**Loan Application Classification Report**

**Problem Statement:**

Assume you are a loan risk officer at a large bank and you are tasked with determining whether a two-wheeler loan application will be accepted or rejected based on the data shared by the loan applicant and some additional data extracted about them from 3rd party sources.

**1. Approach Taken**

**Data Understanding**

**Data Files:**

* **Assignment\_Train.csv:** Has labeled data with the "Application Status" variable.
* **Assignment\_Test.csv:** Has unlabeled data without the "Application Status" variable.
* **Assignment\_FeatureDictionary.xlsx:** Gives descriptions and details about the variables in the datasets.

**Preprocessing Steps:**

1. **Loading Data:** We put the datasets into pandas DataFrames.
2. **Date Processing:** We changed the APPLICATION LOGIN DATE column to datetime format. We then pulled out Month, Day, and Year details.
3. **Feature Encoding:**We turned categorical features into numbers using LabelEncoder. We fit label encoders on the training data and used them the same way on the test data.
4. **Handling Missing Values:** We filled in missing values using SimpleImputer with the 'most\_frequent' option.
5. **Feature Scaling:** We standardized features using StandardScaler.
6. **Model Selection:** We picked a Random Forest Classifier because it's strong and can handle many types of data.

**Model Training and Evaluation**

1. **Train-Test Split**: The training data was split into training and validation sets.
2. **Model Training**: The Random Forest Classifier was trained on the training set.
3. **Model Evaluation**: Model performance was evaluated using metrics such as accuracy, precision, recall, and F1-score on the validation set.

**Prediction**

* The trained model was used to predict the "Application Status" for the test data.
* Predictions were saved in a CSV file for submission by the name of predictions.csv.

**2. Insights and Conclusions from Data**

**Data Insights**

* **Feature Distribution**: Analysis of categorical and numerical features revealed patterns and distributions, which guided preprocessing and feature engineering steps.
* **Missing Values**: Identified and handled missing values to ensure data quality and completeness.

**Date Features:**

* Extracted month, day, and year from the APPLICATION LOGIN DATE to capture temporal patterns which might affect the loan application status.

**Categorical Encoding:**

* Consistent encoding of categorical features ensured that the model could interpret the test data correctly.

**Feature Scaling:**

* Standardized features to improve model performance by ensuring that all features contribute equally to the prediction.

**Model Performance:**

* The Random Forest Classifier was chosen due to its ability to handle complex relationships and interactions between features.

**3. Performance on Train Dataset**

**Metrics**

* **Accuracy**: This tells us how many loan applications were correctly classified by the model out of all the applications. A higher accuracy means the model is doing a better job overall.
* **Precision**: This measures how many of the applications that the model predicted as accepted were actually accepted. High precision means fewer mistakes in predicting acceptance.
* **Recall**: This measures how many of the applications that were actually accepted were correctly predicted by the model. High recall means the model finds most of the accepted applications.
* **F1-Score**: This is a combination of precision and recall. It helps us understand the model's performance by balancing both precision and recall.

A screen shot of a computer

Description automatically generated

**Results**

* **Accuracy**: The model achieved an accuracy of [insert accuracy here]%, which shows how well it correctly classified loan applications.
* **Precision, Recall, and F1-Score**: The classification report gives a detailed look at how well the model performed in different areas, such as finding and correctly predicting accepted applications.

**4. Conclusion**

**Summary**

* We followed a thorough process for preparing the data and training the model.
* The Random Forest Classifier performed well based on the accuracy and other measures, meaning it is good at predicting loan applications.
* The model was tested on a separate validation set to make sure it works well and doesn’t just fit the training data.

**Next Steps**

* **Model Improvement**: We might try tweaking the model or using different algorithms to see if we can get even better results.
* **Feature Engineering**: Look for other features or ways to improve the data to make the model even more accurate.
* **Cross-Validation**: Check how well the model performs on different parts of the data to ensure it is reliable.