CREDIT CARD APPROVAL PREDICTION USING MACHINE LEARNING

NAME : DILLI GANESH

CYBERNAUT PROJECT INTERN

MONTH 2 MAJOR PROJECT (TEAM 11)

INTRODUCTION

In today's financial world, banks and financial institutions receive a very large number of credit card applications every day. Manually verifying each application is time-consuming, costly, and prone to human error. To overcome this problem, machine learning techniques can be used to automate the credit approval process by analyzing applicant details and predicting whether a customer is likely to be approved or rejected.

This project, **Credit Card Approval Prediction**, focuses on building an intelligent system that predicts credit card approval status using historical customer data. Multiple machine learning models are trained and compared to identify the most accurate and reliable model. The project also handles data imbalance and includes fairness analysis to ensure unbiased predictions.

OBJECTIVES OF THE PROJECT

The main objectives of this project are:

- To analyze credit card application and credit history data
- To preprocess and clean large real-world datasets
- To handle imbalanced datasets using SMOTE
- To train and evaluate multiple machine learning models
- To compare model performance using accuracy and visualizations
- To select the best performing model for prediction
- To provide an interactive prediction system for new inputs

DATASET DESCRIPTION

This project uses **two real-world datasets**:

APPLICATION RECORD DATASET

This dataset contains demographic and financial information of customers who applied for a credit card. Some important attributes include:

- Gender
- Ownership of car and house
- Number of children
- Annual income
- Employment details
- Family status

Dataset Size: - Rows: 438,557 - Columns: 18

MERGE APPLICATION AND CREDIT RECORD

```
data = pd.merge(app, credit_summary, on='ID', how='inner')
data.head()
```

	ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	DAYS_BIRT
0	5008804	M	Y	Y	0	427500.0	Working	Higher education	Civil marriage	Rented apartment	-12005.
1	5008805	M	Y	Y	0	427500.0	Working	Higher education	Civil marriage	Rented apartment	-12005.
2	5008806	M	Y	Y	0	112500.0	Working	Secondary / secondary special	Married	House / apartment	-21474.
3	5008808	F	N	Y	0	270000.0	Commercial associate	Secondary / secondary special	Single / not married	House / apartment	-19110.
4	5008809	F	N	Y	0	270000.0	Commercial associate	Secondary / secondary special	Single / not married	House / apartment	-19110.

Next steps: Generate code with data | New interactive sheet

CREDIT RECORD DATASET

This dataset contains monthly credit repayment history of customers. The STATUS column indicates repayment behavior.

- ‘C’, ‘0’ → Good credit behavior
- ‘1’ to ‘5’ → Delayed payments or risk

Dataset Size: - Rows: 1,048,575 - Columns: 3

TARGET VARIABLE CREATION

The credit record dataset does not directly contain a target label. Therefore, a new target variable named **Approved** is created based on the credit status.

- Customers with good repayment history are marked as 0
- Customers with poor repayment behavior are marked as 1

The credit records are aggregated customer-wise using the maximum status value to represent overall credit risk.

CREATE TARGET VARIABLE FROM CREDIT RECORD

[5] ✓ 0s

```
credit['STATUS'] = credit['STATUS'].replace({
    'C': 0,
    '0': 0,
    '1': 1,
    '2': 1,
    '3': 1,
    '4': 1,
    '5': 1
})
```

AGGREGATE CREDIT RECORD PER CUSTOMER

[6] ✓ 0s

```
credit['STATUS'] = credit['STATUS'].replace('X', 0)
credit['STATUS'] = pd.to_numeric(credit['STATUS'], errors='coerce')

credit_summary = credit.groupby('ID')['STATUS'].max().reset_index()
credit_summary.rename(columns={'STATUS': 'Approved'}, inplace=True)
```

DATA PREPROCESSING

DATA MERGEING

The application dataset and the aggregated credit record dataset are merged using the common ID column. Only customers present in both datasets are considered for analysis.

REMOVING UNNECESSARY COLUMNS

The ID column is removed after merging, as it does not contribute to prediction.

HANDLING MISSING VALUES

- Categorical columns are filled using **mode**
- Numerical columns are filled using **median**

This ensures that no data is lost during preprocessing.

ENCODING CATEGORICAL VARIABLES

Machine learning models require numerical input. Therefore, categorical variables are converted into numerical form using **Label Encoding**.

FEATURE AND TARGET SPLIT

After preprocessing:

- **Features (X):** All independent variables
- **Target (y):** Approved

This separation allows the models to learn patterns from features and predict the target variable.

CLASS IMBALANCE HANDLING

Initially, the dataset is imbalanced, meaning one class has significantly more samples than the other. This can reduce model performance.

CLASS DISTRIBUTION BEFORE SMOTE

A count plot is used to visualize class imbalance.

SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE is applied to balance the dataset by generating synthetic samples for the minority class.

CLASS DISTRIBUTION AFTER SMOTE

After applying SMOTE, both classes are equally represented.

TRAIN-TEST SPLIT AND FEATURE SCALING

- Dataset is split into **80% training** and **20% testing** data
- Feature scaling is performed using **StandardScaler** to normalize values

This improves model convergence and accuracy.

MACHINE LEARNING MODELS USED

The following machine learning models are trained and evaluated:

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. Gradient Boosting Classifier
5. XGBoost Classifier

Each model is trained using the same dataset to ensure fair comparison

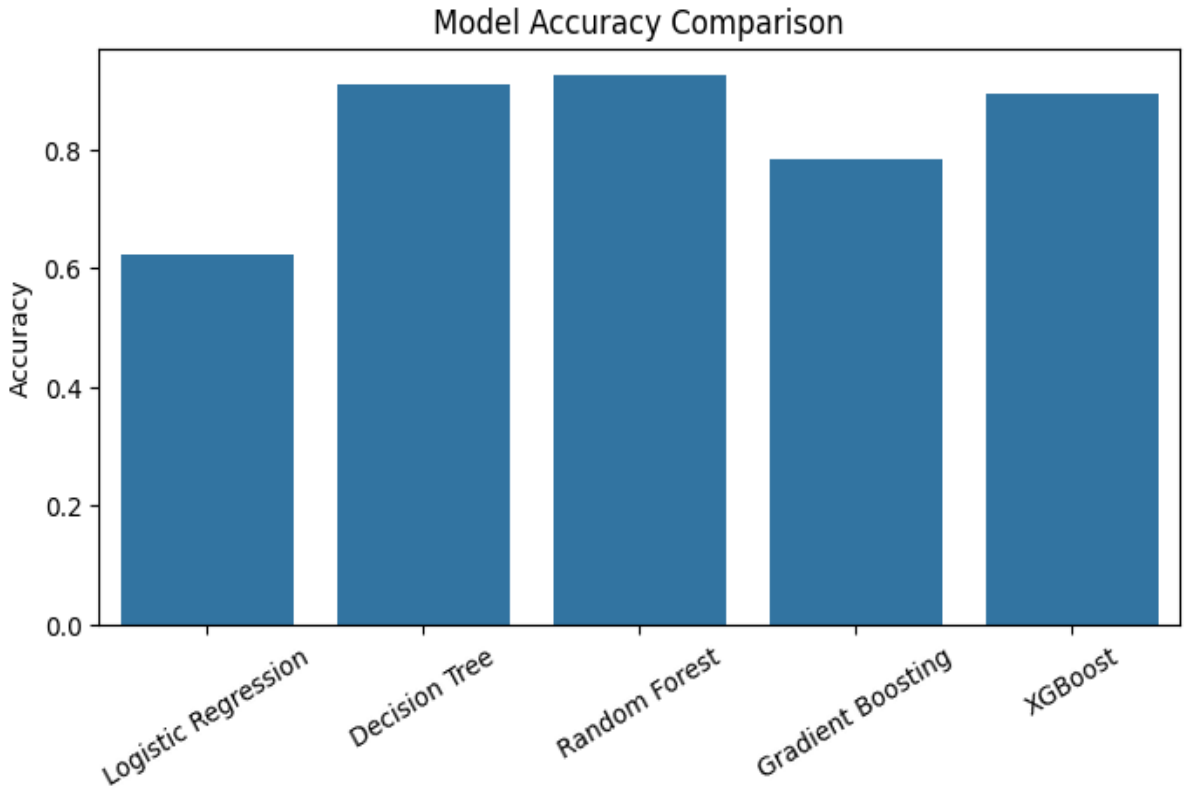
MODEL EVALUATION AND RESULTS

The models are evaluated using:

- Accuracy Score
- Precision
- Recall
- F1-Score
- Classification Report

ACCURACY COMPARISON

Among all models, **Random Forest** achieved the highest accuracy.



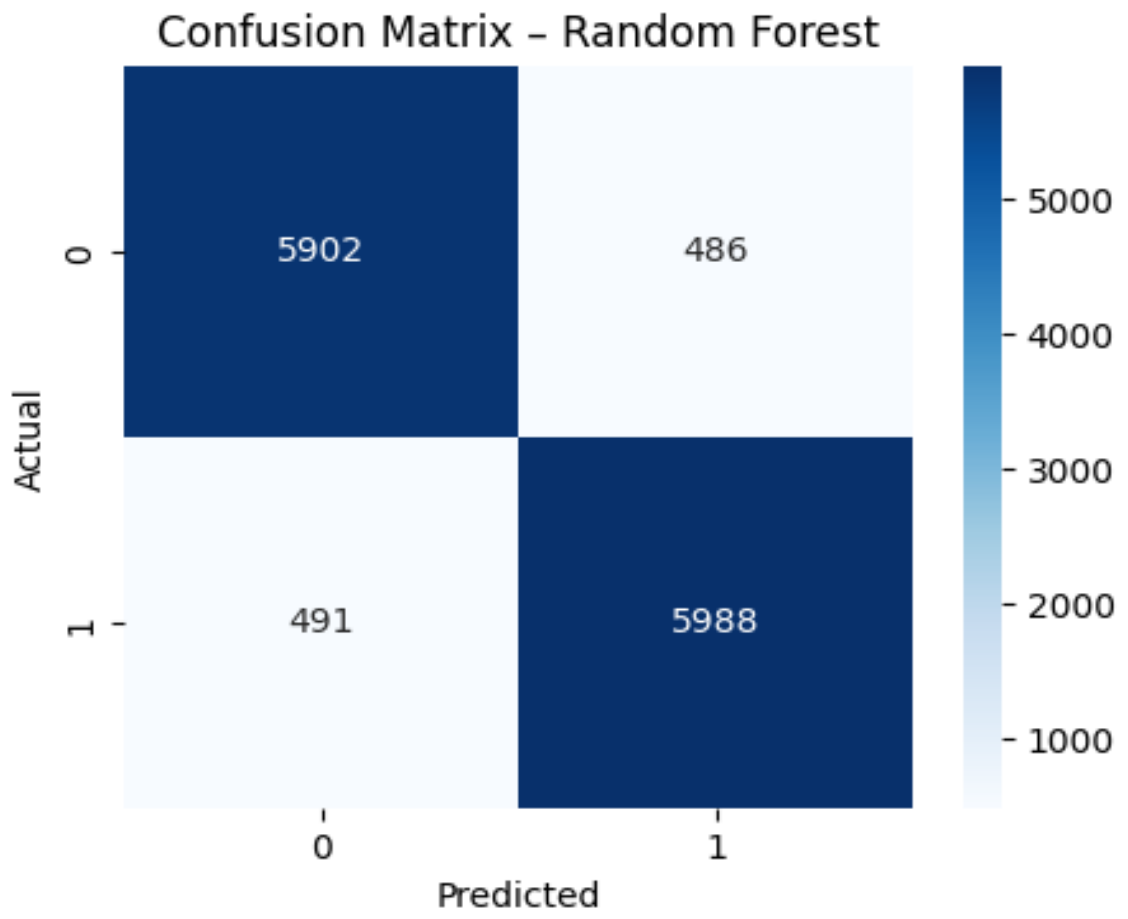
BEST MODEL: RANDOM FOREST

The Random Forest model is selected as the final model due to:

- High accuracy
- Better generalization
- Robustness to overfitting

CONFUSION MATRIX

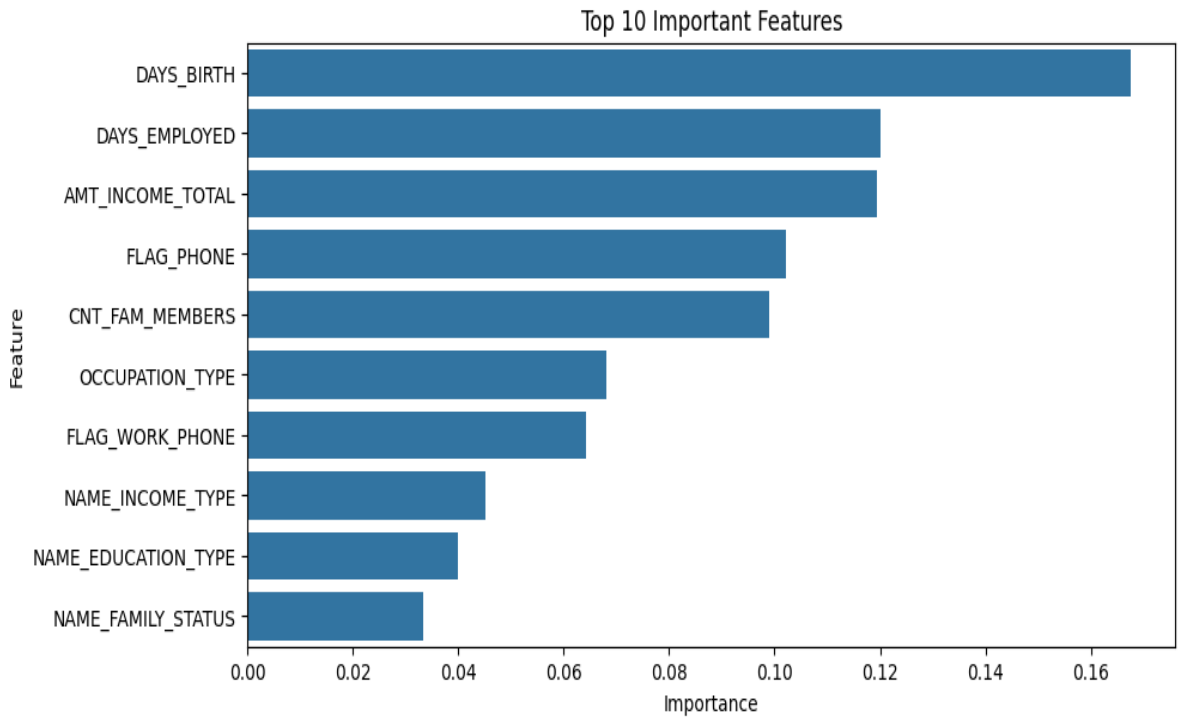
A confusion matrix is plotted to visualize true and false predictions.



FEATURE IMPORTANCE ANALYSIS

Random Forest provides feature importance values that show which attributes influence predictions the most.

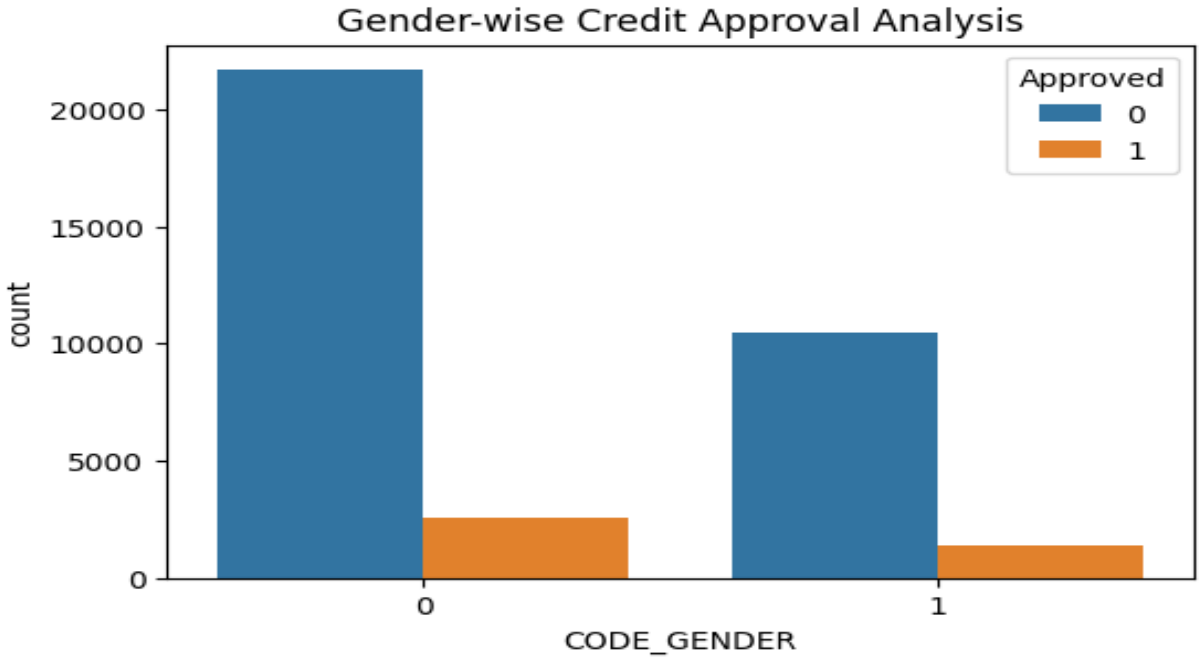
The top 10 most important features are visualized using a bar chart



FAIRNESS ANALYSIS

To ensure the model does not show bias, a fairness analysis is performed based on gender.

- Approval trends are visualized for different genders
- This helps in identifying potential discrimination



INTERACTIVE PREDICTION SYSTEM

An interactive prediction function is created that accepts new applicant details and predicts whether the credit card will be approved or rejected.

Output: - Approved – Rejected

CONCLUSION

This project successfully demonstrates how machine learning can be applied to automate credit card approval decisions. By preprocessing large datasets, handling imbalance, and comparing multiple models, an accurate and reliable prediction system is developed.

The Random Forest model provided the best performance and was used for final predictions. The project also highlights the importance of fairness analysis in financial applications.

FUTURE ENHANCEMENTS

- Deploy the model using a web application
- Include more financial and behavioral features
- Use deep learning techniques for better performance
- Perform advanced bias and explainability analysis

TOOLS AND TECHNOLOGIES USED

- Python
- Pandas, NumPy
- Matplotlib, Seaborn
- Scikit-learn
- XGBoost
- SMOTE (Imbalanced-learn)

REFERENCES

- Scikit-learn Documentation
- Kaggle Credit Card Datasets
- Machine Learning Research Papers