📑 Bank Marketing Campaign – Machine Learning Project Report

👨‍💼 Project By : Anirudha Johare

🔗 GitHub: [Bank Marketing ML Model]

1. 🎯 Objective

The goal of this project is to build a predictive model that can help a Portuguese bank identify whether a customer will subscribe to a term deposit (**yes** or **no**) as part of a marketing campaign. This will help the bank save costs and improve its targeting strategy using data-driven decisions.

2. 📚 Dataset Information

- **Source**: UCI Machine Learning Repository
- **Dataset Size**: 41,188 records and 21 attributes
- **Target Variable**: y – binary classification (yes or no)
- **Features**:
  - **Categorical**: job, marital, education, default, housing, loan, contact, month, day_of_week, poutcome
  - **Numerical**: age, duration, campaign, pdays, previous

3. 🔍 Exploratory Data Analysis (EDA)

- Performed data quality check for missing/null values – none found
- Univariate and bivariate analysis done using:
  - Histograms
  - Count plots
  - Box plots for outlier detection
- Key Observations:
  - Duration has a strong influence on subscription.
  - Previous outcomes of marketing campaigns impact current success.
  - Age groups above 30 have higher conversion rates.

4. 📈 Data Preprocessing

- Handled missing values (none in this dataset)
- Converted categorical features using LabelEncoder / OneHotEncoder
- Removed outliers using IQR method for continuous variables
- Feature scaling using StandardScaler
- **SMOTE** used to balance the imbalanced dataset (original y was skewed towards "No")

5. 🤖 Model Building

1. Logistic Regression

- **Accuracy**: ~88%
- **Precision**: 0.76
- **Recall**: 0.70
- **F1 Score**: 0.73

2. Random Forest (Before Tuning)

- **Accuracy**: ~90%
- Handled interactions better than Logistic Regression

3. Random Forest (After Hyperparameter Tuning)

- **Technique**: GridSearchCV
- **Parameters Tuned**:
  - n_estimators: 100 to 300
  - max_depth: 10 to 30
  - min_samples_split: 2 to 10

- ✅ **Final Accuracy**: **~93%**
- ✅ **ROC-AUC Score**: **0.96**
- ✅ **F1 Score**: **0.91**

4. XGBoost (Default Settings)

- Accuracy: ~91%
- Highly effective, but slightly behind tuned Random Forest

🧪 Model Evaluation

- ROC Curve plotted for all models
- Confusion matrix used to compare True Positive, False Positive, etc.

✅ Final Model Chosen: Tuned Random Forest

- Excellent balance between bias and variance
- Better interpretability compared to XGBoost
- Robust against overfitting due to cross-validation and pruning via max_depth

5. 📈 Results Summary

⚙️ Best Performing Model: Random Forest (After

Hyperparameter Tuning)

Metric	Value
Accuracy	**93.4%**
Precision	91.2%
Recall (Sensitivity)	94.7%
F1 Score	92.9%
ROC-AUC Score	0.96
Confusion Matrix	True Positives and Negatives balanced with low False Negatives

🔍 Why Random Forest Worked Best:

- Captures complex feature interactions.
- Robust to overfitting with tuned hyperparameters (n_estimators, max_depth, min_samples_split).
- Performed well after SMOTE balanced the dataset.

🧠 Techniques Used:

- **Data Preprocessing:** Label Encoding, One-Hot Encoding, Missing Value Handling
- **SMOTE:** Balanced the target class distribution
- **Model Evaluation:** Cross-validation, ROC-AUC, Confusion Matrix
- **Model Tuning:** Grid Search CV on Random Fores

📈 Business Impact:

- Helps bank identify potential term deposit subscribers with high accuracy.
- Can be used to optimize marketing efforts, reduce costs, and improve conversion rates.

6. 💡 Key Learnings

- **SMOTE** significantly improved recall without hurting precision
- Hyperparameter tuning plays a major role in boosting model accuracy
- Improved model interpretability using feature importance plots
- Learned effective preprocessing techniques for mixed data types
- Developed strong understanding of model evaluation beyond accuracy
- Built an end-to-end ML pipeline — from loading data to model optimization

📫 Contact & Profiles

- **GitHub**: Anirudha Johare
- **LinkedIn**: linkedin.com/in/anirudhajohare19