****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Programming for AI**

**ML Project**

Syed Ahmad Ali SP23-BAI-050

**Aksam Iftikhar**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Random Forest Classifier on Default of Credit Card Clients Dataset**

**1. Introduction**

**Dataset Description:**

* **Dataset Name:** Default of Credit Card Clients Dataset
* **Source:** UCI Machine Learning Repository
* **Key Features:**
  + Client demographic details (e.g., age, education level)
  + Credit and payment history
  + Bill statements and payment amounts
* **Target Variable:** default.payment.next.month (0 = No default, 1 = Default)
* **Size:** 30,000 records with 24 features
* **Missing Values:** No missing data was observed

**2. Methodology**

**Preprocessing Steps:**

1. **Data Loading:** The dataset was loaded using pandas.
2. **Exploratory Data Analysis:**
   * Examined dataset structure, feature statistics, and class distribution.
   * Visualized class imbalance using bar charts.
3. **Handling Class Imbalance:**
   * Applied SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset.
4. **Feature Scaling:**
   * Standardized the features using Standard Scaler to ensure uniform feature scaling.

**Algorithms Applied:**

* **Random Forest Classifier:**
  + Ensemble learning method combining multiple decision trees.

**Optimization Techniques:**

* **RandomizedSearchCV:**
  + Performed hyperparameter tuning on the following parameters:
    - Number of estimators (n\_estimators)
    - Maximum depth (max\_depth)
    - Minimum samples required to split a node (min\_samples\_split)
    - Minimum samples required in a leaf node (min\_samples\_leaf)
    - Maximum features considered for splits (max\_features)
    - Bootstrap sampling (bootstrap)
    - Splitting criterion (criterion)
  + Conducted 3-fold cross-validation with 50 iterations.

**3. Results**

**Best Hyperparameters:**

* The best hyperparameters identified by RandomizedSearchCV:
  + **n\_estimators**: 150
  + **max\_depth**: 30
  + **bootstrap:** False
  + **max\_features:** log2
  + **criterion=** entropy

**Performance Metrics:**

* **Accuracy:** 0.86
* **Classification Report**:
  + Precision: 0.87
  + Recall: 0.82
  + F1-Score: 0.84

**Confusion Matrix:**

|  |  |  |
| --- | --- | --- |
|  | **Predicted: No Default** | **Predicted: Default** |
| **No Default** | 4111 | 553 |
| **Default** | 790 | 3892 |

**Visualizations:**

1. **Class Distribution** (Before and After SMOTE):
   * Bar charts showing balanced class distribution after applying SMOTE.
2. **Confusion Matrix**:
   * Visualized using Confusion Matrix Display.
3. **Performance Metrics**:
   * Bar chart displaying precision, recall, F1-score, and accuracy.

**4. Analysis**

**Insights:**

1. The Random Forest Classifier performed well in predicting credit card defaults, achieving an accuracy of 86%.
2. SMOTE effectively balanced the dataset, improving the model’s performance on the minority class.

**Challenges Faced:**

1. **Class Imbalance**: The original dataset had a significant class imbalance, which was addressed using SMOTE.
2. **Hyperparameter Tuning**: Finding the optimal hyperparameters required computational resources due to the large parameter grid.

**Table:**

**Comparison of Machine Learning Algorithms on Default of Credit Card Clients Dataset**

| **Algorithm** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **Best Hyperparameters** | **Execution Time** | **Remarks** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| KNN | 0.85 | 0.84 | 0.86 | 0.85 | n\_neighbors=5, metric=Euclidean | 7**(s)** | Best overall performance. |
| Logistic Regression | 0.64 | 0.64 | 0.65 | 0.64 | Penalty: l1, Solver: saga, Maximum iterations: 300 | 4.4**(s)** | Fast but Computations Measurements are very bad |
| Random Forest | 0.86 | 0.86 | 0.82 | 0.84 | bootstrap=False, criterion='entropy', max\_depth=30, max\_features='log2', n\_estimators=150) | 8.4**(m)** | Computations are not bad but very much Time Complexity |

**Conclusion:**

The **Random Forest Classifier**, optimized using **RandomizedSearchCV** and applied to the Default of Credit Card Clients Dataset, demonstrated robust performance. With an accuracy of **86%** and balanced class predictions, the model shows promise in real-world applications of credit risk prediction.