Final Project Report

Title: Bank Marketing Classification and Customer Churn Prediction – Reproducing and Extending a Research Paper
Course: DS 301 – Machine Learning Techniques
Student: Moataz Samara
Instructor: Mirza Zaeem Baig
Date: adddateadd dateadddate

---

1. Summary

This project focuses on reproducing and extending a real research paper on bank telemarketing.
The original paper uses the UCI **Bank Marketing** dataset to predict whether a client will subscribe to a **term deposit** after a phone call. The main models in the paper are **Logistic Regression** and **Decision Tree**.

In my project, I first rebuild the methodology described in the paper for the Bank Marketing dataset using modern **scikit-learn Pipelines**. I then **extend** the work in three main directions:

1. I add **SMOTE** to handle class imbalance and compare the performance of Logistic Regression with and without SMOTE.

2. I experiment with **additional classification models** such as Random Forest, SVM, KNN and a simple Neural Network (MLP), to see if they can outperform the original models.

3. I apply the same overall methodology to a second dataset – a **Bank Customer Churn** dataset from Kaggle – in order to test if the approach generalizes to a different but related business problem.

The main evaluation metric across all experiments is **ROC-AUC**, because it is suitable for imbalanced binary classification.

The key findings are:

- For Bank Marketing, **Logistic Regression** clearly outperforms the Decision Tree (ROC-AUC ≈ 0.77 vs 0.61).

- **SMOTE** slightly changes ROC-AUC (0.77 to 0.76) but can help recall for the minority class.

- On the churn dataset, all models perform extremely well, with **Logistic Regression** again slightly best (ROC-AUC ≈ 0.999).

---

2. Motivation

The motivation for this project is both **academic** and **practical**.

From an academic perspective, I wanted to practise the full workflow of a real data science project:

- reading a research paper,
- reproducing the methodology,
- implementing models with proper preprocessing and evaluation, and
- making my own contributions and extensions.

From a business perspective, the two problems – **term deposit subscription** and **customer churn** – are very important for banks:

- In telemarketing, banks want to call the **right customers**, reduce the number of calls, and increase the probability of success.
- In churn prediction, banks want to **identify customers at risk of leaving** so they can offer special deals or better service in advance.

I chose this paper and these datasets because they are:

- publicly available,
- well-known in the machine learning community, and
- directly aligned with what we learned in DS 301 (Logistic Regression, Decision Trees, SVM, Random Forest, SMOTE, etc.).

---

3. Research Paper Details

The research paper I used as a starting point is titled:

“A Data Driven Approach to Predict the Success of Bank Telemarketing”

The paper uses the **Bank Marketing** dataset from the UCI Machine Learning Repository. In this dataset, each row corresponds to a client contacted by the bank via phone, and the target variable `y` indicates whether the client subscribed to a **term deposit** (“yes” or “no”).

Key points from the original paper:

- **Objective:** predict whether a telemarketing call will lead to a successful term deposit subscription.
- **Data:** characteristics of clients (age, job, marital status, education, etc.) and information about the current and previous campaigns (contact type, day, month, duration, outcome of previous contacts, etc.).
- **Models:** the paper mainly focuses on **Logistic Regression** and **Decision Trees** as classification algorithms.
- **Evaluation:** they evaluate models using metrics such as accuracy and ROC-AUC and discuss the trade-off between interpretability and performance.

My work uses the same core idea – predicting `y` – but is implemented fully in **Python and scikit-learn** and is extended with new models, SMOTE and an additional dataset.

---

4. Dataset Details

4.1 Bank Marketing Dataset (UCI)

- **Source:** UCI Machine Learning Repository
- **Shape:** ≈ 45,000 rows and 17 input features
- **Target:** `y` – term deposit subscription (`yes`/`no`)
- **Feature types:**
    - **Numeric:** age, balance, day, duration, campaign, pdays, previous

    - **Categorical:** job, marital, education, default, housing, loan, contact, month, poutcome

The target is **imbalanced**: there are many more “no” responses than “yes” responses. This motivated the use of **stratified splitting** and **SMOTE**.

4.2 Bank Customer Churn Dataset (Kaggle)

- **Source:** Kaggle – Customer-Churn-Records.csv
- **Shape:** ≈ 10,000 rows and around 10–12 input features
- **Target:** `Exited` – 1 if the customer churned, 0 otherwise
- **Feature types:**
    - **Numeric:** credit score, age, tenure, balance, estimated salary
    - **Categorical:** geography, gender, number of products, whether the customer has a credit card or is an active member

This dataset is used to test if the methodology from the Bank Marketing problem can **generalize** to another banking classification task.

---

5. Data Preprocessing and Feature Engineering

For both datasets, I followed a consistent preprocessing strategy implemented in **scikit-learn Pipelines**:

1. **Dropping irrelevant or leakage features**
    - In the Bank Marketing dataset, I dropped the `duration` feature because it is only known **after** the call and therefore leaks information from the future.
    - In the churn dataset, I removed pure ID columns that do not contain predictive information.
2. **Handling missing values**

    - Both datasets have very few missing values, so the main approach was to use scikit-learn's default handling inside the pipeline or simple imputers if needed.

3. **Encoding categorical variables**
    - All categorical features were transformed using **One-Hot Encoding** inside a `ColumnTransformer`.

4. **Scaling numeric features**
    - Numeric features were scaled using `StandardScaler`, which is important for models such as Logistic Regression and SVM.

5. **Train/test split**
    - I used a **70/30 train–test split** with `stratify=y` (or `stratify=Exited`) to preserve the class distribution in both sets.

6. **Feature engineering**
    - No heavy feature engineering was required, but I ensured that all features were in a clean numeric form and that no information leakage occurs.

Critical observations:

- The Bank Marketing dataset is **strongly imbalanced**, which affects performance for the minority "yes" class.
- Pipelines help to keep preprocessing and modeling together, making it easier to reproduce experiments and avoid mistakes.

---

6. Steps Reproduced from the Paper

From the original research paper, I reproduced the following key elements:

1. **Prediction problem:** Predicting term deposit subscription (`y`), using the same UCI Bank Marketing dataset.

2. **Models:** Implemented **Logistic Regression** and **Decision Tree** as the main models, similar to the paper.

3. **Preprocessing logic:** Encode categorical features, scale numeric features, and perform a train–test split.

4. **Evaluation:** Used metrics similar to the paper, focusing on **ROC-AUC** to compare models.

In my notebook, these steps are implemented using:

- `Pipeline` and `ColumnTransformer` for combined preprocessing + model,
- Evaluation functions that print accuracy, precision, recall, F1-score, classification report, confusion matrix and ROC curves, and
- Tables summarizing ROC-AUC results for the original models.

---

7. Contributions

My main contributions beyond simply reproducing the paper are:

1. **Pipeline-based implementation:** I implemented the entire workflow using **scikit-learn Pipelines**, which makes the process cleaner, modular, and easier to reuse.

2. **SMOTE experiment for class imbalance:**

    - Applied **SMOTE (Synthetic Minority Over-sampling Technique)** on the Bank Marketing dataset.

    - Trained Logistic Regression with and without SMOTE to study the effect on performance.

3. **Extended model comparison:**

    - Added additional models beyond Logistic Regression and Decision Tree:

        - k-Nearest Neighbors (KNN)

        - Random Forest

        - SVM
        - A simple Neural Network (MLP)
    - Tuned their hyperparameters using smaller **GridSearchCV** setups to keep training time reasonable.

4. **Second dataset – Customer Churn:**
    - Applied the same methodology (preprocessing, models, evaluation) to a **Bank Customer Churn** dataset from Kaggle.
    - Compared multiple models on churn and analyzed their ROC-AUC scores.

5. **Unified reporting of results:**
    - Created comparison tables for Bank Marketing models, SMOTE vs no-SMOTE, and churn models, linking everything back to the course concepts.

---

8. Significant Improvements

The most significant improvements and insights from my contributions are:

1. **Model performance insights on Bank Marketing:**
    - Logistic Regression reaches **ROC-AUC ≈ 0.77**, clearly outperforming the Decision Tree (**≈ 0.61**).
    - This confirms the original paper's conclusion and adds a clean, reproducible implementation.

2. **Deeper understanding of SMOTE:**
    - Logistic Regression without SMOTE: **ROC-AUC ≈ 0.77**
    - Logistic Regression with SMOTE: **ROC-AUC ≈ 0.76**
    - This shows that **SMOTE does not always improve ROC-AUC**, although it can help recall for the minority class. This is an important practical lesson.

3. **Strong generalization to churn dataset:**

- On the churn dataset, all models achieve extremely high ROC-AUC:
  - Logistic Regression: **≈ 0.9990**
  - Random Forest: **≈ 0.9988**
  - SVM: **≈ 0.9976**
  - Decision Tree: **≈ 0.9953**
- This demonstrates that the same methodology from the research paper can work very well on a different banking problem.

4. **Broader model comparison:**
   - By including Random Forest, SVM and MLP, I show that while more complex models can perform very well, **a simple Logistic Regression is already extremely strong**, especially on churn.

---

9. Challenges

During the project I faced several challenges:

1. **Class imbalance:**
   - The Bank Marketing dataset has many more “no” than “yes” cases.
   - This makes metrics like accuracy misleading and can cause models to ignore the minority class.
   - I had to use ROC-AUC, confusion matrices and SMOTE to properly analyze performance.
2. **Runtime and GridSearch:**
   - Some models (like SVM and MLP) combined with GridSearchCV were **very slow**, especially in Google Colab.
   - I had to reduce the search space, simplify kernels (e.g., using linear SVM), and sometimes use subsets of the training data for tuning.

3. **Keeping the notebook clean and reproducible:**
    - With many models and experiments, it was easy to create messy code.
    - Using Pipelines and helper functions helped, but required careful design.
4. **Balancing depth vs. clarity:**
    - I needed to show enough experiments to make meaningful contributions without making the project too complex or confusing, especially in the presentation.

---

10. Conclusion and Future Scope

In this project, I successfully:

- **Reproduced** the methodology from a real Bank Marketing research paper using Logistic Regression and Decision Trees.
- **Confirmed** that Logistic Regression is a strong baseline for predicting term deposit subscription (ROC-AUC ≈ 0.77).
- **Analyzed SMOTE** for class imbalance and found that it slightly changes ROC-AUC but can improve minority-class recall, so it should be used carefully depending on business goals.
- **Extended** the approach to a second dataset on Bank Customer Churn and showed that the same pipeline design and models achieve extremely high performance (ROC-AUC above 0.99), with Logistic Regression again slightly best.

Future scope:

- Try more advanced models such as **Gradient Boosting** and **XGBoost**, and compare them with the current models.
- Use **feature importance** and **SHAP** values to better explain the predictions and help business stakeholders understand why a client is predicted to subscribe or churn.
- Explore **cost-sensitive learning**, where different types of errors (false positives vs. false negatives) have different costs, which is very relevant for banking decisions.

- Apply the pipeline to **additional real-world datasets** or connect it to a simple dashboard or API to support decision-makers in marketing and customer retention teams.

Overall, this project gave me hands-on experience in reading a paper, reproducing its methodology, extending it with new ideas, and presenting the results in a clear and practical way.