# Task 3: Customer Churn Prediction Analysis

## Machine Learning Project Report

---

## Project Overview

This project implements a machine learning solution for predicting customer churn in the banking sector. Using ensemble learning techniques, the analysis identifies customers likely to leave the bank, enabling proactive retention strategies and reducing customer acquisition costs.

### Dataset Description

The bank customer churn dataset contains customer demographics, account information, and service usage patterns:

- **Demographics**: Age, Gender, Geography (country)
- **Account Details**: Credit Score, Balance, Tenure
- **Product Usage**: Number of Products, HasCrCard, IsActiveMember
- **Financial**: EstimatedSalary
- **Target**: Exited (1 = churned, 0 = retained)

---

## Technical Implementation

### Libraries Used

- **pandas**: Data manipulation and analysis
- **matplotlib & seaborn**: Statistical visualization and plotting

- **scikit-learn components**:
  - `train_test_split`: Dataset partitioning
  - `LabelEncoder`: Binary categorical encoding
  - `RandomForestClassifier`: Ensemble learning algorithm
  - `confusion_matrix, classification_report, accuracy_score`: Performance evaluation

### Data Loading and Initial Setup

```python
df = pd.read_csv("Churn_Modelling.csv")
```

Dataset loaded directly from CSV file containing comprehensive customer information for churn analysis.

---

## Data Preprocessing Pipeline

### 1. Data Cleaning

```python
df.drop(["RowNumber", "CustomerId", "Surname"], axis=1, inplace=True)
```

**Purpose**: Remove non-predictive identifiers

- **RowNumber**: Sequential index with no predictive value
- **CustomerId**: Unique identifier irrelevant for modeling
- **Surname**: Personal identifier that could introduce bias

## 2. Categorical Encoding Strategy

### Binary Encoding for Gender:

python

```
le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])  # Male:1, Female:0
```

**Rationale**: Simple binary encoding for two-category variable

### One-Hot Encoding for Geography:

python

```
df = pd.get_dummies(df, columns=['Geography'], drop_first=True)
```

**Rationale**: Creates binary columns for each country, avoiding ordinal assumptions

- **drop_first=True**: Prevents multicollinearity by removing one category
- **Result**: Geography_Germany, Geography_Spain columns (France as baseline)

---

# Machine Learning Implementation

## 1. Feature-Target Separation

python

```
X = df.drop('Exited', axis=1)  # All features
y = df['Exited']              # Target variable
```

**Structure**: Clean separation of predictive features from target variable

## 2. Data Splitting Strategy

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

**Configuration**:

- **80-20 split**: Standard proportion for training vs testing
- **random_state=42**: Ensures reproducible results
- **Stratification**: Maintains class distribution across splits

## 3. Model Selection and Training

```python
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

**Algorithm Choice: Random Forest**

- **Ensemble Method**: Combines multiple decision trees
- **Handles Mixed Data**: Effective with numerical and categorical features
- **Feature Importance**: Provides interpretable feature rankings

- **Robust Performance**: Resistant to overfitting
- **No Scaling Required**: Tree-based methods handle different scales naturally

**Hyperparameters**:

- **n_estimators=100**: Sufficient trees for stable performance
- **random_state=42**: Reproducible model training

---

## Model Evaluation Framework

### 1. Prediction Generation

```python
y_pred = model.predict(X_test)
```

**Process**: Generate binary predictions on unseen test data

### 2. Comprehensive Performance Analysis

**Confusion Matrix:**

```python
confusion_matrix(y_test, y_pred)
```

**Analysis Components**:

- **True Negatives**: Correctly predicted retained customers
- **False Positives**: Incorrectly predicted churners (Type I error)

- **False Negatives**: Missed churners (Type II error)
- **True Positives**: Correctly identified churners

**Classification Report:**

```python
classification_report(y_test, y_pred)
```

**Detailed Metrics**:

- **Precision**: Proportion of predicted churners who actually churned
- **Recall**: Proportion of actual churners correctly identified
- **F1-Score**: Harmonic mean of precision and recall
- **Support**: Number of samples in each class

**Overall Accuracy:**

```python
accuracy_score(y_test, y_pred)
```

**Interpretation**: Percentage of correct predictions across both classes

---

## Feature Importance Analysis

### 1. Importance Extraction

```python
```

```
importances = pd.Series(model.feature_importances_, index=X.columns).sort_values(ascending=False)
```

**Process**: Extract and rank feature contributions from trained Random Forest

### 2. Visualization

```python
plt.figure(figsize=(10, 6))
sns.barplot(x=importances, y=importances.index)
```

**Purpose**: Visual representation of feature importance hierarchy

- **Horizontal bar chart**: Clear comparison of feature contributions
- **Descending order**: Most important features at top
- **Quantitative scale**: Numeric importance scores

---

## Expected Business Insights

### Customer Behavior Patterns

**High-Impact Factors** (typically include):

- **Age**: Older customers may have different loyalty patterns
- **Balance**: Account balance often correlates with churn risk
- **Number of Products**: Product usage indicates customer engagement
- **Geography**: Regional differences in customer behavior

- **IsActiveMember**: Activity level strongly predicts retention

### Risk Segmentation

- **High-Risk Customers**: Low engagement, single products, specific demographics
- **Low-Risk Customers**: High balance, multiple products, active usage
- **Geographic Patterns**: Country-specific churn tendencies

---

## Business Applications

### Proactive Retention Strategies

- **Targeted Campaigns**: Focus resources on high-risk customers
- **Personalized Offers**: Tailor retention incentives based on risk factors
- **Early Warning System**: Identify at-risk customers before they churn

### Resource Optimization

- **Cost Efficiency**: Reduce blanket retention spending
- **ROI Improvement**: Higher success rates with targeted interventions
- **Customer Lifetime Value**: Extend profitable customer relationships

### Strategic Planning

- **Product Development**: Address features driving churn
- **Market Segmentation**: Understand regional and demographic patterns
- **Service Improvements**: Focus on factors most impacting retention

---

## Technical Advantages

### Random Forest Benefits

- **Interpretability**: Clear feature importance rankings
- **Robustness**: Handles missing values and outliers well
- **No Preprocessing**: Minimal data preparation requirements
- **Balanced Performance**: Good results across different class distributions

### Scalability Considerations

- **Fast Predictions**: Efficient scoring for large customer bases
- **Parallel Processing**: Random Forest supports multi-threading
- **Memory Efficient**: Reasonable computational requirements
- **Easy Updates**: Simple retraining with new customer data

---

## Model Enhancement Opportunities

### Advanced Techniques

- **Hyperparameter Tuning**: Grid search for optimal parameters
- **Feature Engineering**: Create derived variables (balance-to-salary ratio)
- **Ensemble Stacking**: Combine multiple algorithm types
- **Time-Series Features**: Include temporal patterns in customer behavior

### Business Integration

- **Real-Time Scoring**: Live churn probability calculations

- **A/B Testing**: Validate retention campaign effectiveness
- **Feedback Loops**: Incorporate intervention outcomes into model
- **Threshold Optimization**: Balance false positives vs false negatives

---

## Conclusion

This customer churn prediction project demonstrates effective application of ensemble learning for business problem-solving. The Random Forest approach provides both strong predictive performance and interpretable insights into customer behavior drivers. The systematic preprocessing pipeline and comprehensive evaluation framework create a robust foundation for production deployment.

The feature importance analysis enables data-driven retention strategies, while the model's architecture supports scalable implementation across large customer bases. This solution bridges the gap between advanced machine learning techniques and practical business applications in customer relationship management.

---

**Project Type**: Binary Classification - Customer Churn Prediction
**Dataset**: Bank customer data with demographic and account features
**Algorithm**: Random Forest Classifier (100 estimators)
**Tools**: Python, pandas, scikit-learn, matplotlib, seaborn
**Key Output**: Feature importance rankings for retention strategy development