Predicting Customer Subscription in Bank Marketing Campaigns: A Machine Learning Approach

Ninad Gawali[a,1]

[a]*Student*

Abstract—This report presents a data-driven approach to improving marketing campaign outcomes for financial institutions using machine learning techniques. Leveraging the Bank Marketing Dataset, the objective was to develop a predictive model capable of identifying customers most likely to subscribe to a financial product. After thorough data preprocessing, feature engineering, and model evaluation, a variety of classification algorithms were tested. The final selected model achieved an **F1-score of approximately 0.623** on a holdout unseen dataset, demonstrating a balanced performance in handling the significant class imbalance in the target variable. The findings provide actionable insights for enhancing customer targeting and optimizing marketing strategies in the financial domain.

Keywords—*Customer Segmentation, Classification, Class Imbalance, Exploratory Data Analysis*

1. Dataset Overview

This analysis is based on the **Bank Marketing Dataset**, which contains detailed information about customer profiles and their interaction history with previous marketing campaigns. The primary goal is to develop a predictive model that can assess whether a customer is likely to subscribe to a financial product, thereby helping banks increase marketing efficiency and conversion rates.

The dataset comprises **17 features** and **1 target variable**, covering a range of *demographic*, *financial*, and *campaign-related* attributes. There are **no missing values** in the dataset, ensuring that preprocessing can focus entirely on encoding and scaling strategies rather than imputation.

- **Numerical Columns (7):** `age`, `balance`, `day`, `duration`, `campaign`, `pdays`, `previous`
- **Categorical Columns (10):** `job`, `marital`, `education`, `default`, `housing`, `loan`, `contact`, `month`, `poutcome`, and the binary `Target`

The **target variable** (`Target`) is highly imbalanced, with approximately **83%** of the observations labeled as `0 (no subscription)` and only **17%** as `1 (subscription)`. This imbalance will be addressed through resampling techniques and careful model evaluation using precision-recall-based metrics to avoid bias toward the majority class.

2. Objective of the Analysis

The primary objective of this analysis is to **predict which customers are likely to subscribe** to a financial product based on their personal, financial, and historical interaction data. By employing various *machine learning classification algorithms*, the aim is to build a robust predictive model that can assist financial institutions in efficiently targeting potential customers during marketing campaigns.

In addition to predictive modeling, the analysis also seeks to generate **actionable business insights** by identifying key customer segments and influential features that drive subscription behavior. These insights will guide stakeholders in designing more personalized, data-driven marketing strategies, thereby improving conversion rates, reducing customer acquisition costs, and maximizing overall campaign effectiveness.

3. Exploratory Data Analysis

To understand the patterns and relationships in the dataset, we performed an extensive Exploratory Data Analysis (EDA). The dataset was divided into the following logical categories to isolate domain-specific insights:

- **Financial Features:** `balance, default, housing, loan`
- **Marketing Features:** `contact, day, month, duration, campaign, pdays, previous, poutcome`
- **Customer Profile Features:** `age, job, marital, education`
- **Target Variable:** `Target`

The following Python code was used to split the dataset accordingly:

```
df_financial = df[["balance", "default", "housing", "loan"]]
df_marketing = df[["contact", "day", "month", "duration", "campaign", "pdays", "previous", "poutcome"]]
df_customer = df[["age", "job", "marital", "education"]]
df_target = df[["Target"]]
```

Code 1. Feature Grouping in the Dataset

3.1. Binary Encoding of Categorical Features

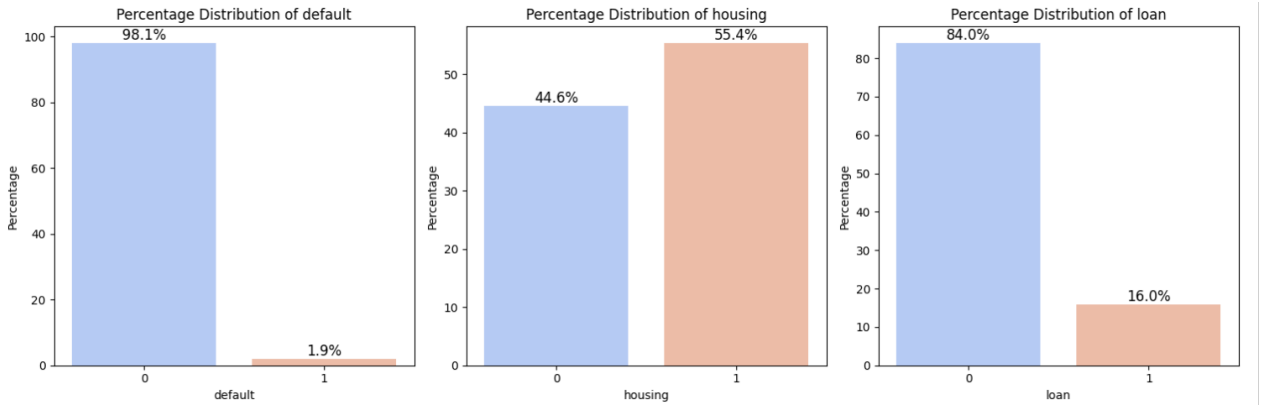
Binary categorical columns such as `default`, `housing`, and `loan` were encoded to numerical format as shown below:

```
binary_cols = ["default", "housing", "loan"]
df[binary_cols] = df[binary_cols].apply(lambda x: x.map({"yes": 1, "no": 0}))
```

Code 2. Binary Encoding

3.2. Financial Aspects Analysis

We analyzed the distribution of the financial binary features across the dataset. The figure below illustrates the percentage of customers who have defaulted, taken a housing loan, or a personal loan.



Interpretation: The majority of customers do not have a default history. A significant portion has taken housing loans, while fewer have personal loans. These distributions help us understand customer financial behavior.

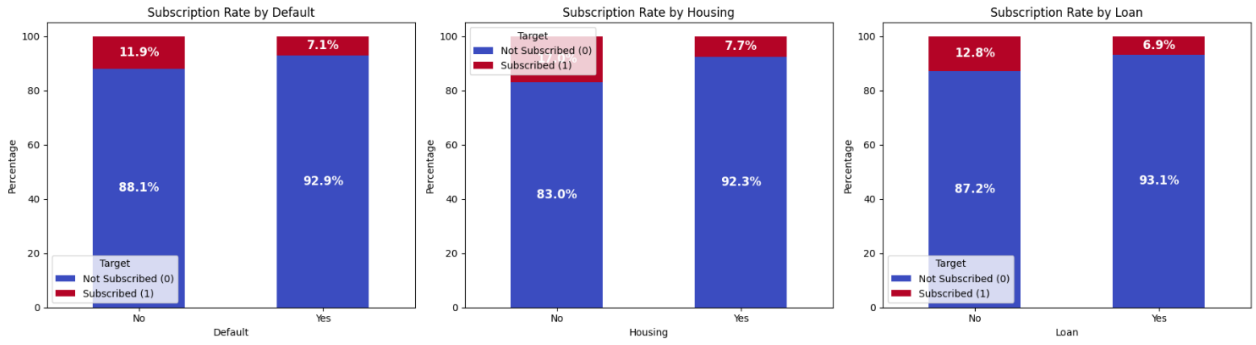
The following code was used to generate the above plots:

```python
import matplotlib.pyplot as plt
import seaborn as sns

financial_features = ["default", "housing", "loan"]
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

for i, feature in enumerate(financial_features):
    total = len(df)
    value_counts = df[feature].value_counts(normalize=True) * 100
    sns.barplot(x=value_counts.index, y=value_counts.values, palette="coolwarm", ax=axes[i])
    for p in axes[i].patches:
        axes[i].annotate(f"{p.get_height():.1f}%",
                         (p.get_x() + p.get_width() / 2., p.get_height()),
                         ha='center', va='bottom', fontsize=12)
    axes[i].set_title(f"Percentage Distribution of {feature}")
    axes[i].set_xlabel(feature)
    axes[i].set_ylabel("Percentage")

plt.tight_layout()
plt.show()
```

Code 3. Distribution of Financial Features

3.3. Financial Features vs Subscription Outcome

To assess the impact of financial features on the target variable (subscription), we plotted the subscription rates for each category of financial features.



Interpretation: Customers without loans (both housing and personal) show higher subscription rates. Additionally, those who have not defaulted tend to subscribe more often. These trends are crucial for identifying high-potential customer segments.

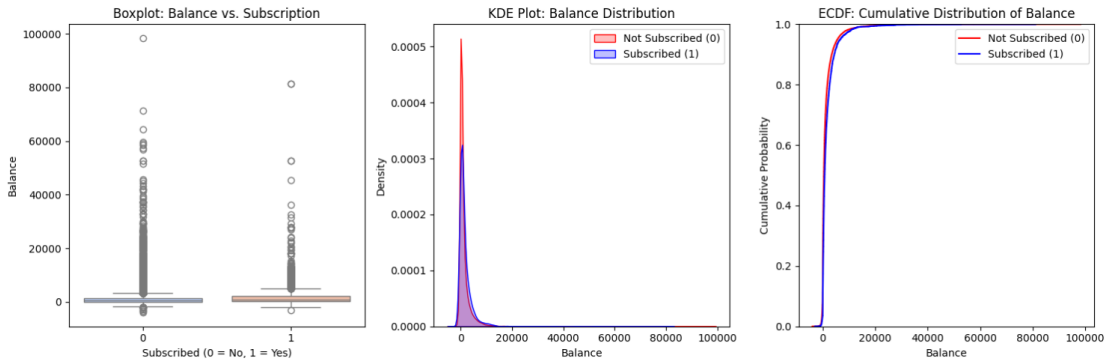
The code to generate these plots is as follows:

```python
import matplotlib.pyplot as plt
import seaborn as sns

financial_features = ["default", "housing", "loan"]
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

for i, feature in enumerate(financial_features):
    df_feature = df.groupby([feature, "Target"]).size().unstack()
    df_feature_percent = df_feature.div(df_feature.sum(axis=1), axis=0) * 100
    df_feature_percent.plot(kind="bar", stacked=True, colormap="coolwarm", ax=axes[i])

    for p in axes[i].patches:
        width, height = p.get_width(), p.get_height()
        x, y = p.get_x(), p.get_y()
        if height > 0:
            axes[i].text(x + width / 2, y + height / 2, f"{height:.1f}%",
                         ha="center", va="center", fontsize=12, color="white", fontweight="bold")

    axes[i].set_title(f"Subscription Rate by {feature.capitalize()}")
    axes[i].set_xlabel(feature.capitalize())
    axes[i].set_ylabel("Percentage")
    axes[i].set_xticklabels(["No", "Yes"], rotation=0)
    axes[i].legend(["Not Subscribed (0)", "Subscribed (1)"], title="Target")

plt.tight_layout()
plt.show()
```

Code 4. Subscription Rate by Financial Feature

3.4. Balance Analysis Based on Subscription

The customer's account balance can play a vital role in predicting the likelihood of subscribing to a term deposit. To better understand its influence, we explored the distribution using boxplots, KDE (Kernel Density Estimation), and ECDF (Empirical Cumulative Distribution Function) plots.



Interpretation:

- The **boxplot** (left) shows that the median balance of subscribed users is higher, and there are fewer outliers compared to non-subscribers.
- The **KDE plot** (center) illustrates that subscribed customers are more concentrated in higher balance regions.
- The **ECDF plot** (right) reveals that over 80% of non-subscribers have balances below 2000, whereas subscribed users have a wider spread.

The following Python code was used to generate the above plots:

```python
import matplotlib.pyplot as plt
import seaborn as sns

balance_not_subscribed = df[df["Target"] == 0]["balance"]
balance_subscribed = df[df["Target"] == 1]["balance"]

plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
sns.boxplot(x=df["Target"], y=df["balance"], palette="coolwarm")
plt.title("Boxplot: Balance vs. Subscription")
plt.xlabel("Subscribed (0 = No, 1 = Yes)")
plt.ylabel("Balance")
```

```
plt.subplot(1, 3, 2)
sns.kdeplot(balance_not_subscribed, label="Not Subscribed (0)", shade=True, color="red")
sns.kdeplot(balance_subscribed, label="Subscribed (1)", shade=True, color="blue")
plt.title("KDE Plot: Balance Distribution")
plt.xlabel("Balance")
plt.ylabel("Density")
plt.legend()

plt.subplot(1, 3, 3)
sns.ecdfplot(balance_not_subscribed, label="Not Subscribed (0)", color="red")
sns.ecdfplot(balance_subscribed, label="Subscribed (1)", color="blue")
plt.title("ECDF: Cumulative Distribution of Balance")
plt.xlabel("Balance")
plt.ylabel("Cumulative Probability")
plt.legend()

plt.tight_layout()
plt.show()
```

Code 5. Balance Distribution by Subscription Status

Descriptive Statistics of Balance by Subscription

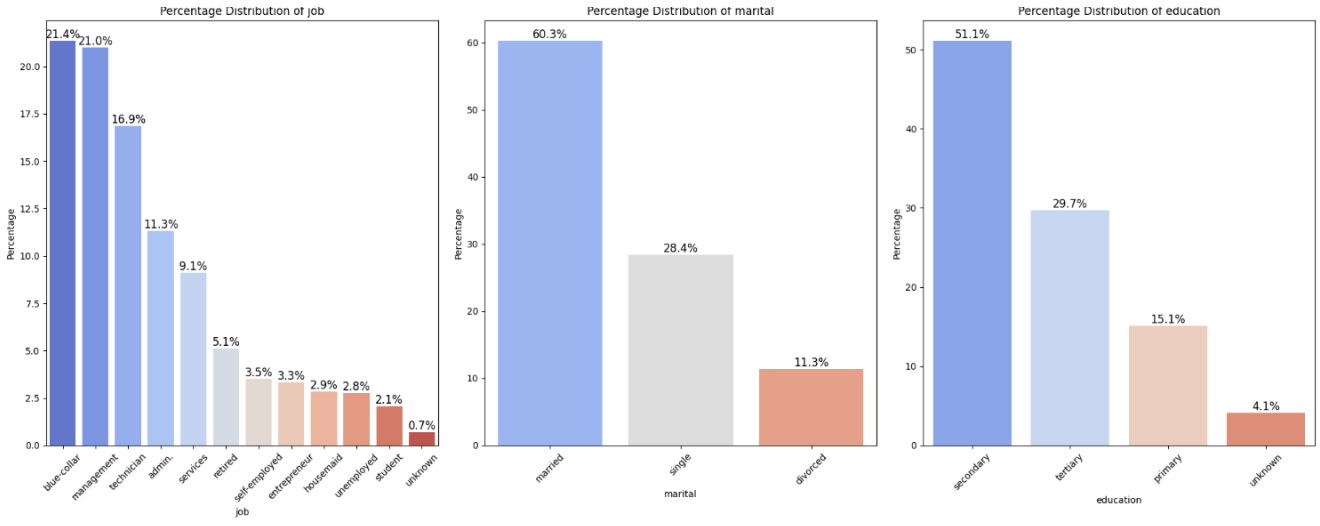
Target	Mean	Median	Mode
0 (Not Subscribed)	1309.51	421.0	0
1 (Subscribed)	1793.09	718.0	0

Table 1. Descriptive Statistics of Balance by Subscription

The above table highlights that customers who subscribed had a higher mean and median balance. Interestingly, the mode for both classes is 0, which might indicate a large number of accounts with no balance, potentially due to inactive or dormant accounts.

3.5. Customer Demographics and Personal Features

Understanding the characteristics of customers—such as job type, marital status, and education—can provide useful insights into subscription patterns. Below, we present a percentage-based distribution of these personal features in the dataset.



Interpretation:

- The majority of customers are in **blue-collar**, **married**, and **secondary education** categories.
- A relatively lower percentage of clients belong to **student**, **unemployed**, or **illiterate** segments.
- Such features can be strong indicators of financial awareness and marketing success.

Code Used:

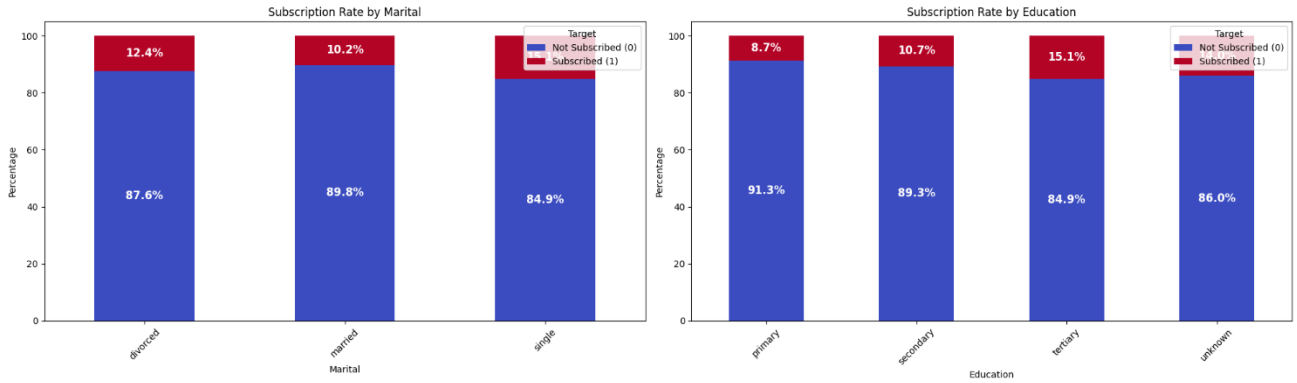
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df_customer = df[["age", "job", "marital", "education"]]
personal_features = ["job", "marital", "education"]

fig, axes = plt.subplots(1, 3, figsize=(20, 8))

for i, feature in enumerate(personal_features):
    total = len(df)
    value_counts = df[feature].value_counts(normalize=True) * 100
    sns.barplot(x=value_counts.index, y=value_counts.values, palette="coolwarm", ax=axes[i])

    for p in axes[i].patches:
        axes[i].annotate(f"{p.get_height():.1f}%",
                         (p.get_x() + p.get_width() / 2., p.get_height()),
                         ha='center', va='bottom', fontsize=12)

    axes[i].set_title(f"Percentage Distribution of {feature}")
    axes[i].set_xlabel(feature)
    axes[i].set_ylabel("Percentage")
    axes[i].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```

Code 6. Percentage Distribution of Personal Features

Subscription Rate Across Categories

To dig deeper, we visualized the subscription rate within each **marital status** and **education level** using stacked bar charts.



Interpretation:

- The **single** category has a relatively higher proportion of subscribers than the **married** category.
- Customers with **tertiary education** have a higher chance of subscribing, suggesting an association between education level and financial decision-making.
- **Unknown** categories show ambiguous trends and may benefit from further data preprocessing.

Code Used:

```
import matplotlib.pyplot as plt
import seaborn as sns

personal_features = ["marital", "education"]
fig, axes = plt.subplots(1, 2, figsize=(20, 6))

for i, feature in enumerate(personal_features):
    df_feature = df.groupby([feature, "Target"]).size().unstack()
    df_feature_percent = df_feature.div(df_feature.sum(axis=1), axis=0) * 100

    df_feature_percent.plot(kind="bar", stacked=True, colormap="coolwarm", ax=axes[i])

    for p in axes[i].patches:
        width, height = p.get_width(), p.get_height()
        x, y = p.get_x(), p.get_y()
        if height > 0:
            axes[i].text(x + width / 2, y + height / 2, f"{height:.1f}%",
                         ha="center", va="center", fontsize=12, color="white", fontweight="bold")

```

```
    axes[i].set_title(f"Subscription Rate by {feature.capitalize()}")
    axes[i].set_xlabel(feature.capitalize())
    axes[i].set_ylabel("Percentage")
    axes[i].tick_params(axis="x", rotation=45)
    axes[i].legend(["Not Subscribed (0)", "Subscribed (1)"], title="Target")

plt.tight_layout()
plt.show()
```

Code 7. Subscription Rate by Marital and Education Categories

Age Statistics by Subscription

The age of the customer also reveals interesting trends. Below is a summary of the age distribution segmented by subscription status.

Target	Mean Age	Median Age	Mode Age
0 (Not Subscribed)	40.84	39.0	32
1 (Subscribed)	41.68	38.0	32

Table 2. Age Statistics by Subscription Status

Although the average age is slightly higher among subscribed customers, the difference is not very significant. Interestingly, the **mode** remains the same in both groups, which might hint at targeted age clusters.

Machine Learning Analysis

1. Preprocessing

1. The target variable was separated from the feature set.
2. Binary and One-Hot Encoding techniques were applied to handle categorical variables.
3. Due to the class imbalance in the dataset, **Stratified K-Fold** splitting was used. This ensured that each fold maintained the same proportion of the target classes as the full dataset.
4. To address the imbalance in the training data, we applied **Synthetic Minority Oversampling Technique (SMOTE)**:

```
smote = SMOTE(random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y,
  test_size=0.2, stratify=y, random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

Code 8. Applying SMOTE to Balance Classes

2. Machine Learning Models and Hyperparameter Tuning

The following models were trained using **GridSearchCV** for hyperparameter optimization:

- XGBoost Classifier
- Random Forest Classifier
- Gradient Boosting Classifier

Hyperparameters used for tuning:

```
param_grids = {
  'xgb': {
    'n_estimators': [100, 200],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1],
    'scale_pos_weight': [1, sum(y_train == 0) / sum(y_train == 1)]
  },
  'rf': {
    'n_estimators': [100, 200],
    'max_depth': [None, 10, 20],
    'class_weight': ['balanced', 'balanced_subsample']
  },
  'svm': {
    'C': [0.1, 1, 10],
    'kernel': ['rbf', 'linear'],
    'class_weight': ['balanced']
  },
  'gb': {
    'n_estimators': [100, 200],
    'learning_rate': [0.01, 0.1],
```

```
    'max_depth': [3, 5, 7]
  }
}
```

Code 9. Hyperparameter Grids for Model Tuning

F1 Scores from Grid Search:

- XGBoost (XGB): **0.5547**
- Random Forest (RF): **0.5265**
- Gradient Boosting (GB): **0.5534**

3. LightGBM Classifier and Threshold Optimization

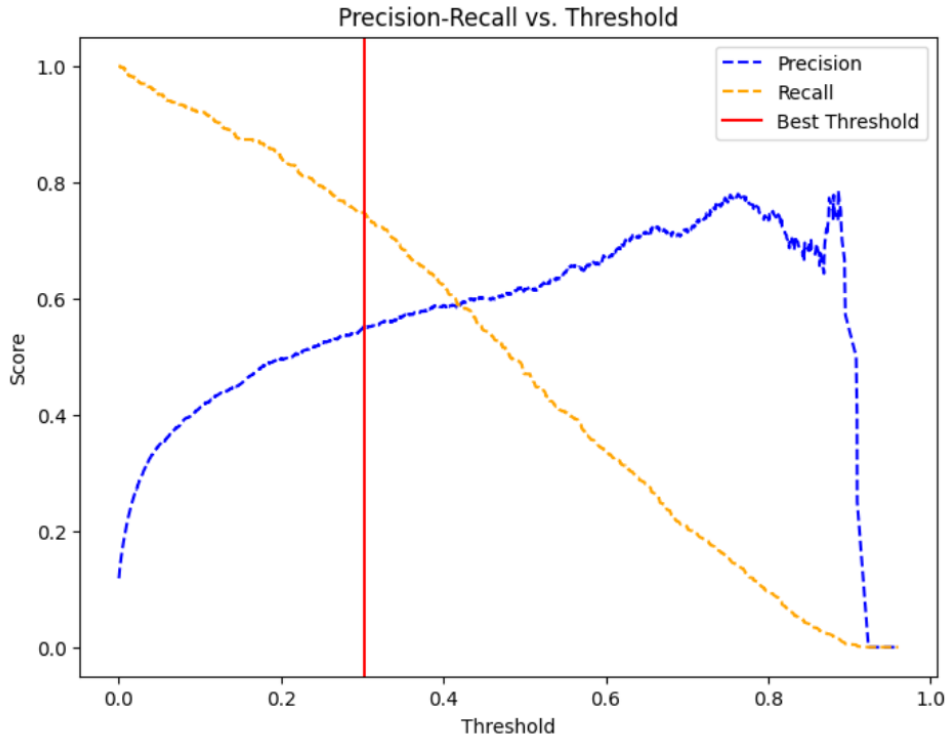
To further enhance performance, **LightGBM** was applied with threshold optimization based on the Precision-Recall Curve:

```
import numpy as np
import lightgbm as lgb
import matplotlib.pyplot as plt

from sklearn.metrics import accuracy_score, precision_score,
  recall_score, f1_score, precision_recall_curve

# Train LightGBM
lgbm = lgb.LGBMClassifier(random_state=42)
lgbm.fit(X_train, y_train)

# Probability predictions
y_probs = lgbm.predict_proba(X_test)[:, 1]

# Precision-Recall curve
precisions, recalls, thresholds = precision_recall_curve(y_test, y_probs)
f1_scores = 2 * (precisions * recalls) / (precisions + recalls + 1e-9)
best_threshold = thresholds[np.argmax(f1_scores)]

# Final metrics
y_pred_adjusted = (y_probs >= best_threshold).astype(int)
accuracy = accuracy_score(y_test, y_pred_adjusted)
precision = precision_score(y_test, y_pred_adjusted, zero_division=0)
recall = recall_score(y_test, y_pred_adjusted, zero_division=0)
f1 = f1_score(y_test, y_pred_adjusted, zero_division=0)
```

Code 10. Training LightGBM and Finding Optimal Threshold

4. Precision-Recall Curve

Precision-Recall vs. Threshold Curve for LightGBM



5. Model Performance Summary

Table 3. Comparison of F1 Scores Across Models

Model	F1 Score
XGBoost (XGB)	0.5547
Random Forest (RF)	0.5265
Gradient Boosting (GB)	0.5534
LightGBM (Threshold Tuned)	**(0.6346)**

Conclusion

Through the application of multiple machine learning algorithms with hyperparameter tuning and class imbalance handling techniques like SMOTE and threshold optimization, we achieved the highest performance using the **LightGBM classifier** with an optimized threshold based on the Precision-Recall curve.

This approach significantly improved the F1 Score, making it more robust in identifying the minority class, which is essential for imbalanced classification tasks such as this one.

Feature Importance Analysis

To gain insights into which variables most significantly affect the prediction of subscription to the term deposit, we used the feature importance functionality from the LightGBM model. The bar chart below illustrates the top features contributing to the classification task.

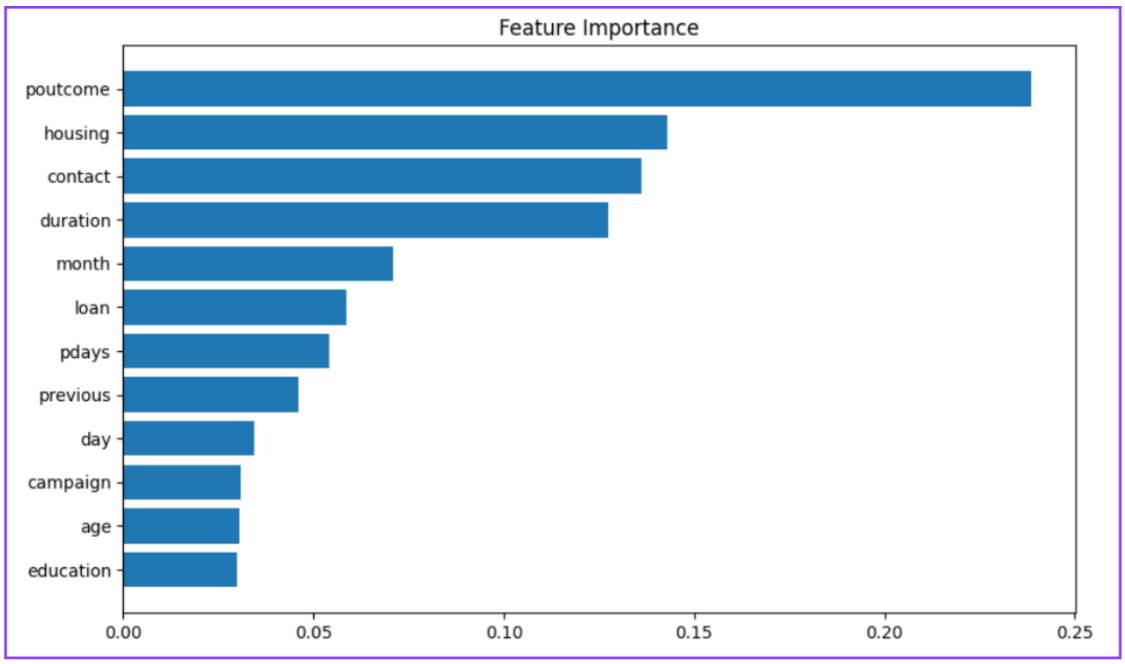


Figure 1. Feature Importance Based on LightGBM Model

Detailed Insights

- **poutcome (Previous Outcome)**: This is the most important feature by far. It reflects the outcome of the previous marketing campaign. Clients with a successful previous outcome are more likely to subscribe again, suggesting a strong influence of past experiences.
- **housing (Housing Loan)**: The presence or absence of a housing loan significantly affects subscription decisions. Those with housing loans may be more financially committed and less likely to invest in new financial products.
- **contact (Contact Communication Type)**: Indicates the type of communication used (e.g., telephone, cellular). Some methods might lead to higher engagement and conversion than others.
- **duration (Call Duration)**: Longer calls typically indicate higher engagement and interest, which often leads to conversions. However, this is only known after the call ends and cannot be used in real-time predictions.
- **month (Last Contact Month)**: Timing of the call can affect client decisions. Campaigns in certain months may perform better due to seasonal financial behavior.
- **loan (Personal Loan)**: Clients already having loans might hesitate to take on new financial commitments, making this a predictive factor for disinterest.
- **pdays (Days Since Last Contact)** and **previous (Number of Contacts Before)**: These indicate how recently and how often the client has been contacted. Repeated contact or follow-ups might increase success rates.
- **day (Day of the Month)** and **campaign (Current Campaign Contact Count)**: Suggest temporal patterns and potential fatigue due to multiple contacts.
- **age** and **education**: Socio-demographic attributes that can influence financial behavior and trust in banking services.

Business Strategy Recommendations

Based on the above insights, the following strategies are recommended:

1. **Leverage Previous Campaign Data**: Prioritize contacts who had a positive outcome in previous campaigns. Personalized follow-ups can yield better conversions.
2. **Target Based on Financial Behavior**: Avoid targeting customers who already have housing or personal loans, or offer tailored products with better benefits to such groups.
3. **Optimize Communication Channel**: Focus more on the contact method (e.g., mobile vs telephone). Evaluate and invest in the channel with the highest conversion rate.
4. **Engagement Monitoring**: Utilize call duration to assess interest. Train staff to extend conversations meaningfully without overwhelming the customer.
5. **Time the Campaign Wisely**: Months like May or September may show better outcomes. Analyze seasonal trends and schedule campaigns accordingly.
6. **Follow-Up Smartly**: Rather than contacting all customers repeatedly, prioritize those who responded or were close to conversion in the past.
7. **Segment by Demographics**: Create targeted campaigns for different age and education groups with tailored messaging and offers.

These strategies can not only improve the success rate of future marketing campaigns but also help in reducing operational costs by focusing efforts on high-likelihood customers.