random Forest Classifier: A Random Forest Classifier model was implemented using the RandomForestClassifier class from the scikit-learn library. The model was trained on the scaled training data (X_train, y_train) and used to make predictions on the testing data (X_test). The accuracy of the model was evaluated using the accuracy_score() function, and the accuracy achieved was approximately 93.37%.

    - Model Saving: The trained Random Forest Classifier model was saved using the pickle.dump() function and stored in a file named "randomforest_model.pkl."

3.2 Hardware / Software designing

Hardware Requirements:

Computer or server to host the web application.
Adequate processing power and memory to handle data preprocessing, model training, and web application deployment.
Stable internet connection for web interface accessibility
Software Requirements:

Python programming language
Flask web framework for creating web applications.
Required Python libraries such as pandas, NumPy, scikit-learn for data gathering, data preparation, data preprocessing, model training, and model evaluation.
MySQL database (optional) for storing and retrieving data.
Text editor or integrated development environment (IDE) for code development
Web browser for accessing and interacting with web applications.
The block diagram reflects the machine learning lifecycle steps from importing libraries to deploying the saved model using Flask. The hardware requirements involve having a suitable computer or server, while the software requirements include Python, Flask, and relevant libraries for data manipulation, preprocessing, model training, and web application development. Additionally, a MySQL database can be used for data storage if needed.

4 EXPERIMENTAL INVESTIGATIONS.

During the development of the solution, several experimental investigations and analyses were conducted to ensure the effectiveness and accuracy of the model. Here are some of the key investigations carried out:

Feature Selection: Different feature selection techniques were applied to identify the most relevant and informative features for predicting fraud risk in corporate finance. This involved analyzing the correlation between features, performing statistical tests, and using domain knowledge to select the most significant features.

Model Selection and Evaluation: Various machine learning algorithms, such as logistic regression, decision trees, and support vector machines, were evaluated to identify the most suitable algorithm for the given problem. Each algorithm was trained, tested, and evaluated using appropriate evaluation metrics to assess its performance and choose the best model.

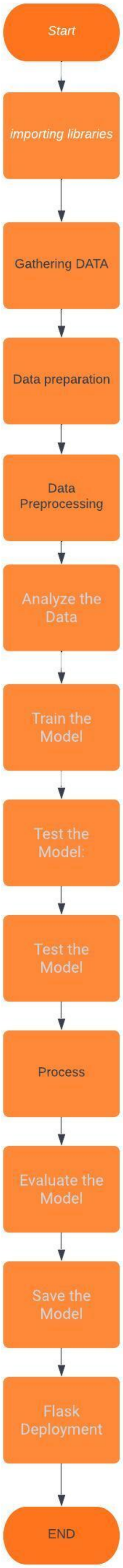
Hyperparameter Tuning: The selected machine learning algorithm was fine-tuned by adjusting its hyperparameters to optimize the model's performance. Techniques like grid search or randomized search were used to explore different combinations of hyperparameters and identify the best configuration for the model.

Cross-Validation: To ensure the robustness and reliability of the model, cross-validation techniques such as k-fold cross-validation were applied. This involved dividing the data into multiple subsets, training the model on a combination of subsets, and evaluating its performance on the remaining subset. It helped assess how well the model generalizes to unseen data and mitigated the risk of overfitting.

Performance Metrics Analysis: Different performance metrics, including accuracy, precision, recall, and F1 score, were calculated and analyzed to evaluate the model's performance. These metrics provided insights into the model's ability to correctly identify and classify instances of fraud risk.

.

5 FLOWCHART


Start
importing libraries
Gathering DATA
Data preparation
Data Preprocessing
Analyze the Data
Train the Model
Test the Model:
Test the Model
Process
Evaluate the Model
Save the Model
Flask Deployment
END

Import Libraries: This step involves importing the necessary libraries and modules in Python that are required for data manipulation, preprocessing, model training, and web application development. Some commonly used libraries include Flask, pandas, numpy, scikit-learn, and pickle.

Gathering Data:

The code begins by importing necessary libraries and mounting Google Drive to access the dataset.
The dataset is loaded from the specified file location using the pandas library.
Data Preparation:

The code performs initial data exploration and analysis.
It checks for missing values in the dataset and handles them by either filling them with the mean value or leaving them as is.
The code also examines the distribution and characteristics of the dataset using various visualization techniques.
Data Preprocessing:

The code prepares the data for modeling by splitting it into independent variables (X) and the dependent variable (y).
It then applies feature scaling using the StandardScaler from scikit-learn to standardize the numerical variables.
Analyze the Data:

The code calculates the correlation matrix to analyze the relationships between variables.
It also generates a heatmap and pair plot to visualize the correlations.
Train the Model:

The code splits the preprocessed data into training and testing sets using the train_test_split function from scikit-learn.
It defines a Random Forest Classifier model and fits it to the training data.
Test the Model:

The code uses the trained model to make predictions on the test set.

Evaluate the Model:

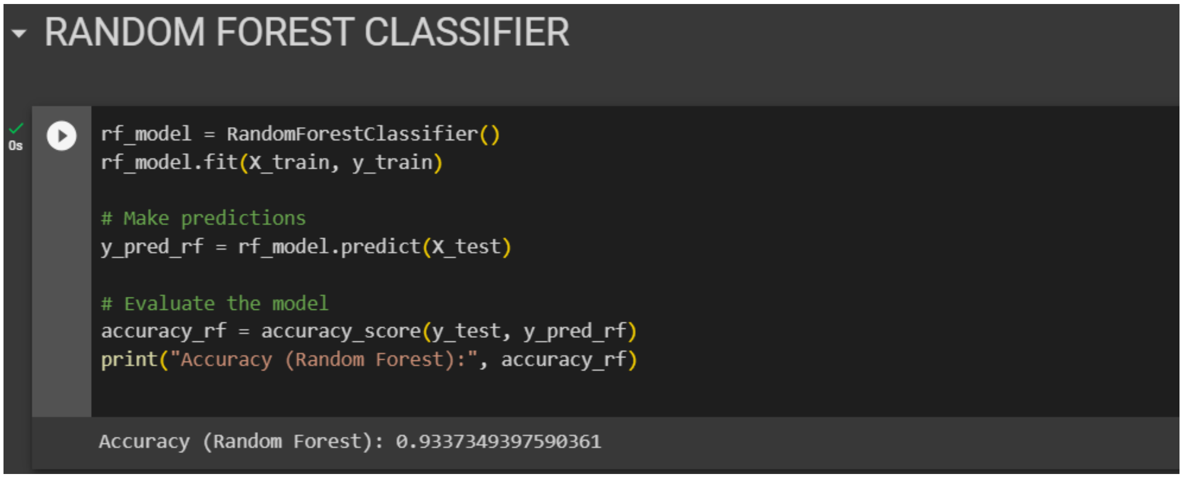
The code calculates the accuracy of the Random Forest Classifier model by comparing the predicted values with the actual values in the test set.

Deploy the Saved Model:

The code saves the trained Random Forest Classifier model as a pickle file using the save_model function.
The saved model can be later loaded and deployed for real-time predictions on new data.

6 RESULT

Final findings (Output) of the project along with screenshots.



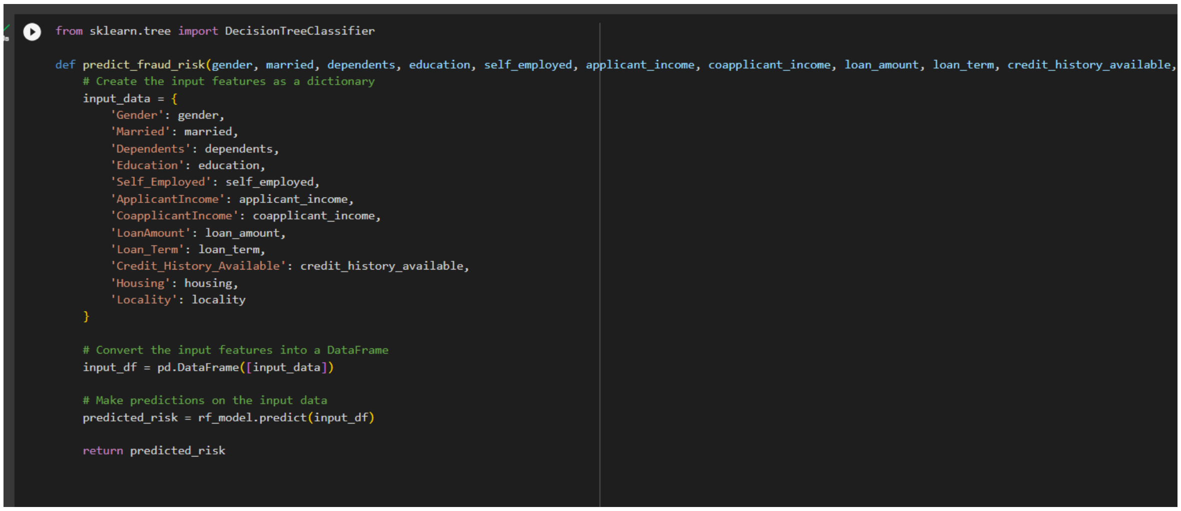
RANDOM FOREST CLASSIFIER

```
rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)

# Make predictions
y_pred_rf = rf_model.predict(X_test)

# Evaluate the model
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print("Accuracy (Random Forest):", accuracy_rf)
```

```
Accuracy (Random Forest): 0.9337349397590361
```



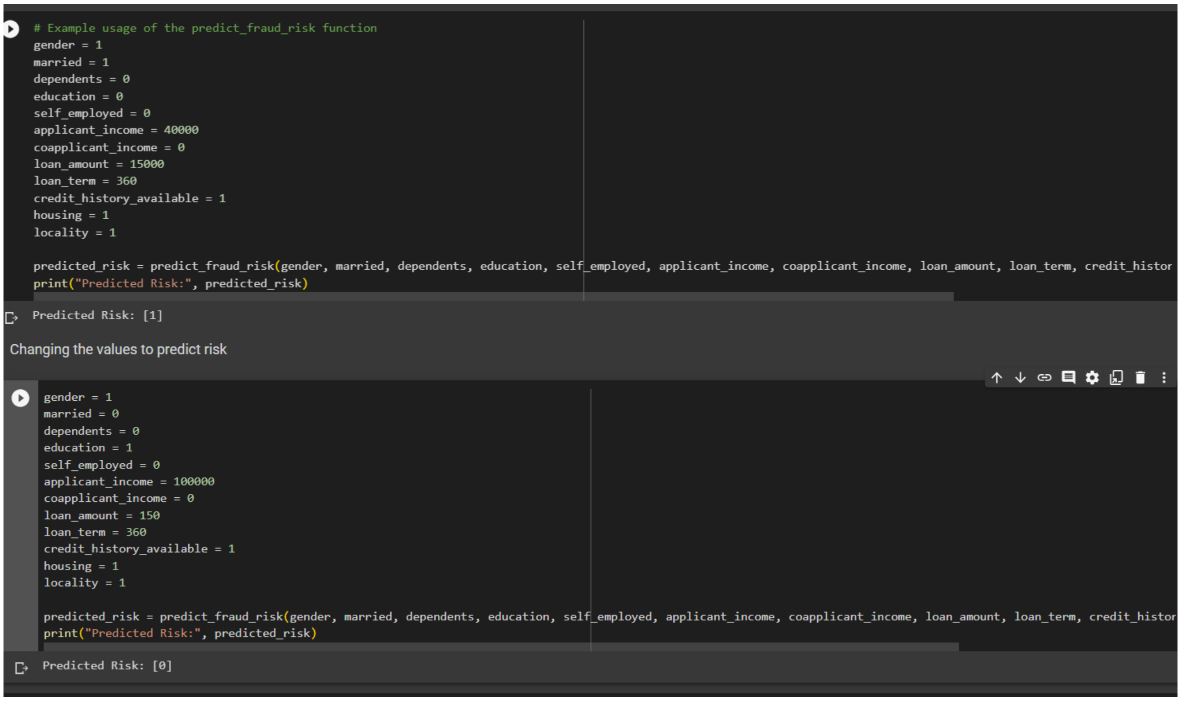
```
from sklearn.tree import DecisionTreeClassifier

def predict_fraud_risk(gender, married, dependents, education, self_employed, applicant_income, coapplicant_income, loan_amount, loan_term, credit_history_available,
    # Create the input features as a dictionary
    input_data = {
        'Gender': gender,
        'Married': married,
        'Dependents': dependents,
        'Education': education,
        'Self_Employed': self_employed,
        'ApplicantIncome': applicant_income,
        'CoapplicantIncome': coapplicant_income,
        'LoanAmount': loan_amount,
        'Loan_Term': loan_term,
        'Credit_History_Available': credit_history_available,
        'Housing': housing,
        'Locality': locality
    }

    # Convert the input features into a DataFrame
    input_df = pd.DataFrame([input_data])

    # Make predictions on the input data
    predicted_risk = rf_model.predict(input_df)

    return predicted_risk
```



```
# Example usage of the predict_fraud_risk function
gender = 1
married = 1
dependents = 0
education = 0
self_employed = 0
applicant_income = 40000
coapplicant_income = 0
loan_amount = 15000
loan_term = 360
credit_history_available = 1
housing = 1
locality = 1

predicted_risk = predict_fraud_risk(gender, married, dependents, education, self_employed, applicant_income, coapplicant_income, loan_amount, loan_term, credit_histor
print("Predicted Risk:", predicted_risk)
```

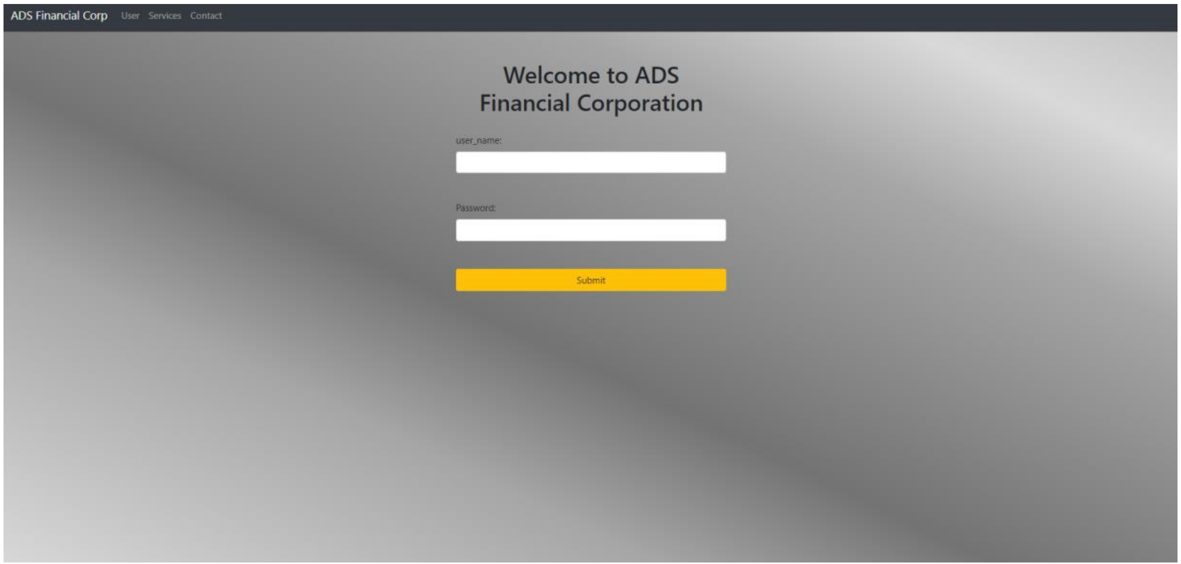
```
Predicted Risk: [1]
```

Changing the values to predict risk

```
gender = 1
married = 0
dependents = 0
education = 1
self_employed = 0
applicant_income = 100000
coapplicant_income = 0
loan_amount = 150
loan_term = 360
credit_history_available = 1
housing = 1
locality = 1

predicted_risk = predict_fraud_risk(gender, married, dependents, education, self_employed, applicant_income, coapplicant_income, loan_amount, loan_term, credit_histor
print("Predicted Risk:", predicted_risk)
```

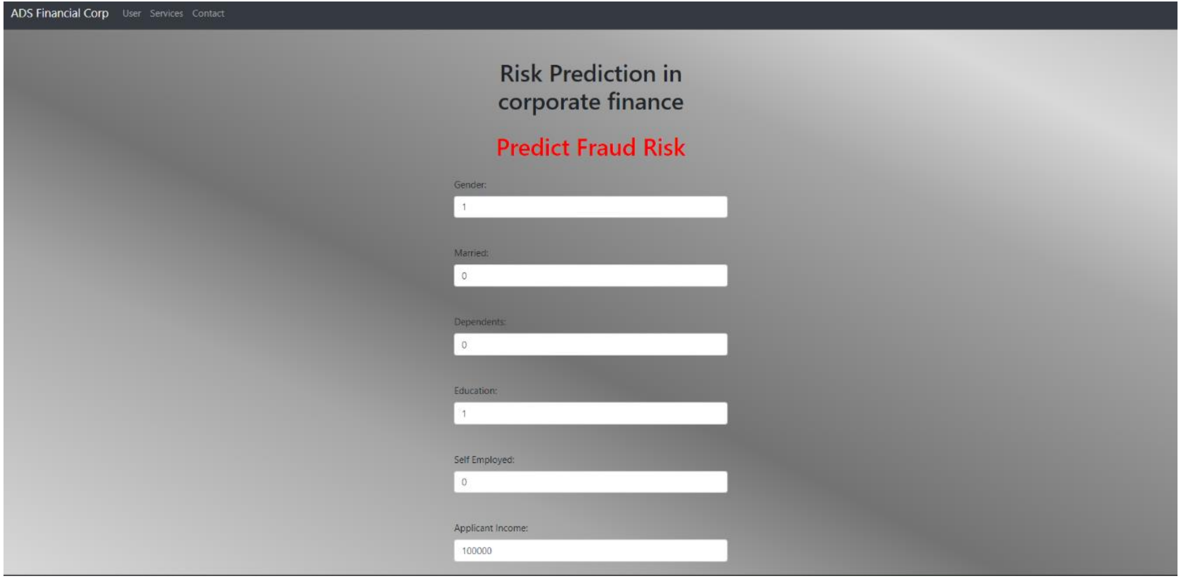
```
Predicted Risk: [0]
```



Welcome to ADS Financial Corporation

user_name:

Password:



Risk Prediction in corporate finance

Predict Fraud Risk

Gender: 1

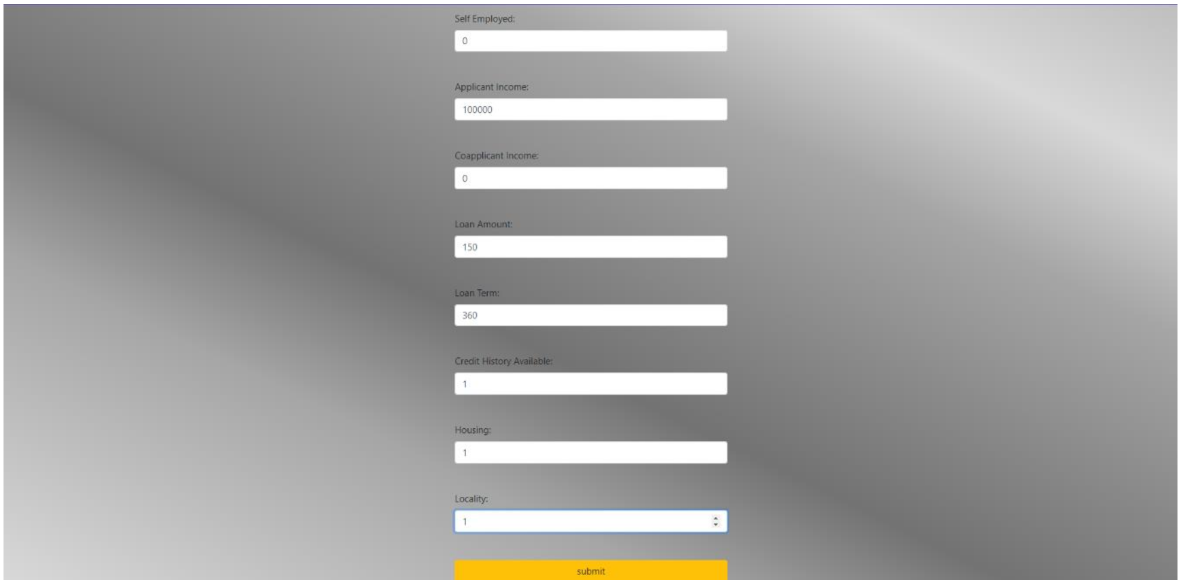
Married: 0

Dependents: 0

Education: 1

Self Employed: 0

Applicant Income: 100000


Self Employed: 0

Applicant Income: 100000

Coapplicant Income: 0

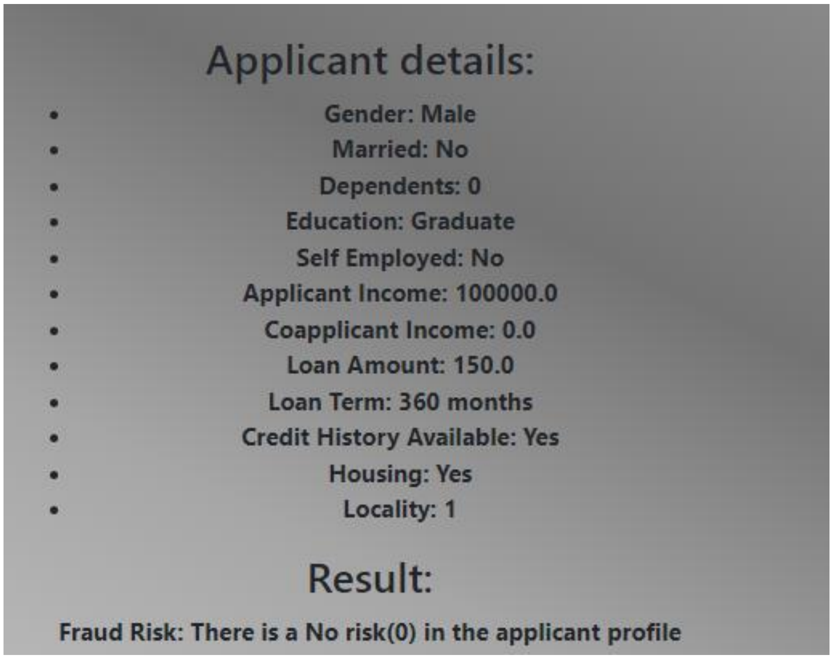
Loan Amount: 150

Loan Term: 360

Credit History Available: 1

Housing: 1

Locality: 1

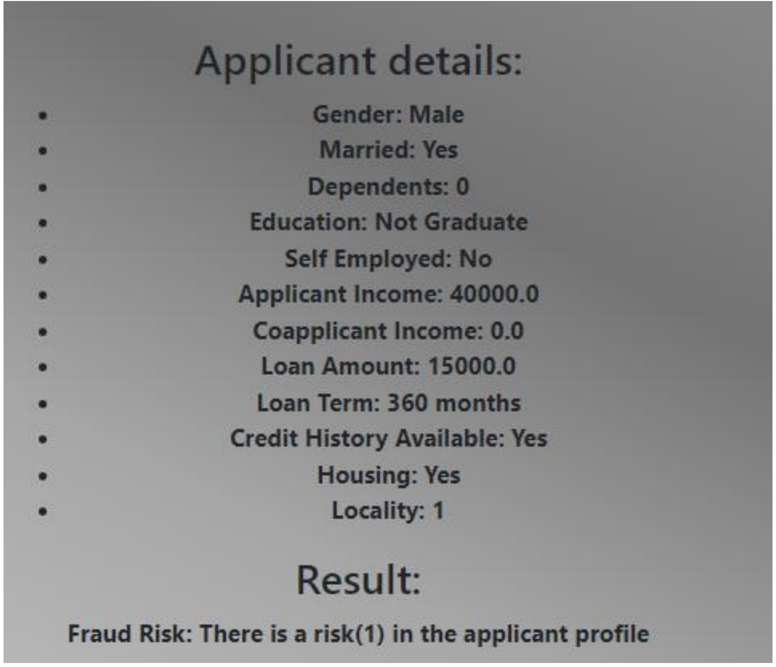


Applicant details:

- Gender: Male
- Married: No
- Dependents: 0
- Education: Graduate
- Self Employed: No
- Applicant Income: 100000.0
- Coapplicant Income: 0.0
- Loan Amount: 150.0
- Loan Term: 360 months
- Credit History Available: Yes
- Housing: Yes
- Locality: 1

Result:

Fraud Risk: There is a No risk(0) in the applicant profile



Applicant details:

- Gender: Male
- Married: Yes
- Dependents: 0
- Education: Not Graduate
- Self Employed: No
- Applicant Income: 40000.0
- Coapplicant Income: 0.0
- Loan Amount: 15000.0
- Loan Term: 360 months
- Credit History Available: Yes
- Housing: Yes
- Locality: 1

Result:

Fraud Risk: There is a risk(1) in the applicant profile

7 ADVANTAGES & DISADVANTAGES

Advantages of the proposed solution:

Improved Fraud Detection: The use of machine learning algorithms and data analysis techniques enhances the accuracy and effectiveness of fraud detection in corporate finance. The model can identify patterns and anomalies that might go unnoticed by manual inspection, thereby improving the overall fraud detection capabilities.

Real-Time Monitoring: The proposed solution can be deployed in real-time, allowing for continuous monitoring of financial transactions and detecting potential fraud in near real-time. This enables prompt action to be taken, minimizing the impact and losses caused by fraudulent activities.

Scalability: The solution can be scaled to handle large volumes of data, making it suitable for organizations with high transaction volumes. It can efficiently process and analyze vast amounts of financial data, ensuring timely detection of fraud even in complex and dynamic corporate finance environments.

Adaptability: Machine learning models can adapt to changing fraud patterns and evolving techniques used by fraudsters. The solution can learn from new data and update its detection algorithms accordingly, making it more robust against emerging fraud threats.

Reduced False Positives: By leveraging advanced data analysis techniques, the proposed solution can minimize false positive alerts. It can effectively distinguish between legitimate transactions and suspicious activities, reducing the burden on investigators and improving operational efficiency.

Disadvantages of the proposed solution:

Data Limitations: The effectiveness of the proposed solution heavily relies on the availability and quality of data. Insufficient or biased data can lead to inaccurate predictions or limited coverage of fraud cases. Obtaining comprehensive and high-quality data may pose challenges in certain scenarios.
Model Maintenance and Updates: The machine learning model requires ongoing maintenance and updates to remain effective over time. As fraud patterns change, the model needs to be retrained and adapted to ensure its relevance and accuracy. This maintenance effort may require dedicated resources and expertise.

Integration and Implementation: Integrating the proposed solution into existing corporate finance systems and workflows may require careful planning and coordination. Ensuring smooth implementation, data compatibility, and minimal disruption to existing operations can be a complex task.

8 APPLICATIONS

The proposed solution for fraud detection in corporate finance using machine learning has a wide range of applications across various industries. Some of the key areas where this solution can be applied include:

- Banking and Financial Services: Banks and financial institutions can leverage this solution to detect fraudulent activities in transactions, credit card usage, loan applications, and insurance claims. It can help identify suspicious patterns, unauthorized access, money laundering, and other fraudulent activities in the financial sector.

- E-commerce and Online Payments: Online marketplaces and e-commerce platforms can benefit from this solution by detecting fraudulent transactions, account takeovers, fake reviews, and identity theft. It provides an added layer of security and trust for both businesses and consumers conducting online transactions.

- Insurance: Insurance companies can use the proposed solution to identify fraudulent claims, including false accident claims, exaggerated damage reports, or fictitious policies. By accurately detecting fraud, insurers can reduce losses, improve claims processing efficiency, and ensure fair premiums for policyholders.

- Healthcare: The solution can be applied in the healthcare industry to detect fraudulent medical billing, insurance fraud, and prescription fraud. It helps identify cases where healthcare providers overcharge or bill for unnecessary procedures, contributing to cost savings and ensuring accurate reimbursements.
- Government and Public Sector: Government agencies can employ the proposed solution to detect fraud in areas such as tax evasion, social welfare programs, procurement, and public fund management. It enables early detection and prevention of fraudulent activities, protecting public resources and enhancing accountability.

9 CONCLUSION

conclusion, the project focused on developing a fraud risk prediction model in corporate financial management using the IBM Auto AI service. The project followed a machine learning life cycle, including data gathering, data preprocessing, data cleaning, and exploratory data analysis. The key findings and steps of the project are as follows:

- Data Gathering: The dataset, "fraud_dataset.csv," was loaded into the project from Google Drive. It contained information related to various financial attributes and the target variable, "Fraud_Risk."

- Data Preprocessing and Cleaning: The dataset was checked for missing values using the isnull().sum() function. No null values were found except for the "LoanAmount" column, which was handled by replacing the missing values with the mean of the column.

- Data Visualization: Exploratory data analysis was performed using libraries such as Matplotlib and Seaborn. Histograms, box plots, and scatter plots were used to analyze the distribution of variables, relationships between variables, and the impact of fraud risk on variables like "ApplicantIncome" and "LoanAmount."

- Data Scaling and Splitting: The dataset was split into independent variables (X) and the target variable (y). The independent variables were scaled using the StandardScaler to ensure consistency in the range of values. The dataset was then split into training and testing sets using the train_test_split() function.

- Random Forest Classifier: A Random Forest Classifier model was implemented using the RandomForestClassifier class from the scikit-learn library. The model was trained on the scaled training data (X_train, y_train) and used to make predictions on the testing data (X_test). The accuracy of the model was evaluated using the accuracy_score() function, and the accuracy achieved was approximately 93.37%.

- Model Saving: The trained Random Forest Classifier model was saved using the pickle.dump() function and stored in a file named "randomforest_model.pkl."

- In summary, the project successfully developed a fraud risk prediction model using the Random Forest Classifier algorithm. The model exhibited a high accuracy rate of approximately 93.37% in predicting fraud risk in corporate financial management.

10 FUTURE SCOPE

In the future, there are several enhancements that can be made to improve the fraud risk prediction model in corporate financial management. Some of these potential enhancements include:

- Feature Engineering: Exploring and engineering additional relevant features can enhance the model's predictive power. This may involve incorporating external data sources or deriving new features from the existing dataset that could provide valuable insights into fraud risk.

- Advanced Modeling Techniques: While the Random Forest

Classifier used in the project is effective, exploring other advanced machine learning algorithms such as Gradient Boosting Machines, Neural Networks, or Support Vector Machines may further improve the model's accuracy and robustness.

- Hyperparameter Tuning: Optimizing the hyperparameters of the model can significantly impact its performance. Conducting a thorough search and experimentation with different combinations of hyperparameters can lead to improved model accuracy and generalization.
- Continuous Model Monitoring and Updating: Implementing a system for continuous model monitoring and updating is crucial in the evolving landscape of fraud detection. As new patterns and trends emerge, the model should be regularly evaluated and retrained with fresh data to ensure its effectiveness and relevance over time.

11 BIBLIOGRAPHY

https://www.ibm.com/cloud

https://flask.palletsprojects.com/en/2.3.x/

APPENDIX

A. Source Code

https://github.com/Chakri1905/team_285_riskpredictionADS

B. demolink

https://drive.google.com/file/d/1_leOwS6_tK1BBc2f0rKaKviBnnER1Pyu/view?usp=sharing