**Loan Risk Classification**

**1. Approach Taken**

**1.1 Data Preparation**

* **Data Loading**:
  + Loaded training and test datasets from CSV files: Assignment\_Train.csv and Assignment\_Test.csv.
* **Feature Selection**:
  + Dropped irrelevant columns such as UID, personal names, and other non-essential features.
  + Retained features that are relevant to predicting the loan status.
* **Feature Engineering**:
  + Converted APPLICATION LOGIN DATE to datetime format and extracted features like Login\_Year, Login\_Month, and Login\_Day to capture temporal aspects.
* **Handling Missing Values**:
  + Used SimpleImputer to fill missing values for both numerical and categorical features.
* **Categorical Encoding**:
  + Applied OneHotEncoder to handle categorical variables, which converted categorical features into a format suitable for machine learning algorithms.

**1.2 Preprocessing**

* **Scaling**:
  + Standardized numerical features using StandardScaler to ensure that all features contribute equally to the model.
* **Pipeline Creation**:
  + Constructed a machine learning pipeline that integrated preprocessing steps with the logistic regression model. This ensures a seamless workflow from data preprocessing to model training and evaluation.

**1.3 Model Training**

* **Model Selection**:
  + Chose LogisticRegression as the classification model due to its simplicity and effectiveness for binary classification tasks.
* **Model Fitting**:
  + Trained the logistic regression model on the processed training data.

**1.4 Evaluation**

* **Performance Metrics**:
  + Evaluated model performance using accuracy, precision, recall, and F1-score on the training data to gauge how well the model fits the data.

**1.5 Prediction and Submission**

* **Test Data Preparation**:
  + Preprocessed the test data using the same steps applied to the training data to ensure consistency.
* **Prediction**:
  + Used the trained model to make predictions on the test data.
* **Submission File**:
  + Created a CSV file (predictions1.csv) with UID and the corresponding predictions for submission.

**2. Insights and Conclusions from Data**

**2.1 Insights**

* **High Model Accuracy**:
  + The logistic regression model achieved a training accuracy of **99.81%**. This high accuracy indicates that the model is effectively distinguishing between loan application statuses.
* **Feature Importance**:
  + The preprocessing steps, such as scaling and encoding, likely contributed to the model's ability to learn effectively from the data.

**2.2 Conclusions**

* **Model Effectiveness**:
  + The logistic regression model demonstrates strong performance, with perfect precision, recall, and F1-scores for both classes.
* **Risk of Overfitting**:
  + The exceptionally high performance on the training data suggests a potential risk of overfitting. The model might perform well on training data but could potentially underperform on unseen data.
* **Feature Relevance**:
  + Temporal features and categorical data related to applicant details and asset information are significant in predicting loan outcomes.

**3. Performance on Training Dataset**

**3.1 Classification Report**

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| 0 | 1.00 | 1.00 | 1.00 | 6677 |
| 1 | 1.00 | 0.99 | 1.00 | 3323 |
| **Accuracy** |  |  | **1.00** | 10000 |
| **Macro Avg** | 1.00 | 1.00 | 1.00 | 10000 |
| **Weighted Avg** | 1.00 | 1.00 | 1.00 | 10000 |

* **Precision**: Measures the proportion of true positives among all positive predictions.
* **Recall**: Measures the proportion of true positives among all actual positives.
* **F1-Score**: Harmonic mean of precision and recall, providing a balance between the two metrics.
* **Accuracy**: Overall proportion of correct predictions.

**3.2 Training Accuracy**

* **Training Accuracy**: **99.81%**

**4. Recommendations for Future Work**

1. **Validation and Cross-Validation**:
   * Test the model on a separate validation set or use cross-validation to confirm that the high performance is not due to overfitting.
2. **Model Experimentation**:
   * Experiment with other models like Gradient Boosting or XGBoost to compare performance and potentially improve results.
3. **Feature Analysis**:
   * Conduct feature importance analysis to identify which features most influence predictions.
4. **Continuous Monitoring**:
   * Implement the model in a production environment and continuously monitor its performance, making adjustments as needed based on new data.

**1.Logistic Regression**

The logistic regression model achieved the following performance metrics on the training dataset:

* **Training Accuracy**: **75.95%**
* **Classification Report**:

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| 0 | 0.82 | 0.82 | 0.82 | 6677 |
| 1 | 0.64 | 0.63 | 0.64 | 3323 |
| **Accuracy** |  |  | **0.76** | 10000 |
| **Macro Avg** | 0.73 | 0.73 | 0.73 | 10000 |
| **Weighted Avg** | 0.76 | 0.76 | 0.76 | 10000 |

**2. Conclusions**

**2.1 Model Effectiveness**

* The model shows a moderate level of effectiveness with an overall accuracy of 75.95%.
* Precision and recall for class 0 (accepted applications) are notably higher than for class 1 (rejected applications). This suggests the model is better at identifying accepted applications than rejected ones.

**2.2 Precision and Recall**

* **Class 0 (Accepted Applications)**:
  + **Precision**: 0.82 – Indicates that when the model predicts an application is accepted, it is correct 82% of the time.
  + **Recall**: 0.82 – Shows that the model correctly identifies 82% of the actual accepted applications.
* **Class 1 (Rejected Applications)**:
  + **Precision**: 0.64 – Indicates that when the model predicts an application is rejected, it is correct 64% of the time.
  + **Recall**: 0.63 – Shows that the model correctly identifies 63% of the actual rejected applications