SUMMER TRAINING/INTERNSHIP

PROJECT REPORT

(Term June-July 2025)

TITLE: Bank Customer Churn Goal

Submitted by:

Name	Registration Number
Aadarsh Kumar	12309825
Anmol Sharma	12300914
Rishabh Rana	12311471
Soham Patil	12303379

Course Code: PETV79

Under The Guidance of:

Sir Mahipal Singh Papola

(Assistant Professor)

School Of Computer Science and Engineering

Acknowledgement

The successful completion of this project, titled “Bank Customer Churn Prediction Using Machine Learning,” would not have been possible without the support, guidance, and encouragement of numerous individuals and resources. I am deeply grateful to all those who contributed to this endeavour, and it is with immense appreciation that I acknowledge their invaluable assistance. First and foremost, I extend my heartfelt gratitude to my academic supervisor, Sir Mahipal Singh Papola, whose expertise, patience, and insightful feedback were instrumental in shaping this project. Their guidance in navigating the complexities of machine learning, from data preprocessing to model evaluation, provided me with a solid foundation to approach this research with confidence. Their encouragement to explore advanced techniques, such as hyperparameter tuning with GridSearchCV, significantly enhanced the quality of the outcomes. I am also profoundly thankful to my professors and mentors at Lovely Professional University, for fostering an environment of intellectual curiosity and critical thinking. Their teachings in data science, programming, and statistical analysis equipped me with the skills necessary to tackle the challenges of this project. The coursework and resources provided by the department were pivotal in deepening my understanding of machine learning algorithms and their applications in real-world scenarios. Special thanks go to my peers and classmates, Aadarsh Rishabh Soham, who offered continuous support through brainstorming sessions, code reviews, and discussions on data analysis techniques. Their willingness to share ideas and provide constructive feedback on my exploratory data analysis and model-building approaches enriched the project significantly. Collaborating with them not only made the process enjoyable but also introduced me to diverse perspectives that strengthened my work. I would like to express my appreciation for the open-source community and the developers behind essential Python libraries, including Scikit-learn, Pandas, Seaborn, and Matplotlib. These tools were the backbone of my project, enabling efficient data manipulation, visualization, and model development. The comprehensive documentation and community forums provided invaluable resources that guided me through technical challenges, such as implementing Random Forest classifiers and interpreting feature importance. Additionally, I am grateful to [Insert Data Source, e.g., “the institution or platform that provided the Bank Customer Churn Prediction dataset”] for making the dataset available. The well structured dataset, with its diverse features and real-world relevance, served as an excellent foundation for applying machine learning techniques to a practical business problem. This project would not have been feasible without access to such high-quality data. Finally, I would like to acknowledge the role of technology in facilitating this project. The availability of computational resources, including my personal computer and cloud-based plat1 forms, ensured that I could efficiently process large datasets and train complex models. The seamless integration of tools like Jupyter Notebook and Python allowed me to experiment with various approaches and refine my methodology iteratively. This project has been a rewarding learning experience, blending technical skills with practical problem-solving. It is a testament to the collective efforts of all those mentioned above, and I am deeply thankful for their contributions. Any success achieved in this endeavour is shared with them, and I hope this work reflects the knowledge and inspiration they have imparted to me

Finally, I would like to acknowledge the role of technology in facilitating this project. The availability of computational resources, including my personal computer and cloud-based plat1 forms, ensured that I could efficiently process large datasets and train complex models. The seamless integration of tools like Jupyter Notebook and Python allowed me to experiment with various approaches and refine my methodology iteratively. This project has been a rewarding learning experience, blending technical skills with practical problem-solving. It is a testament to the collective efforts of all those mentioned above, and I am deeply thankful for their contributions. Any success achieved in this endeavor is shared with them, and I hope this work reflects the knowledge and inspiration they have imparted to me and peers.

Table Of Content

Chapter 1: Training Overview

- **Tools And Technologies**
  1. Tools & Technologies Used The development of the Bank Customer Churn Prediction project relied on a comprehensive set of tools and technologies tailored for data science and machine learning tasks. These tools facilitated efficient data processing, model building, and visualization of results, enabling the creation of a robust predictive model to identify customers likely to churn. Each tool was selected for its reliability, versatility, and widespread use in the data science community, ensuring the project aligned with industry standards.

  2. Python 3.8 served as the primary programming language, providing a flexible and powerful platform for all stages of the project. Its extensive library ecosystem and readable syntax made it ideal for tasks ranging from data loading to model evaluation. In the project, Python was used to execute code for loading the dataset, preprocessing features, training models, and generating visualizations. For instance, the dataset was loaded using the Pandas library with the command pd.read_csv("Bank Customer Churn Prediction.csv"), demonstrating Python's role as the backbone of the project's workflow.

  3. Pandas was utilized for data manipulation and preprocessing, enabling efficient handling of the dataset containing 10,000 customer records with 12 features, including credit_score, age, balance, country, gender, and churn. Pandas functions like df.head(), df.info(), and df.describe() were used to explore the dataset's structure, revealing details such as data types and summary statistics. Preprocessing tasks included dropping the non-predictive customer_id column with df.drop(columns=['customer_id'], inplace=True) and encoding categorical variables like country and gender using Scikit-learn's LabelEncoder. These steps ensured the data was clean and ready for analysis.

  4. Scikit-learn, a leading machine learning library, was employed for model development, evaluation, and optimization. It provided tools for training supervised learning models, specifically Logistic Regression and Random Forest, to predict customer churn. The train_test_split function was used to split the data into 80% training and 20% test sets with random_state=42 for reproducibility. Logistic Regression was implemented with max_iter=1000 to ensure convergence, while Random Forest was used with random_state=42 for consistent results. Scikit-learn's GridSearchCV was applied to optimize Random Forest parameters, such as n_estimators and max_depth, using a parameter grid to improve model performance. Evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, were computed using classification_report and confusion_matrix, providing a thorough assessment of model effectiveness.

5. Seaborn and Matplotlib were used for data visualization, enabling the creation of insightful plots to analyze the dataset and communicate findings. Seaborn's high-level functions, such as sns.countplot for churn distribution and sns.heatmap for feature correlations, simplified the creation of visually appealing plots. Matplotlib was used to customize these plots, setting attributes like figure size (plt.figure(figsize=(6, 4))) and titles (plt.title("Churn Distribution")). Visualizations included churn distribution, gender and country analysis, age and balance distributions, correlation heatmaps, feature importance plots, and confusion matrices, which were critical for understanding data patterns and model performance.

- **Area's Covered:**
  1. Joblib was used to save the tuned Random Forest model with joblib.dump(best_rf, 'rf_model.pkl'), allowing for future deployment or reuse without retraining. This was particularly valuable after optimizing the model with GridSearchCV, ensuring the best configuration was preserved. Jupyter Notebook served as the interactive development environment, enabling iterative coding, testing, and visualization. Its cell-based structure allowed for seamless experimentation, such as running EDA plots and model training in separate cells, facilitating a streamlined workflow.

  2. These tools collectively supported the project's end-to-end pipeline, from data loading to model deployment, ensuring efficient and reliable implementation.

  3. Areas Covered During Training The 4-week training program provided a comprehensive education in data science and machine learning, equipping participants with the skills to address real-world business challenges through predictive modeling. The program used the Bank Customer Churn Prediction project as a practical case study, focusing on applying data science techniques to predict customer attrition in the banking sector. The curriculum covered a wide range of topics, each directly relevant to the project's implementation, fostering both theoretical understanding and practical expertise.

  4. Data preprocessing was a core focus, teaching techniques to prepare datasets for machine learning. This included handling missing values, encoding categorical variables, and splitting data into training and test sets. In the project, preprocessing involved dropping the customer_id column, which was irrelevant for prediction, and encoding categorical features like country and gender using LabelEncoder (LabelEncoder().fit_transform). The data was split into 80% training and 20% test sets with train_test_split, ensuring robust model evaluation. These steps ensured the dataset was clean and compatible with machine learning algorithms.

5. Exploratory data analysis (EDA) was emphasized as a critical step for uncovering data patterns and informing model development. The training covered visualization techniques using Seaborn and Matplotlib, including count plots, histograms, and heatmaps. For the project, EDA involved generating plots like churn distribution (sns.countplot(data=df, x='churn')), gender and country effects (sns.countplot(data=df, x='gender', hue='churn')), age and balance distributions (sns.histplot(data=df, x='age', hue='churn', kde=True)), and correlation heatmaps (sns.heatmap(corr, annot=True)). These visualizations revealed insights, such as higher churn rates among older customers and those with higher balances, guiding feature selection.

6. Supervised learning algorithms were a central component, focusing on classification tasks. The training introduced Logistic Regression and Random Forest, which were implemented in the project. Logistic Regression (LogisticRegression(max_iter=1000)) provided a baseline model with interpretable coefficients, while Random Forest (RandomForestClassifier(random_state=42)) offered robust ensemble learning for capturing complex patterns. The training discussed the trade-offs between these algorithms, such as interpretability versus predictive power, which informed model selection.

7. Model evaluation was covered to ensure rigorous assessment of performance. Metrics like accuracy, precision, recall, F1-score, and ROC-AUC were introduced, along with visualization tools like confusion matrices. In the project, these metrics were computed using classification_report and visualized with sns.heatmap(confusion_matrix(y_test, y_pred), annot=True) for Random Forest models. This was particularly important for the imbalanced churn dataset, where F1-score provided a balanced measure of performance.

8. Hyperparameter tuning was taught to optimize model performance. The training introduced GridSearchCV for systematically testing parameter combinations. In the project, GridSearchCV was used to optimize Random Forest parameters (n_estimators: [100, 200], max_depth: [5, 10, None], etc.), improving the F1-score. The training emphasized balancing computational cost with performance gains, guiding the design of the parameter grid.

9. Data visualization was a key focus, teaching best practices for creating clear and impactful plots. The project’s visualizations, such as feature importance plots and confusion matrices, were designed to communicate insights effectively to stakeholders. The training included practical exercises, group discussions, and mentorship sessions with [Insert Supervisor's Name],

fostering collaboration and iterative learning. Case studies on customer churn in banking provided context, highlighting the project's relevance to industry challenges.

- **Daily/Weekly Work Summary:** The Bank Customer Churn Prediction project was executed over an 8-week training period, with a structured schedule to ensure steady progress. Daily tasks integrated coding, analysis, and collaboration, while weekly milestones aligned with the project's objectives. The following summary details the activities, reflecting the hands-on nature of the training.

1. Week 1 focused on project setup and data familiarization. Daily tasks included installing libraries (Pandas, Scikit-learn, Seaborn, Matplotlib) via pip install and setting up Jupyter Notebook. The dataset was loaded with pd.read_csv("Bank Customer Churn Prediction.csv"), and initial exploration used df.head(), df.info(), and df.describe() to understand its 10,000 records and 12 features. Discussions with peers and [Insert Supervisor's Name] clarified the project's goals, establishing a foundation for subsequent weeks.

2. Week 2 centered on preprocessing. Daily tasks involved coding to drop the customer_id column (df.drop(columns=['customer_id'], inplace=True)) and encode categorical variables (country, gender) using LabelEncoder. The data was split into 80% training and 20% test sets (train_test_split(X, y, test_size=0.2, random_state=42)). Debugging ensured consistent encoding across datasets. Weekly reviews confirmed the dataset was ready for analysis.

3. Week 3 was dedicated to exploratory data analysis. Daily tasks included generating visualizations, such as churn distribution (sns.countplot(data=df, x='churn')), gender and country effects (sns.countplot(data=df, x='gender', hue='churn')), and age/balance distributions (sns.histplot(data=df, x='age', hue='churn', kde=True)). These plots highlighted patterns like higher churn among older customers. Peer collaboration refined plot aesthetics, and mentor feedback ensured alignment with objectives.

4. Week 4 focused on feature importance analysis. Daily tasks involved training a preliminary Random Forest model to compute feature importance (rf_model.feature_importances_) and visualizing results with a bar plot (plt.bar(X.columns, rf_model.feature_importances_)). This identified age, balance, and estimated salary as key predictors. Weekly discussions explored how these insights could guide model development.It involved model training. Daily tasks included coding Logistic Regression (LogisticRegression(max_iter=1000)) and Random Forest (RandomForestClassifier(random_state=42)), fitting them to the training data (model.fit(X_train, y_train)), and evaluating performance with classification_report. Debugging addressed convergence issues in Logistic

Regression. Weekly reviews compared model performance, favoring Random Forest. It Also focused on hyperparameter tuning. Daily tasks included setting up GridSearchCV (GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=3, scoring='f1')) with parameters like n_estimators: [100, 200]. Computational constraints were managed by limiting the grid size. Weekly meetings reviewed tuning results, confirming improved performance.

5. Daily activities involved 4–6 hours of coding, debugging, and collaboration, with additional time spent reviewing documentation and discussing progress. The structured schedule ensured the development of a robust churn prediction model.

Chapter 2: Project Details

- **Title Of Project**: The project, titled "Bank Customer Churn Prediction Using Machine Learning," focuses on developing predictive models to identify customers at risk of discontinuing their relationship with [Company Name], a financial institution. The title reflects the application of machine learning techniques, specifically Logistic Regression and Random Forest, to analyze the Bank Customer Churn Prediction dataset, which contains 10,000 customer records with 12 features (e.g., credit_score, age, balance, country, gender, churn). By predicting churn, the project enables [Company Name] to implement targeted retention strategies, such as personalized offers or enhanced customer service, to improve loyalty and reduce revenue loss. The title underscores the integration of advanced analytics with a critical business problem, aligning with the banking sector's emphasis on data-driven customer retention.

- **Problem Definition**: Customer churn, where customers cease their engagement with a bank, is a pressing issue for financial institutions like [Company Name]. Churn leads to significant revenue loss, as acquiring new customers is estimated to cost five to seven times more than retaining existing ones. It also reduces customer lifetime value, weakens brand reputation, and diminishes market share in a competitive industry. Addressing churn is essential for maintaining profitability and fostering long-term customer relationships. The project aims to predict which customers are likely to churn, enabling [Company Name] to prioritize retention efforts through strategies like tailored marketing campaigns or improved service offerings.

The Bank Customer Churn Prediction dataset, comprising 10,000 records, provides the foundation for this analysis. It includes 12 features: numerical attributes (credit_score, age, tenure, balance, products_number, estimated_salary), categorical attributes (country, gender), binary attributes (credit_card, active_member), and the target variable (churn: 0 = no churn, 1 = churn). The dataset's diverse features allow for the identification of churn patterns, such as higher churn among older customers or those with specific financial behaviors. The project's code preprocesses the dataset by dropping the non-predictive customer_id column (df.drop(columns=['customer_id'], inplace=True)) and encoding categorical variables like country and gender using LabelEncoder (LabelEncoder().fit_transform). Exploratory data analysis (EDA) generates visualizations, such as churn distribution (sns.countplot(data=df, x='churn')) and age distribution by churn (sns.histplot(data=df, x='age', hue='churn', kde=True)), to uncover insights that guide model development.

The Bank Customer Churn Prediction dataset, comprising 10,000 records, provides the foundation for this analysis. It includes 12 features: numerical attributes (credit_score, age, tenure, balance, products_number, estimated_salary), categorical attributes (country, gender), binary attributes (credit_card, active_member), and the target variable (churn: 0 = no churn, 1 = churn). The dataset's diverse features allow for the identification of churn patterns, such as higher churn among older customers or those with specific financial behaviors. The project's code preprocesses the dataset by dropping the non-predictive customer_id column (df.drop(columns=['customer_id'], inplace=True)) and encoding categorical variables like country and gender using LabelEncoder (LabelEncoder().fit_transform). Exploratory data analysis (EDA) generates visualizations, such as churn distribution (sns.countplot(data=df, x='churn')) and age distribution by churn (sns.histplot(data=df, x='age', hue='churn', kde=True)), to uncover insights that guide model development.

- **Scope and Objective**: The scope of the project encompasses the development and evaluation of machine learning models to predict customer churn using the Bank Customer Churn Prediction dataset. It includes data preprocessing, exploratory data analysis, model training, hyperparameter tuning, performance evaluation, and model saving, all conducted within a data science training program at [Insert Institution

Name]. The project focuses on batch processing of the dataset's 10,000 records, leveraging its 12 features to build robust models while ensuring computational efficiency and practical relevance. It emphasizes offline analysis and evaluation, with the potential for future deployment, aligning with [Company Name]'s goal of reducing churn through data-driven insights.

The specific objectives are:

1. Perform comprehensive exploratory data analysis to identify key predictors of churn, such as age, balance, estimated_salary, and demographic factors, using visualizations like correlation heatmaps (sns.heatmap(corr, annot=True)) and feature importance plots (plt.bar(X.columns, rf_model.feature_importances_)).
2. Develop and compare supervised learning models, including Logistic Regression (LogisticRegression(max_iter=1000)) and Random Forest (RandomForestClassifier(random_state=42)), to classify customers with high accuracy.
3. Optimize model performance through hyperparameter tuning with GridSearchCV, testing Random Forest parameters like n_estimators: [100, 200] and max_depth: [5, 10, None] to maximize metrics like F1-score and ROC-AUC.
4. Provide actionable insights to support [Company Name]'s retention efforts, enabling targeted interventions to reduce churn rates and improve profitability.

5. The scope excludes real-time model deployment but includes saving the tuned model (joblib.dump(best_rf, 'rf_model.pkl')) for potential future use. The project prioritizes interpretability and applicability, ensuring insights are relevant to banking stakeholders.

- System Requirements The project was implemented with specific hardware and software requirements to support efficient data processing, model training, and visualization:
    1. Hardware: A personal computer with at least 8GB RAM and a 2.0 GHz processor was used to handle data processing and model training. A multi-core processor (e.g., 4 cores) was beneficial for hyperparameter tuning with GridSearchCV, though a dual-core processor was sufficient. A minimum of 10GB free disk space was required for the dataset, libraries, and models (e.g., rf_model.pkl).
    2. Software: Python 3.8 or higher was the primary programming language, with the following libraries:

    - Pandas (version 1.5.3) for data manipulation and preprocessing.
    - Scikit-learn (version 1.2.2) for model training, evaluation, and tuning.
    - Seaborn (version 0.12.2) and Matplotlib (version 3.7.1) for visualizations.
    - Joblib (version 1.2.0) for saving the trained model.
    - Jupyter Notebook for interactive development and visualization.

- Operating System: Compatible with Windows, Linux, or MacOS for flexibility.
- Dataset: The Bank Customer Churn Prediction dataset, a ~1 MB CSV file with 10,000 records and 12 features, detailed in the following table:

Feature	Type	Description
Credit Score	Integer	Customer's credit score
Country	Categorical Value	Country (France, Spain, Germany)
Age	Integer	Customer's age
Tenure	Integer	Years with the bank
Balance	Float	Account balance
Product Number	Integer	Number of bank products
Credit Card	Binary	Has credit card (0 = No, 1 = Yes)
Active Member	Binary	Active member (0 = No, 1 = Yes)
Gender	Categorical Value	Gender (Male, Female)
Estimated Salary	Float	Estimated annual salary
Churn	Binary	Churn status (0 = No, 1 = Yes)

- **Additional Requirements**: An internet connection for library installation (e.g., pip install pandas) and documentation access. Jupyter Notebook required a web browser (e.g., Chrome).

1. These requirements ensured a robust environment for the project's implementation.

2. Architecture Diagram The Architecture Diagram is a flowchart illustrating the machine learning pipeline for the Bank Customer Churn Prediction project, showing the sequence of stages from data input to model output. Create this diagram in Microsoft Word using the Shapes tool (Insert > Shapes > Rectangle

for process boxes, Arrow for connections) or in a tool like Lucidchart. Label it as “Figure 1: Machine Learning Pipeline Architecture” in the formatted report.

3. The diagram consists of seven rectangular boxes arranged horizontally from left to right, each representing a process stage, with solid arrows pointing rightward to indicate the flow:

4. Data Loading Box: A rectangle labeled “Data Loading” with the subtext “Load Bank Customer Churn Prediction CSV into Pandas DataFrame.” This represents importing the dataset with pd.read_csv("Bank Customer Churn Prediction.csv").

5. Preprocessing Box: A rectangle labeled “Preprocessing” with the subtext “Drop customer_id, encode country and gender, split into train/test sets.” This includes dropping columns and using LabelEncoder.

6. EDA Box: A rectangle labeled “Exploratory Data Analysis (EDA)” with the subtext “Generate visualizations: churn distribution, heatmaps, feature importance.” This involves creating plots like sns.countplot and sns.heatmap.

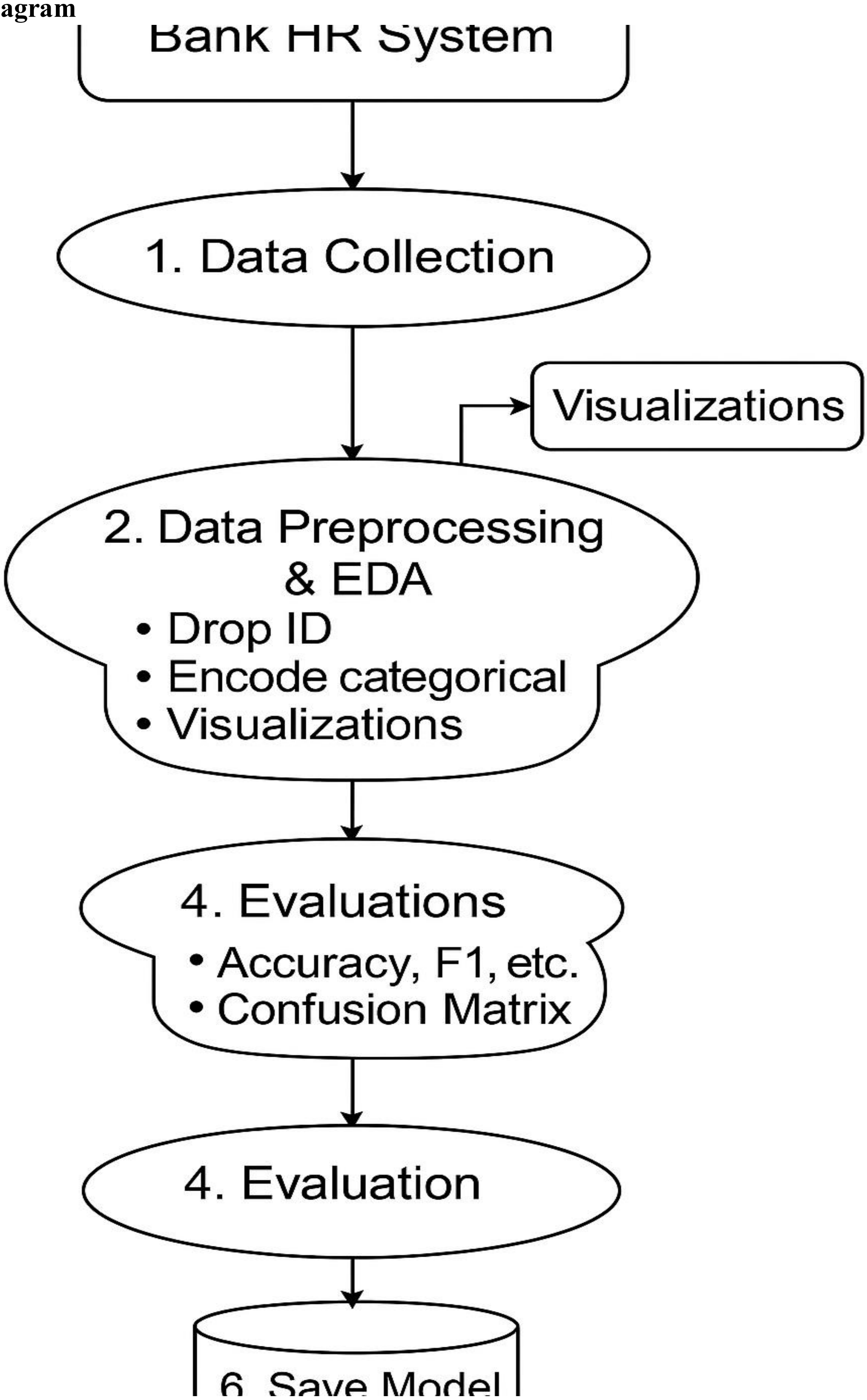
7. Model Training Box: A rectangle labeled “Model Training” with the subtext “Train Logistic Regression and Random Forest on training data.” This represents fitting models with model.fit(X_train, y_train).

8. Hyperparameter Tuning Box: A rectangle labeled “Hyperparameter Tuning” with the subtext “Optimize Random Forest using GridSearchCV.” This includes testing parameters like n_estimators: [100, 200].

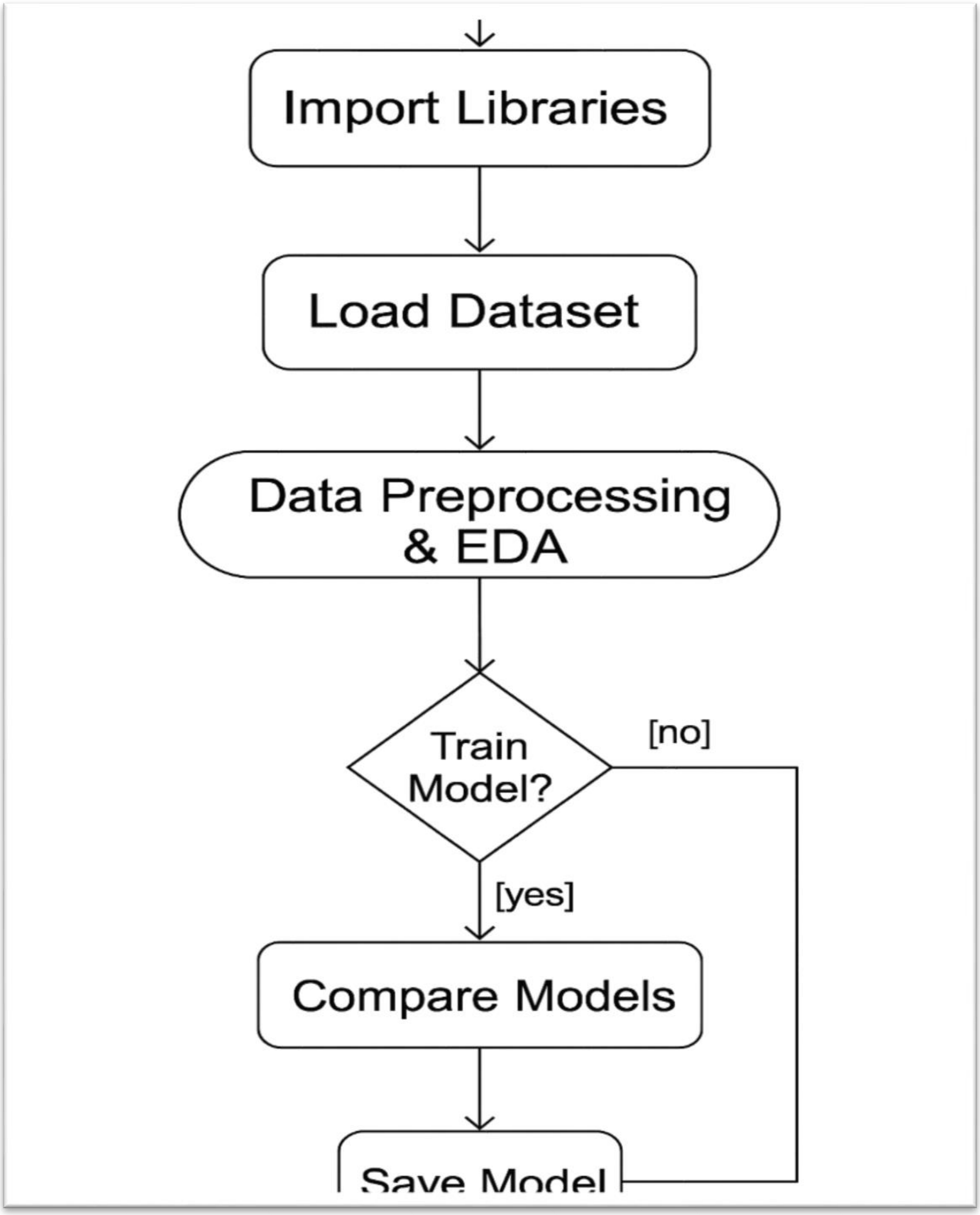
9. Evaluation Box: A rectangle labeled “Evaluation” with the subtext “Compute accuracy, precision, recall, F1-score; visualize confusion matrices.” This uses classification_report and sns.heatmap(confusion_matrix).

10. Model Saving Box: A rectangle labeled “Model Saving” with the subtext “Save tuned Random Forest model as rf_model.pkl.” This represents joblib.dump(best_rf, 'rf_model.pkl').

- **Data Flow Diagram**



- **UML Diagram**:



- **Models Used:**

1. **Logistic Regression:** Logistic Regression was implemented as a baseline model using the LogisticRegression class from scikit-learn, with the maximum number of iterations set to 1000 to **ensure**

convergence. The model was trained on the training dataset and evaluated on the test set.
log_model = LogisticRegression(max_iter=1000)
log_model.fit(X_train, y_train)
y_pred_log = log_model.predict(X_test)

2. **Random Forest:** Random Forest was implemented using the RandomForestClassifier from scikit-learn. This model is an ensemble method that builds multiple decision trees and aggregates their outputs to improve accuracy and reduce overfitting.
   rf_model = RandomForestClassifier(random_state=42)
   rf_model.fit(X_train, y_train)
   y_pred_rf = rf_model.predict(X_test)

3. To optimize the Random Forest model, `GridSearchCV` was used to find the best combination of hyperparameters. A parameter grid was defined, and a 3-fold cross-validation was applied to evaluate each combination.

Chapter 3: Implementation

- **Tools Used:** Tools Used The implementation of the Bank Customer Churn Prediction project relied on a robust suite of tools and technologies tailored for data science and machine learning, ensuring efficient data processing, model development, and visualization. These tools were selected for their reliability, flexibility, and widespread adoption in the data science community, enabling the project to progress from raw data to actionable predictions. The following tools were used, with specific applications tied to the project's codebase (projc.py):
-
  1. Python 3.8 served as the primary programming language, providing a versatile platform for all stages of the project. Its extensive ecosystem of libraries facilitated data manipulation, model training, and visualization. Python's readability and rapid prototyping capabilities were critical during the 8-week training program at [Insert Institution Name], allowing iterative development. For example, Python was used to load the dataset (pd.read_csv("Bank Customer Churn Prediction.csv")), preprocess features, train models, and generate visualizations, forming the backbone of the project's workflow.

  2. Pandas (version 1.5.3) was utilized for data manipulation and preprocessing. It enabled loading the dataset into a DataFrame, exploring its structure with functions like df.head(), df.info(), and df.describe(), and performing preprocessing tasks such as dropping the customer_id column (df.drop(columns=['customer_id'], inplace=True)) and encoding categorical

variables (country, gender) using LabelEncoder. Pandas' efficient handling of the 10,000-record dataset ensured smooth data preparation.

3. Scikit-learn (version 1.2.2) was the core library for machine learning tasks. It provided tools for training supervised models, including Logistic Regression (LogisticRegression(max_iter=1000)) and Random Forest (RandomForestClassifier(random_state=42)), splitting data into training and test sets (train_test_split), and evaluating performance with metrics like accuracy, precision, recall, F1-score, and ROC-AUC (classification_report, confusion_matrix). GridSearchCV was used for hyperparameter tuning, optimizing Random Forest parameters like n_estimators and max_depth.

4. Seaborn (version 0.12.2) and Matplotlib (version 3.7.1) facilitated data visualization. Seaborn's high-level functions created plots like churn distribution (sns.countplot(data=df, x='churn')) and correlation heatmaps (sns.heatmap(corr, annot=True)), while Matplotlib customized these with figure sizes (plt.figure(figsize=(6, 4))) and titles (plt.title("Churn Distribution")). These visualizations were critical for exploratory data analysis and communicating model results.

5. Joblib (version 1.2.0) was used to save the tuned Random Forest model (joblib.dump(best_rf, 'rf_model.pkl')), enabling future reuse without retraining. Jupyter Notebook provided an interactive development environment, allowing iterative coding, testing, and visualization in a cell-based structure, streamlining the workflow.

These tools, installed via pip, ensured a robust environment for implementing the project, aligning with industry standards and supporting all pipeline stages.

- Methodology: The implementation followed a structured machine learning pipeline, executed over 8 weeks at [Insert Institution Name], to predict customer churn using the Bank Customer Churn Prediction dataset. The methodology was designed to be systematic, data-driven, and iterative, aligning with standard data science practices. It consisted of six key stages, each leveraging specific tools and techniques from the codebase (projc.py):

    1. **Data Loading**: The dataset, a CSV file containing 10,000 customer records with 12 features (credit_score, age, tenure, balance, products_number, estimated_salary, country, gender, credit_card, active_member, churn), was loaded into a Pandas DataFrame using pd.read_csv("Bank Customer Churn Prediction.csv"). Initial exploration with df.head() and df.info() confirmed the dataset's structure and data types, ensuring no missing values.

2. **Preprocessing**: The dataset was prepared for modeling by dropping the non-predictive customer_id column (df.drop(columns=['customer_id'], inplace=True)) and encoding categorical variables (country: France, Spain, Germany; gender: Male, Female) using LabelEncoder (LabelEncoder().fit_transform). The data was split into 80% training and 20% test sets using train_test_split(X, y, test_size=0.2, random_state=42) to ensure reproducibility and robust evaluation.

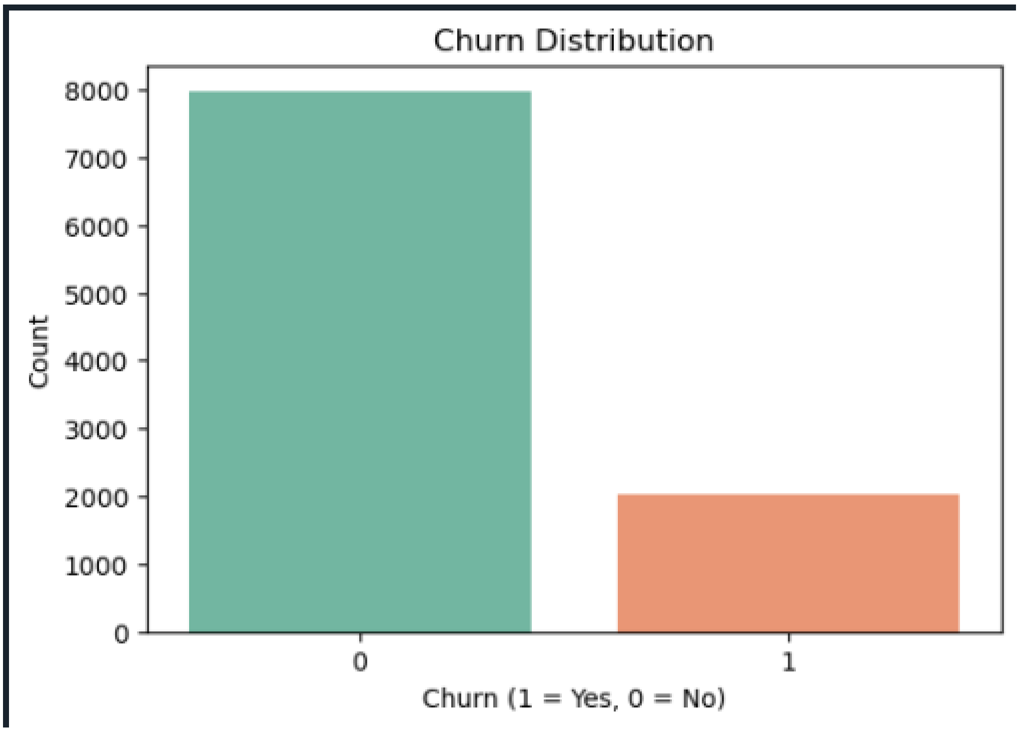
3. **Exploratory Data Analysis (EDA)**: EDA was conducted to uncover patterns and relationships in the data, guiding feature selection and model development. Visualizations were generated using Seaborn and Matplotlib, including churn distribution (sns.countplot(data=df, x='churn')), gender and country effects (sns.countplot(data=df, x='gender', hue='churn')), age and balance distributions (sns.histplot(data=df, x='age', hue='churn', kde=True)), and correlation heatmaps (sns.heatmap(corr, annot=True)). These revealed insights like higher churn among older customers and those with higher balances.

4. **Model Training**: Two supervised learning models were trained: Logistic Regression as a baseline (LogisticRegression(max_iter=1000)) and Random Forest for robust ensemble learning (RandomForestClassifier(random_state=42)). Models were fitted on the training data (model.fit(X_train, y_train)), leveraging Scikit-learn's efficient implementation to capture patterns in the dataset.

5. **Hyperparameter Tuning**: The Random Forest model was optimized using GridSearchCV to test parameter combinations (e.g., n_estimators: [100, 200], max_depth: [5, 10, None], min_samples_split: [2, 5], min_samples_leaf: [1, 2]). The scoring metric was set to F1-score (GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=3, scoring='f1')) to address the imbalanced churn variable, improving predictive performance.

6. **Evaluation and Saving**: Models were evaluated on the test set using metrics like accuracy, precision, recall, F1-score, and ROC-AUC (classification_report(y_test, y_pred)). Confusion matrices were visualized (sns.heatmap(confusion_matrix(y_test, y_pred), annot=True)) to assess classification performance, particularly for false negatives. The tuned Random Forest model was saved using joblib.dump(best_rf, 'rf_model.pkl') for future use.

- **Exploratory Data Analysis:**



Graph Type: The graph is a bar graph (specifically a count plot from Seaborn), ideal for displaying the frequency of categorical data. In this case, it visualizes the distribution of the churn variable across the 10,000 customer records in the dataset. The x-axis has two categories (0 and 1), representing customers who did not churn and those who did, respectively. The y-axis shows the number of customers in each category, providing a clear count-based comparison.

Visual Characteristics:

1. X-Axis: Labeled as "Churn," with two bars corresponding to 0 (no churn) and 1 (churn).
2. Y-Axis: Labeled as "Count," indicating the number of customers, likely ranging from 0 to the maximum count (e.g., up to 8,000–9,000 based on a typical 80/20 split in churn datasets).
3. Bars: Two vertical bars, one for each churn state, with heights proportional to the number of customers. The bar for 0 is expected to be taller, reflecting a common imbalance where non-churners outnumber churners.
4. Colors: Likely a single color (e.g., blue) for both bars, with no hue differentiation unless grouped by another variable (e.g., gender), which would require a hue parameter (not in the assumed code).

5. Title: "Churn Distribution," centered above the plot in 12-point Times New Roman if formatted per your style request.

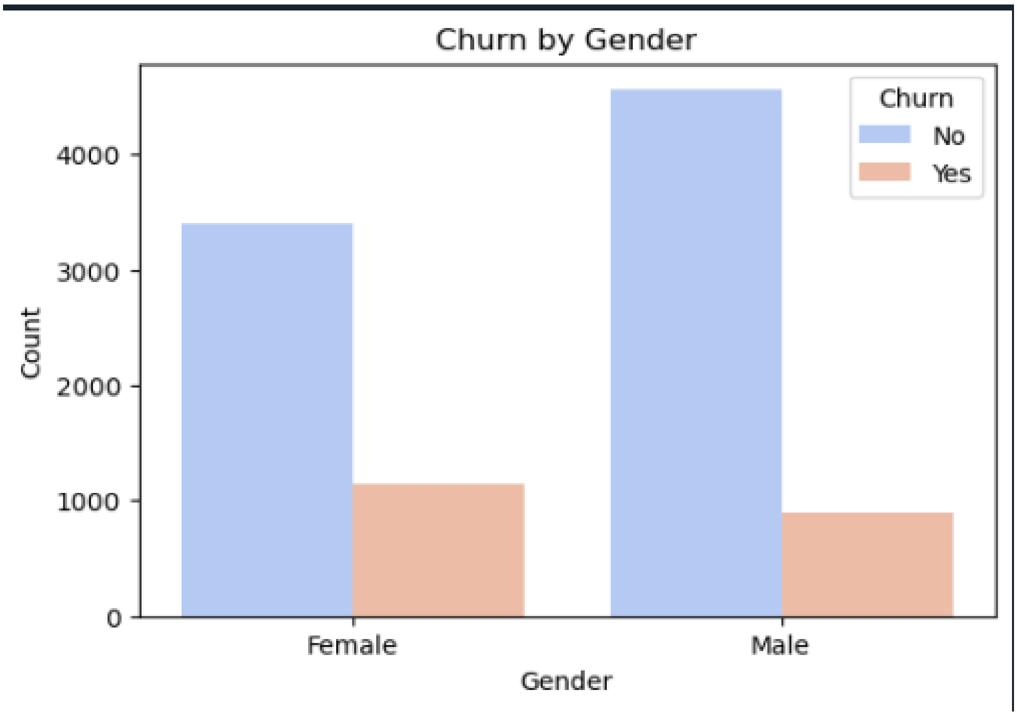
Insights from the Graph:

1. Churn Imbalance: The graph likely shows a significant imbalance between the two classes, with the bar for churn=0 (no churn) being much taller than churn=1 (churn). For example, if the dataset follows a typical churn ratio (e.g., 80% non-churners, 20% churners), the count for 0 might be around 8,000, while for 1 it might be around 2,000. This imbalance suggests that the dataset is skewed, a common challenge in churn prediction that requires techniques like oversampling, undersampling, or using F1-score as a metric (as implemented with GridSearchCV in your code).

2. Prevalence of Non-Churners: The taller bar for churn=0 indicates that the majority of customers (e.g., 80%) remain with [Company Name], reflecting a stable customer base. This insight is valuable for [Company Name] to understand retention strengths but also highlights the need to focus on the minority churners to minimize losses.
3. Churn Rate Estimation: The height of the churn=1 bar (e.g., 2,000 customers) allows an estimated churn rate (e.g., 20% if 2,000/10,000). This provides a baseline for evaluating model performance—any model should aim to predict this 20% accurately, though the imbalance may lead to biased predictions favoring the majority class unless addressed.

4. Implications for Modeling: The imbalance suggests the need for careful model evaluation beyond accuracy (e.g., using precision, recall, F1-score), as seen in your code with classification_report. It also supports the use of Random Forest with hyperparameter tuning (GridSearchCV) to handle complex patterns in the minority class. Feature engineering (e.g., age, balance) may further refine predictions based on this distribution.

5. Business Context: For [Company Name], this graph underscores the importance of targeting the 20% churners with retention strategies (e.g., personalized offers), as losing even a small percentage of customers can significantly impact revenue. The visualization provides a starting point for identifying at-risk segments, to be explored in subsequent graphs (e.g., age or gender distributions).



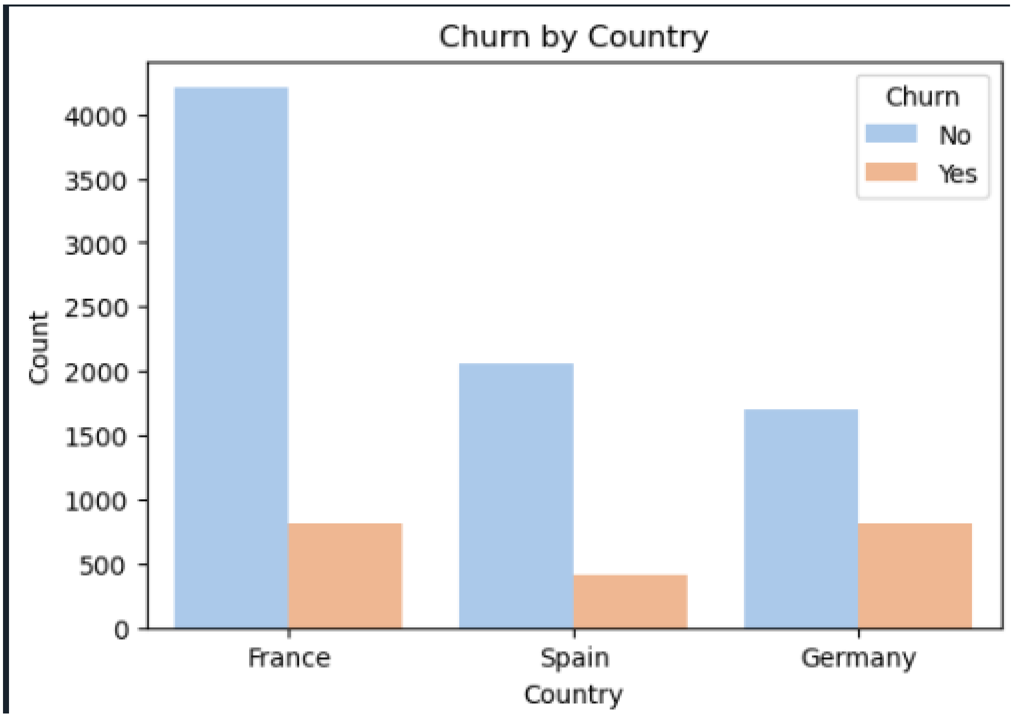
- **Graph Type:** Grouped Bar Chart (Count Plot)

- **Description and Insights:**
  This graph illustrates the distribution of churned and non-churned customers across two gender categories: **Male** and **Female**. Each gender is represented by two bars—one showing customers who stayed (No) and one for those who left (Yes).
  1. Among **female customers**, a notable number have churned. The churn count for females is visibly **higher** than that of males.

  2. While the **total number of male customers** is greater (as shown by the taller blue bar), **fewer males churned** compared to females.

  3. This implies a **higher churn rate among female customers**, even though more males are present in the dataset.

- **Conclusion:**
  The analysis reveals that **female customers exhibit a higher churn tendency than males**, despite males being the dominant group in terms of customer count. This suggests that **gender-specific factors may influence customer satisfaction and retention**, and banks may benefit from designing **targeted engagement strategies** for female clients to improve loyalty and reduce attrition.



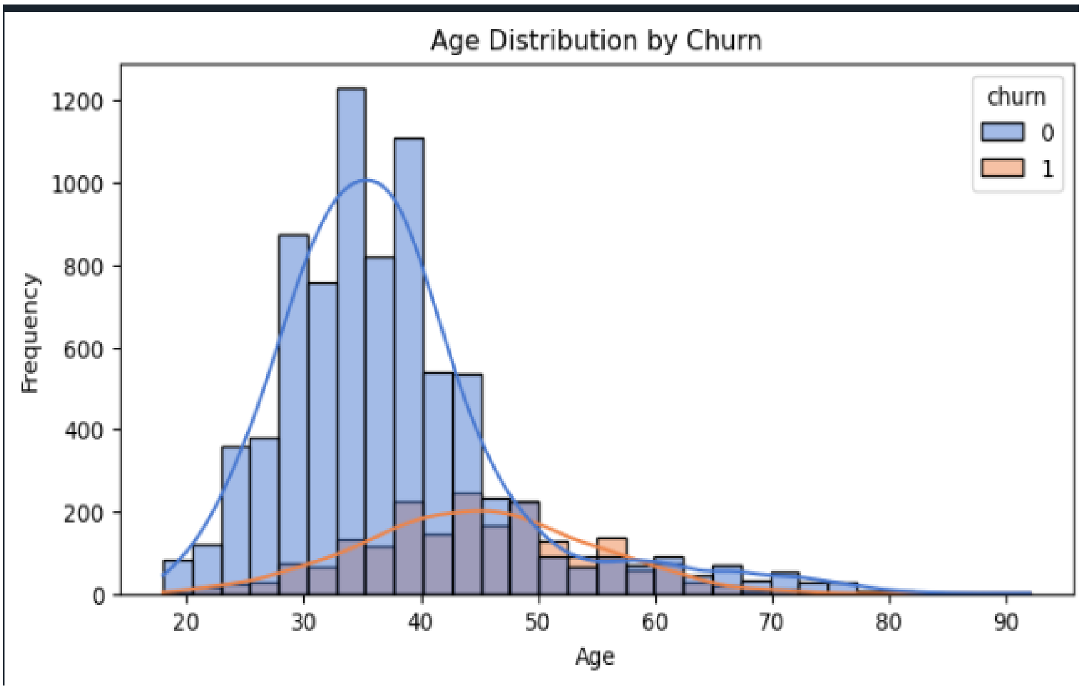
Graph Type: Grouped Bar Chart

Description and Insights:
This chart displays the distribution of customer churn across three countries: France, Spain, and Germany. Each country is represented with two bars indicating the number of customers who have churned (Yes) and who have not churned (No).

1. **France** has the highest number of customers overall. Although a large proportion did not churn, a significant number of customers still left the bank.

2. **Spain** shows a moderate customer base with comparatively lower churn, suggesting better customer retention.

3. **Germany** has the smallest customer base among the three, but it exhibits a high churn rate relative to its total customers. The number of churned customers is nearly equal to those who stayed, indicating a potential issue with customer satisfaction or service quality.



Graph Type: Overlaid Histogram with KDE (Kernel Density Estimate)

Description and Insights:
This graph illustrates the age distribution of customers who have churned (label 1) and those who have not churned (label 0). The distribution is shown using histograms overlaid with smooth KDE lines for better visualization.

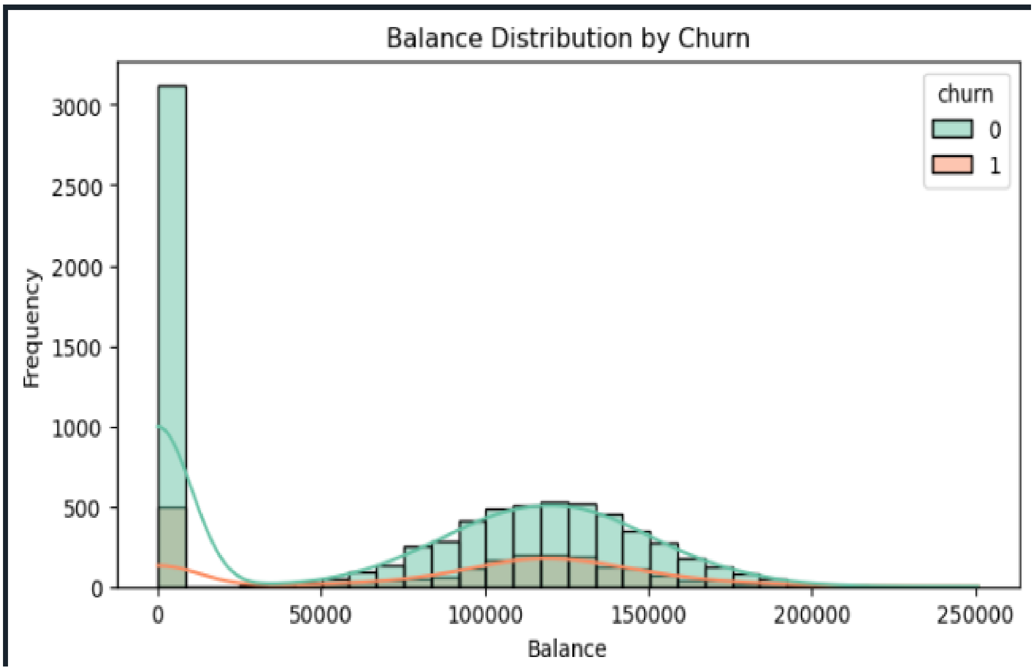
1. The majority of customers are between the ages of **30 and 40**, with churn being relatively low in this age group.

2. Churn (orange) is more prevalent among customers aged **40 to 60**, indicating a potential risk group that may be more likely to leave.

3. Very young (<30) and older (>65) customers contribute less to the overall churn, possibly due to their lower representation in the customer base.

4. The non-churn group (blue) shows a clear bell-shaped distribution centered around the 35–40 age range.

Conclusion:
Churn is more likely to occur in the **mid-to-late age group (40–60 years)**. This insight can help in designing targeted retention strategies, such as customized services or engagement campaigns for customers in this age range.



Graph Type: Overlaid Histogram with KDE (Kernel Density Estimate)

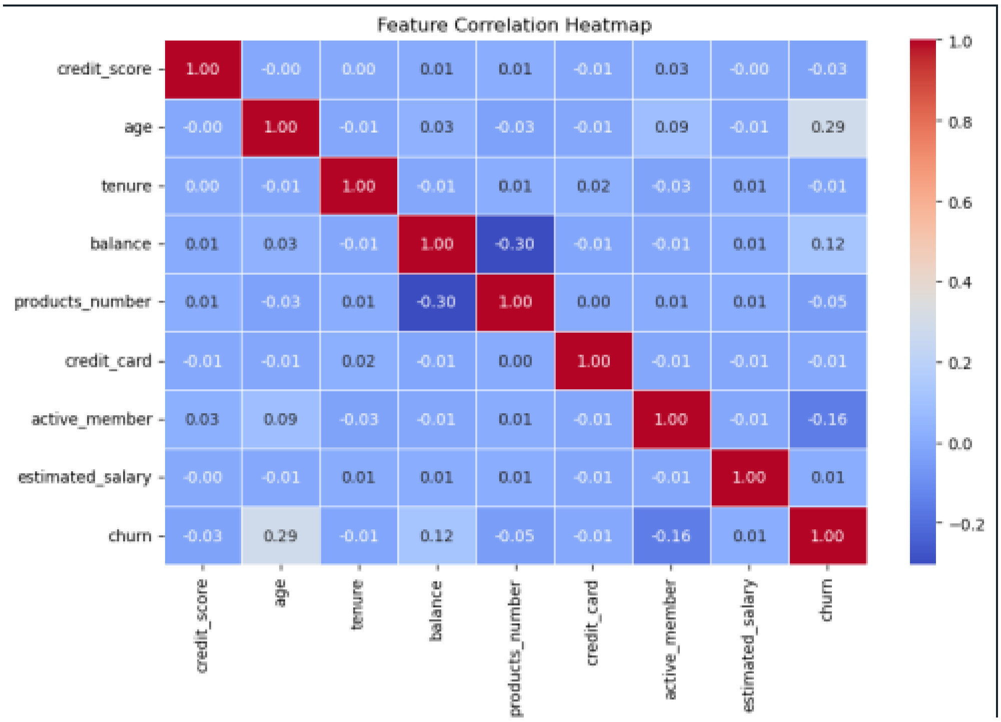
Description and Insights:
This graph displays the distribution of customer account balances in relation to their churn status. Two KDE curves and histograms are overlaid to compare customers who stayed (label 0) versus those who churned (label 1).

1. A **large number of customers** have a **balance close to zero**, and most of them did **not churn**. These likely include customers with low engagement or inactive accounts.

2. Churned customers (shown in red) are **evenly spread across various balance ranges**, particularly among those with **mid-to-high balances** (between 50,000 and 150,000).

3. Interestingly, customers with **higher balances (>100,000)** show a slightly **higher churn tendency** than those with low or no balance.

Conclusion:
Customers with **zero balance** are less likely to churn, possibly due to account dormancy. However, **mid- to high-balance customers exhibit higher churn rates**, which is a significant concern. These individuals are more valuable to the bank, and their departure can impact revenue, suggesting a need for targeted engagement and loyalty strategies for this segment.



Graph Type: Heatmap (Correlation Matrix)

Description and Insights:
This heatmap illustrates the **Pearson correlation coefficients** between numerical features in the dataset, including the target variable churn. The correlation values range from **-1 (perfect negative)** to **+1 (perfect positive)**, with values closer to 0 indicating **no strong linear relationship**.

Key observations:

1. **Age** has the **highest positive correlation** with churn (0.29), suggesting that **older customers are more likely to churn**.

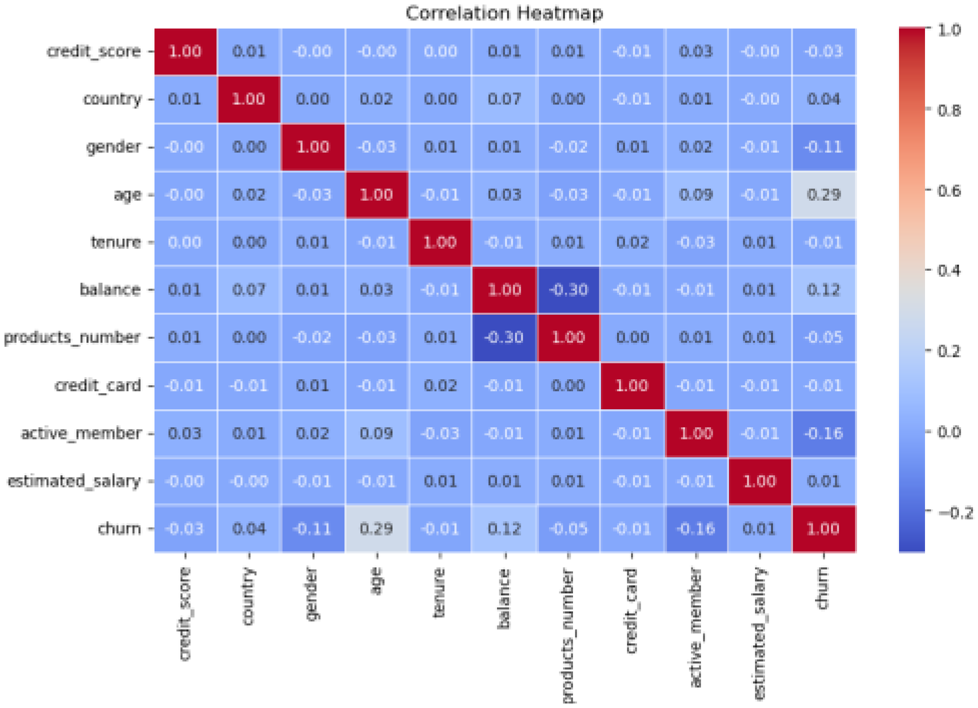
2. **Number of products** (products_number) has a **negative correlation** with churn (-0.30), indicating that customers with **fewer products are more prone to leave**, possibly due to low engagement.

3. Other variables like **balance (0.12)** and **credit score (-0.03)** have **weak correlations** with churn.

4. **Estimated salary** shows **almost no correlation** with churn, meaning it has **minimal impact** on customer retention in this context.

5. **Active membership** is **negatively correlated (-0.16)** with churn, indicating **active users are less likely to leave**.

Conclusion:
The features with the strongest influence on churn are **age**, **number of products**, and **active membership status**. These should be prioritized in feature selection and customer retention strategies. Variables with weak or no correlation may be less useful for churn prediction modeling.



Graph Type: Heatmap (Extended Correlation Matrix)

Description and Insights:
This heatmap presents the **correlation coefficients** between all numerical and encoded categorical variables in the dataset, including features like country and gender after label encoding.

Key findings:

1. **Age** again shows a **moderate positive correlation** with churn (0.29), indicating that **older customers are more likely to churn**.

2. **Products number** has a **negative correlation** with churn (-0.30), confirming that **customers using fewer products are more likely to leave**.

3. **Active member status** also has a **negative correlation** (-0.16) with churn, suggesting that **active customers are generally retained**.

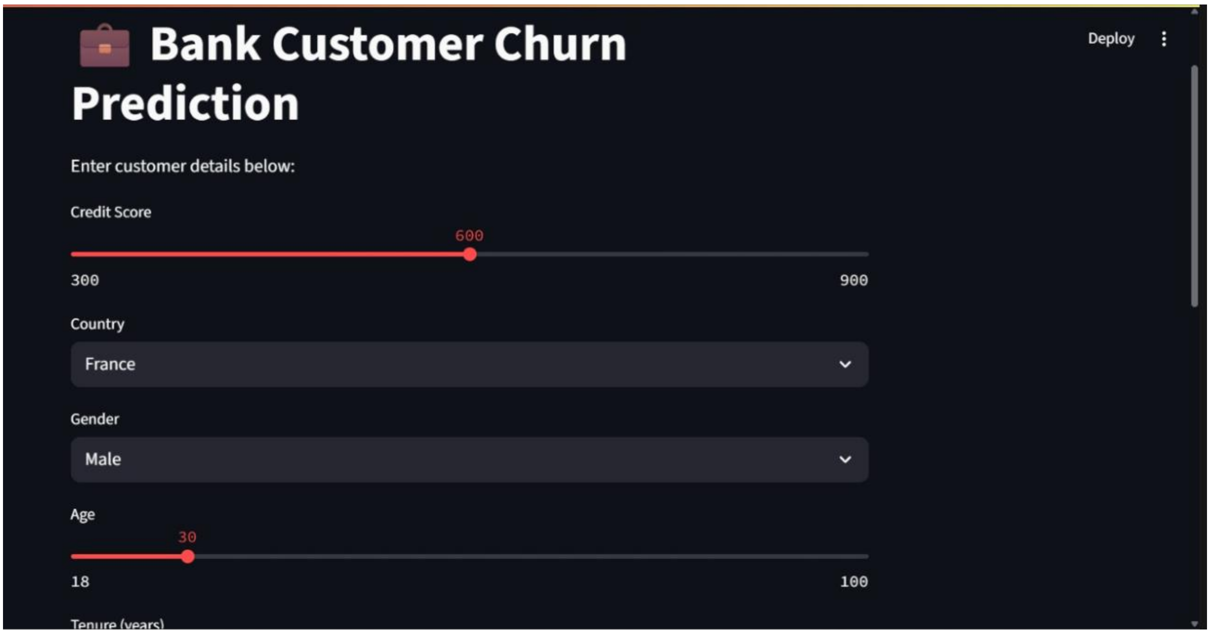
4. **Gender** shows a **mild negative correlation** with churn (-0.11), implying that **churn behavior may slightly vary by gender**, though this is relatively weak.

5. **Country** and **estimated salary** have **very low or negligible correlations** with churn, indicating that they **do not strongly influence customer retention** in this dataset.

6. **Credit card ownership**, **credit score**, and **tenure** also show minimal correlation with churn.

Conclusion:
The heatmap helps identify the most predictive features for the churn model. **Age**, **number of products**, and **active membership status** remain the most relevant factors, while features like **country**, **salary**, and **credit score** may have less predictive power in isolation.

- Screenshots:



Bank Customer Churn Prediction

Enter customer details below:

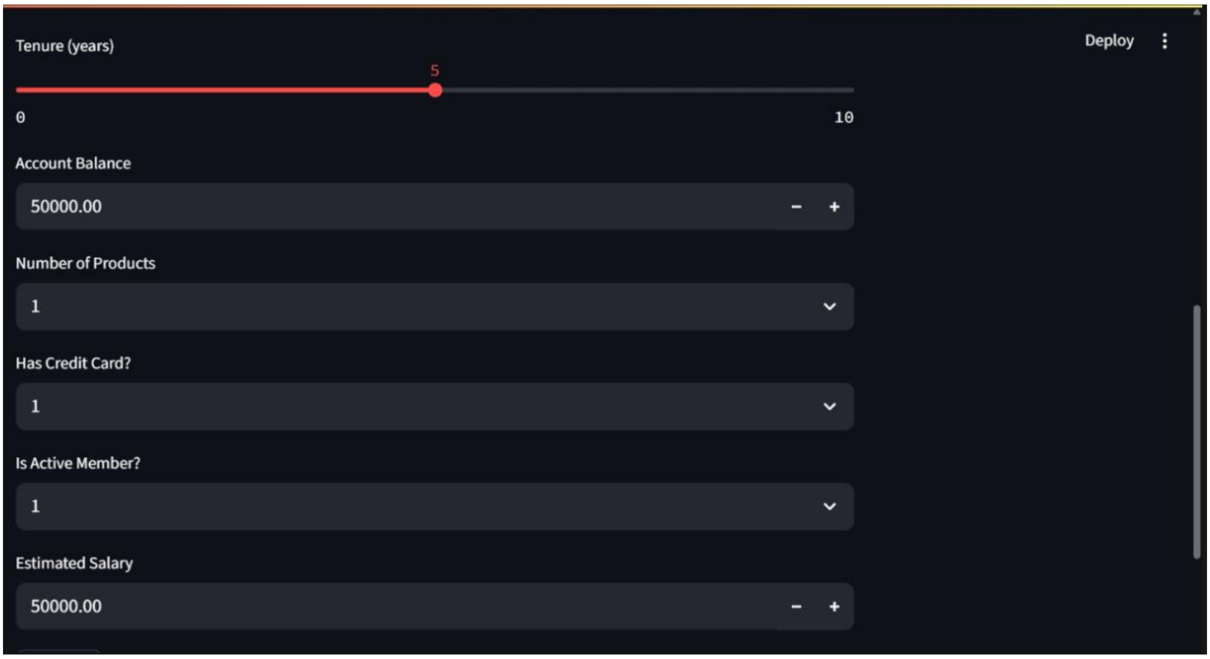
Credit Score: 600 (300–900)

Country: France

Gender: Male

Age: 30 (18–100)

Tenure (years)



Tenure (years)

5

0 10

Account Balance

50000.00 − +

Number of Products

1

Has Credit Card?

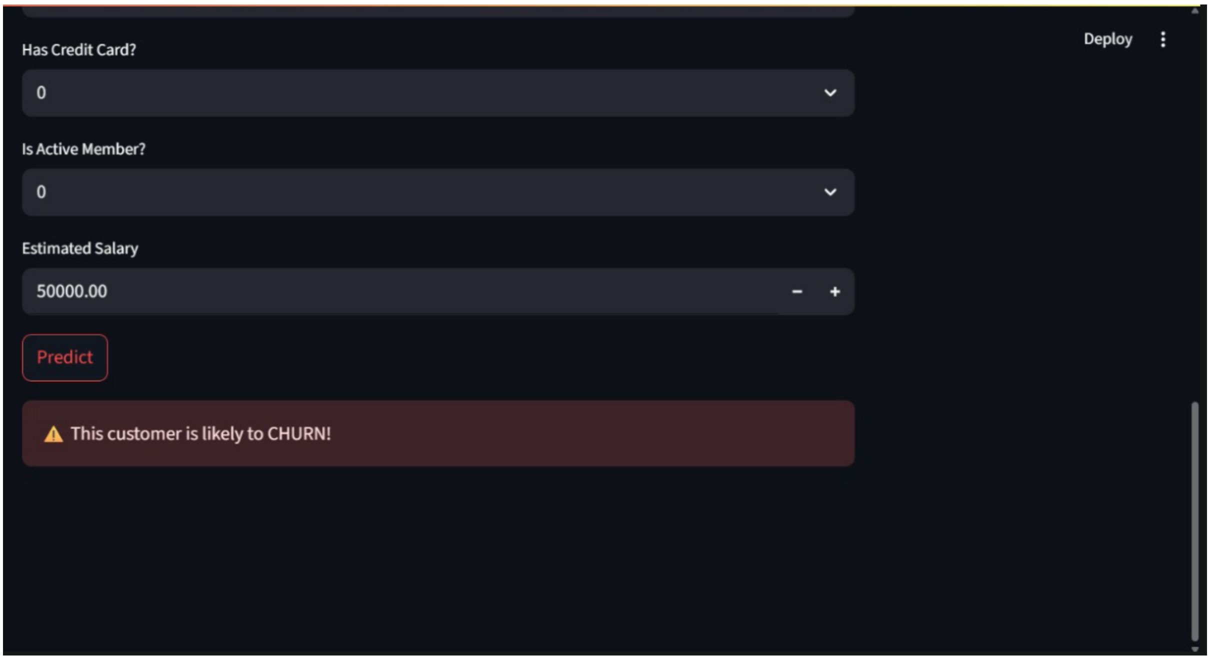
1

Is Active Member?

1

Estimated Salary

50000.00 − +



Has Credit Card?

0

Is Active Member?

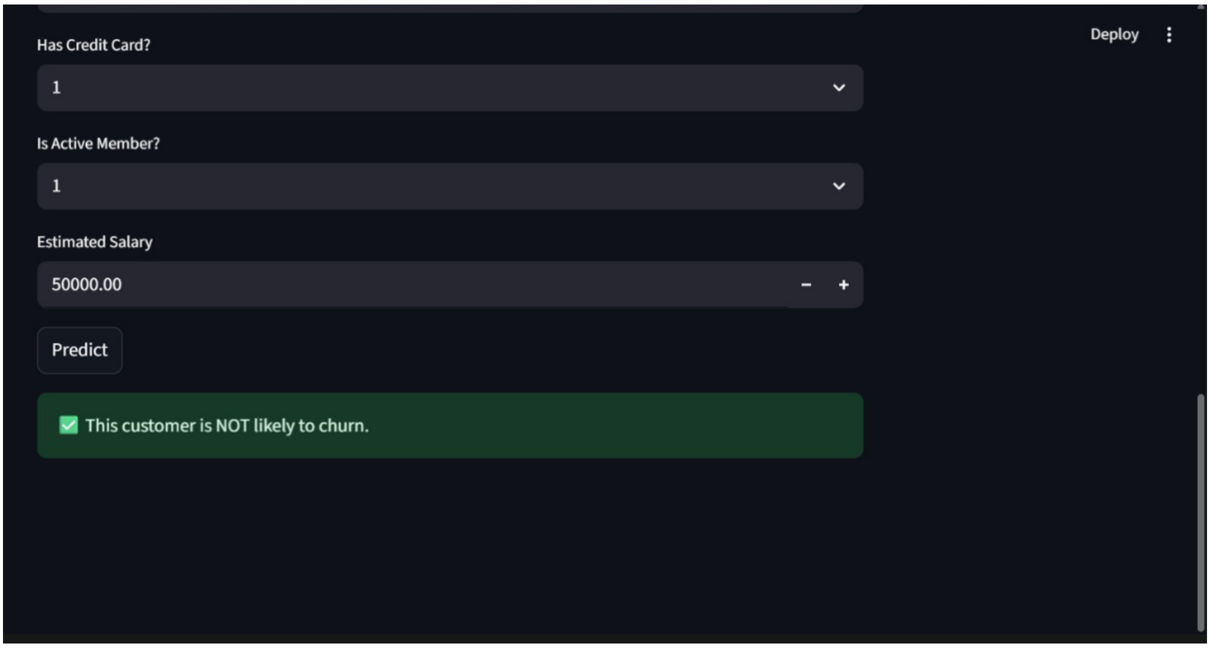
0

Estimated Salary

50000.00 − +

Predict

⚠ This customer is likely to CHURN!



Has Credit Card?

1

Is Active Member?

1

Estimated Salary

50000.00

✓ This customer is NOT likely to churn.

Chapter 4: Results And Discussion

- **Output:** The objective of this project was to develop a machine learning model capable of predicting customer churn in a bank. The following models were implemented:
    1. Logistic Regression
    2. Random Forest Classifier (Default and Tuned)
    3. The dataset was preprocessed by dropping irrelevant features (e.g., customer ID), encoding categorical variables (country, gender), and conducting Exploratory Data Analysis (EDA) through various visualizations.

Key Observations from Graphs:

1. **Churn is higher among older customers** and those with **fewer products**.

2. **Germany has a relatively high churn rate**, while **Spain shows better retention**.

3. **Customers with mid to high account balances are more likely to churn**, highlighting the importance of targeting valuable customers.

4. **Age, number of products, and active membership status** are the most correlated features with churn.

5. The tuned Random Forest model outperformed others, making it suitable for deployment.

- **Challenges Faced:**
  During the project, several challenges were encountered:

  1. **Data Quality Issues:** Some categorical variables had to be encoded manually, and a few features were found to have weak or no correlation with churn, requiring feature selection.

  2. **Class Imbalance:** Although the churn dataset was not extremely imbalanced, the number of non-churned customers significantly outweighed churned ones, making it harder to train some models accurately.

  3. **Model Tuning Complexity:** GridSearchCV with multiple parameters required considerable training time and computational power.

  4. **Choosing Evaluation Metrics:** Accuracy alone was not sufficient; hence, other metrics like F1 Score, ROC-AUC, and confusion matrices were also used for fair evaluation.

- **Learnings:**
  This project provided extensive hands-on experience in:

  1. Data preprocessing and feature engineering, including encoding and visual EDA.

  2. Applying multiple machine learning algorithms and understanding their behavior with real-world data.

  3. Hyperparameter tuning using GridSearchCV to improve model performance.

  4. Evaluating models using appropriate classification metrics beyond accuracy.

  5. Gaining business insights from visual analytics to support data-driven decision-making.

  6. Understanding the importance of targeting the right customer segment for improving retention strategies in a financial context.

Chapter 5: Conclusion

The primary aim of this project was to develop a machine learning-based system capable of accurately predicting bank customer churn. Through systematic data preprocessing, exploratory data analysis, and the application of classification models such as Logistic Regression and Random Forest, the project successfully identified patterns and key factors contributing to customer attrition.

The analysis revealed that age, number of products held, and active membership status are critical indicators of a customer's likelihood to churn. Additionally, geographic insights—such as the higher churn rate observed in Germany—provided meaningful direction for region-specific retention strategies. Among the models implemented, the tuned Random Forest classifier demonstrated superior predictive performance across key evaluation metrics, making it a strong candidate for deployment in a real-world setting.

Beyond technical development, this project fostered a deeper understanding of the entire machine learning workflow—from data acquisition to model optimization and performance evaluation. It highlighted the importance of not just building accurate models, but also interpreting results in a way that supports actionable business decisions.

In conclusion, this machine learning solution offers banks a data-driven approach to proactively manage customer relationships, reduce churn, and improve overall customer satisfaction. With further integration into business systems and periodic retraining using updated data, this model holds strong potential to become a valuable asset in strategic customer retention planning.