

**The Superior University, Lahore**

|  |  |  |  |
| --- | --- | --- | --- |
| Course Title: | AI - LAB | | |
| Programme Name: | BSDS | | |
| Semester: | 3rd | Section: | BSDSM-3A-023 |
| Student’s Name: | Aun Muhammad | | |
| Instructor Name: | Prof. Rasikh Ali | | |

**Project Report**

**Data Loading and Inspection**

1. **Loading the Dataset:** The dataset Churn\_Modelling.csv is loaded using pandas's read\_csv function.
   * The dataset contains various customer attributes, including CustomerId, RowNumber, Surname, and details like Geography, Gender, CreditScore, etc.
2. **Initial Exploration:**
   * churn\_df.head() is used to inspect the first few rows of the dataset to get an understanding of its structure and the features available.

**Data Preprocessing**

1. **Dropping Irrelevant Columns:**
   * The columns CustomerId, RowNumber, and Surname are dropped as they are not useful for the predictive model.
2. **Feature Selection for Preprocessing:**
   * **OneHotEncoder**: The Geography feature is one-hot encoded as it is categorical with multiple categories.
   * **OrdinalEncoder**: The Gender feature is ordinal (male/female) and is encoded accordingly.
   * **StandardScaler**: Continuous features such as CreditScore, Age, Balance, and EstimatedSalary are standardized using the StandardScaler to bring them onto a common scale.
3. **Splitting the Data:**
   * The data is split into training and testing sets using an 80-20 split, with train\_test\_split from sklearn.model\_selection. The target variable is Exited, which indicates whether the customer left the bank or not.
4. **ColumnTransformer:**
   * A ColumnTransformer is used to apply the appropriate transformations to the selected columns: one-hot encoding for Geography, ordinal encoding for Gender, and standard scaling for continuous variables.
   * The transformed data is then converted back into a DataFrame for easier handling.

**Model Building and Evaluation**

1. **Logistic Regression Model:**
   * A LogisticRegression model is trained on the preprocessed training data.
   * Predictions are made on the test set, and performance is evaluated using accuracy\_score, classification\_report, and confusion matrix.
   * The accuracy is printed along with a detailed classification report that includes precision, recall, and F1 score for each class.
2. **Random Forest Classifier:**
   * A RandomForestClassifier is trained on the same training data and evaluated using the same metrics.
   * The confusion matrix is visualized using a heatmap from seaborn, showing the distribution of true and predicted labels for the test set.
3. **Class Distribution Visualization:**
   * A count plot is used to visualize the distribution of the target variable Exited, showing the number of customers who left vs. those who stayed.

**Handling Class Imbalance with SMOTE**

1. **SMOTE Resampling:**
   * The class imbalance is addressed using SMOTE from the imblearn library, which generates synthetic samples for the minority class.
   * After resampling, the Logistic Regression model is retrained on the balanced dataset and evaluated again.
2. **Evaluation Post-SMOTE:**
   * The accuracy and classification report after applying SMOTE are printed.
   * The class distribution post-resampling is visualized to confirm the balancing of the target classes.

**Visualizations**

* **Confusion Matrix Heatmap:** The confusion matrix is visualized for the Random Forest model, displaying the correct and incorrect predictions.
* **Class Distribution Before and After SMOTE:** A bar plot is used to show the class distribution before and after SMOTE resampling to ensure a more balanced dataset.

**Conclusion**

The workflow successfully processes the customer churn data, applies necessary transformations, trains different machine learning models, and addresses the class imbalance using SMOTE. The Random Forest and Logistic Regression models are evaluated for their performance, and class balance is improved through resampling techniques, enhancing the model's ability to predict minority class outcomes.

By handling missing or irrelevant features and addressing class imbalance, the models can make more accurate predictions on whether customers will churn. Further improvements can be made by exploring different algorithms, hyperparameter tuning, or additional feature engineering.