Unit 8: Programming Exercise

Contents

1. Introduction

I have been asked to develop a data-driven model to improve the efficiency of a bank's telemarketing campaigns. By using a random and cleaned sample of data from a Portuguese bank (https://archive.ics.uci.edu/dataset/222/bank+marketing) collected between 2008 and 2013, I will showing how machine learning can boost campaign figures by minimizing the guess work in cold calling.

2. Exploratory Data Analysis

Steps to my EDA:

1. Introduction

Exploratory Data Analysis is a crucial step in understanding the structure and characteristics of a dataset before confirmatory data analysis. (Arunkumar & Thambusamy, 2021).

I will begin by stating several questions that are hypothesis-driven to direct the analysis towards an informative conclusion. (Midway, 2022)

2. Data Understanding and Preprocessing

2.1. Variable Overview

Knowing variable types and having an overview of all variables helps in selecting appropriate visualization and statistical techniques. (Dhanshetti, 2023)

I will dive into the three groups:

- Bank client information: Columns like age, job and marital status.

- Marketing campaign: Columns like contact, month and campaign.

- Target Variable: Sale.

2.2. Data Quality Assessment

Find and decide what to do with missing, duplicate and inconsistent variables, as they can cause inaccurate findings in the statistical analyses. (Dhanshetti, 2023)

3. Univariate Analysis

Univariate analysis examines the distribution of individual variables.

I can use a univariate analysis to find outliers. Outliers are not always a bad thing. Adjustments are done if the influence of outliers look to have a chance of skewing the results. (Dhanshetti, 2023)

4. Bivariate Analysis

Identify strong and weak correlation between variables. Answer some questions I have initiated in the introduction, opening ideas to more in-depth Multivariate Analysis. Identifying trends that may indicate important predictive features. (Dhanshetti, 2023)

5. Multivariate Analysis

Multivariate analysis helps in understanding complex interactions and answer some of the more complex questions I have and attempting to find what the best columns are for plotting machine learning models. Multivariate analysis helps visualize relationships between multiple variables. (Arunkumar & Thambusamy,2021)

6. Conclusion

The EDA reveals patterns in the dataset relevant to predicting sales.

2.1 Data Understanding and Preprocessing

Some of my formulated initial questions:

1. Younger clients are buying long term bank deposits compared to older clients.
2. More previous campaigns mean higher chances of sales.
3. Non-existent poutcome has a higher chance of sales than successful poutcome.

Initially, the dataset contained 4100 observations and 21 columns. Column education was called “k”, so I renamed it and column y was renamed to “sale”. Columns "Duration" and "Default" were removed since "Duration" was not specified in the assignment outline, and "Default" had extensive missing data (“unknown” in this case). With almost 20% missing data, it made more sense to remove the column instead of removing rows with missing values.

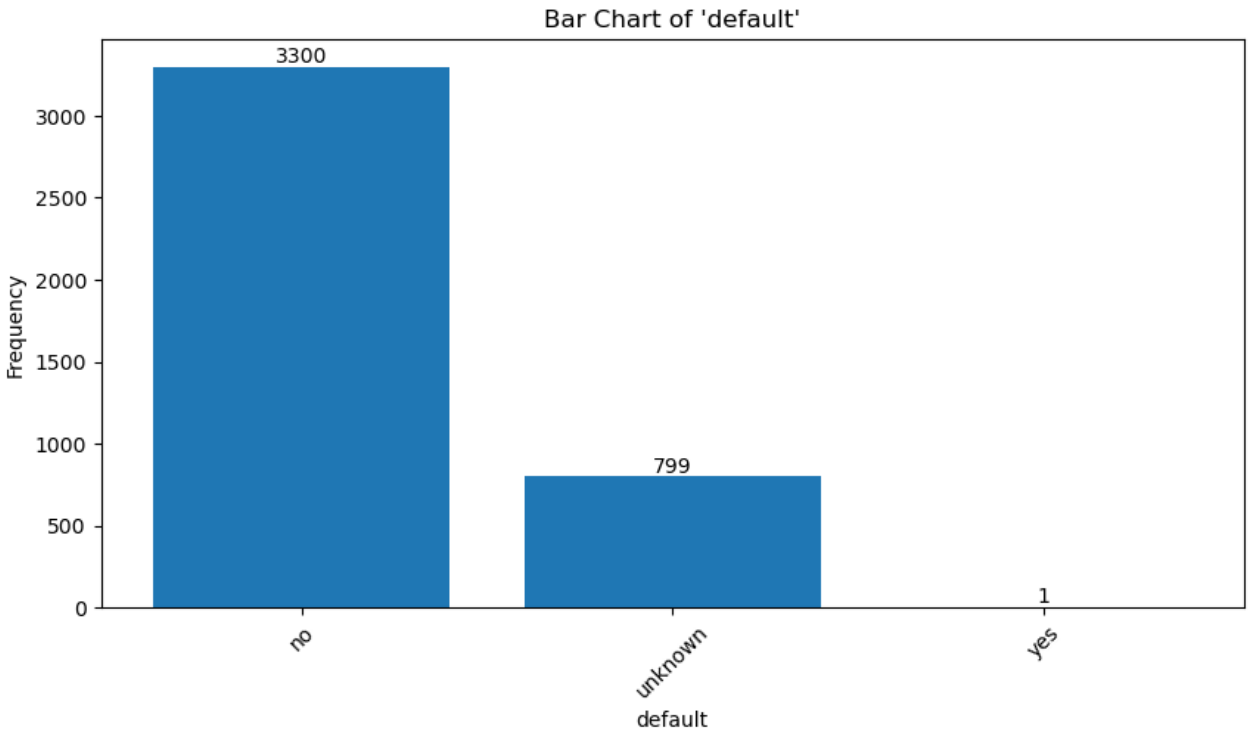

Figure 1: Bar chart of 'default' column later removed in sample

Further investigation found column campaign with significant outliers.

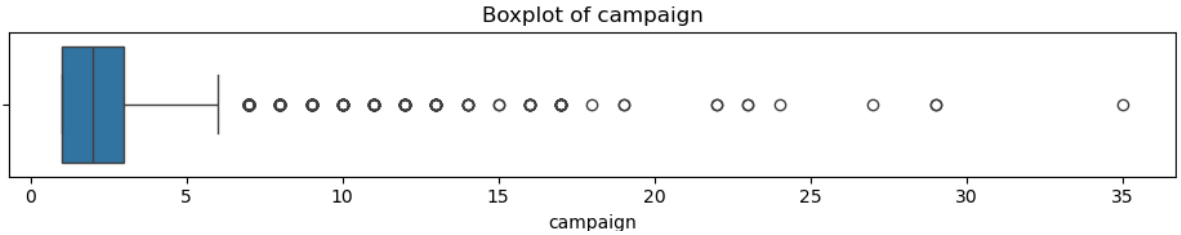

Figure 2: Boxplot of "campaign" column showing significant outliers

I found the outliers in campaign had no sales made from the 98th percentile (11). So, I set all values above 11 to 11, to lessen the outliers.

Column pdays, which represents the days passed after a client had been last contacted on a previous campaign. With rules given that 999 means a client was not previously contacted. I found 397 instances where pdays was equal to 999 and poutcome (outcome of the previous campaign for the client) was either failure or success, which makes no logical sense. A client cannot have never been contacted on a previous campaign and have a successful or failure on a previous campaign. This is clearly inaccurate data and needs to be removed. Then I changed the 999 figures to -1, to allow the box plot of pdays to have a smaller range and hence less outliers.

Mismatch: pdays 999 but poutcome not nonexistent

	age	job	marital	education	housing	loan	contact	month	day_of_week	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx
5	32	services	single	university.degree	no	no	cellular	sep	thu	3	999	2	failure	-1.1	94.199
8	31	services	divorced	professional.course	no	no	cellular	nov	tue	1	999	1	failure	-0.1	93.200
27	28	blue-collar	married	basic.6y	no	no	cellular	may	mon	2	999	1	failure	-1.8	92.893
30	38	technician	married	university.degree	yes	yes	cellular	mar	tue	1	999	1	failure	-1.8	92.843
42	76	retired	married	university.degree	no	no	cellular	aug	thu	1	999	1	failure	-1.7	94.027
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4061	57	self-employed	married	basic.4y	yes	no	telephone	apr	mon	3	999	1	failure	-1.8	93.075
4066	81	retired	married	basic.4y	yes	no	cellular	oct	wed	1	999	2	failure	-1.1	94.601
4072	32	blue-collar	married	professional.course	yes	no	cellular	may	fri	1	999	1	failure	-1.8	92.893
4074	46	admin.	married	university.degree	no	no	cellular	nov	thu	1	999	1	failure	-0.1	93.200
4089	25	admin.	single	university.degree	yes	yes	cellular	oct	fri	1	999	1	failure	-3.4	92.431

397 rows × 19 columns

Table 1: Data frame representing rows where pdays equals 999 and poutcome does not equal 'nonexistent'

This leaves the final sample to 3396 rows and 19 columns. The columns "Duration" and "Default" and 704 (17.2%) rows have been removed to improve data integrity and remove all missing values.

Summary Statistics:

	age	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	3396.000000	3396.000000	3396.000000	3396.000000	3396.000000	3396.000000	3396.000000	3396.000000	3396.000000
mean	39.888987	2.508245	-0.716431	0.072733	0.289576	93.635085	-40.278475	3.850556	5175.345230
std	10.076718	2.163887	1.550516	0.406514	1.494846	0.561766	4.403432	1.642509	69.792955
min	19.000000	1.000000	-1.000000	0.000000	-3.400000	92.201000	-50.800000	0.635000	4963.600000
25%	32.000000	1.000000	-1.000000	0.000000	-1.100000	93.200000	-42.700000	1.413750	5099.100000
50%	38.000000	2.000000	-1.000000	0.000000	1.100000	93.918000	-41.800000	4.858000	5195.800000
75%	47.000000	3.000000	-1.000000	0.000000	1.400000	93.994000	-36.400000	4.962000	5228.100000
max	88.000000	11.000000	21.000000	6.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

Table 2: Data frame representing the summary statistics of the final sample used for plotting

campaign - Average times the client gets contacted is 2.5 times. With a maximum of 11. 11 looks to be an outlier but should not be an issue with a small range of 10.

2.2 Univariate Analysis

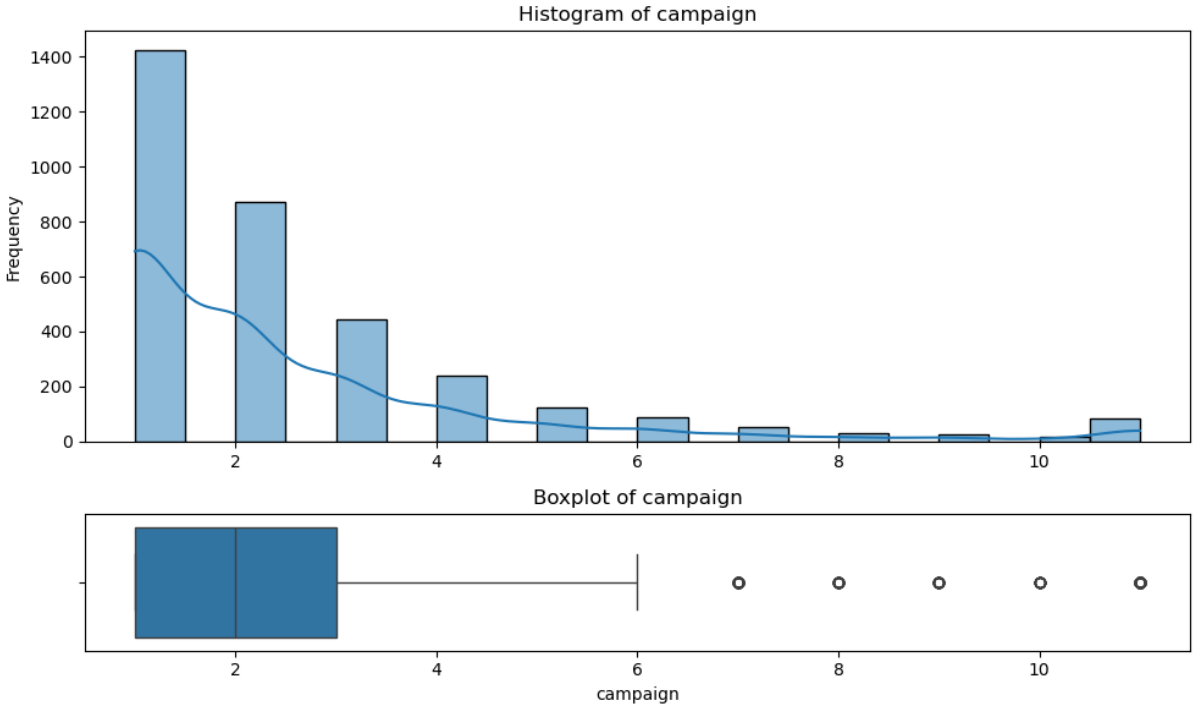

Figure 3: Histogram and boxplot of campaign

With majority of clients not being contacted in previous campaigns and also the adjustment of making 11 the maximum. The plot is a right-skewed graph which does have a slight increase on the far right. Some outliers shown but they are valuable data we require for accurate results.

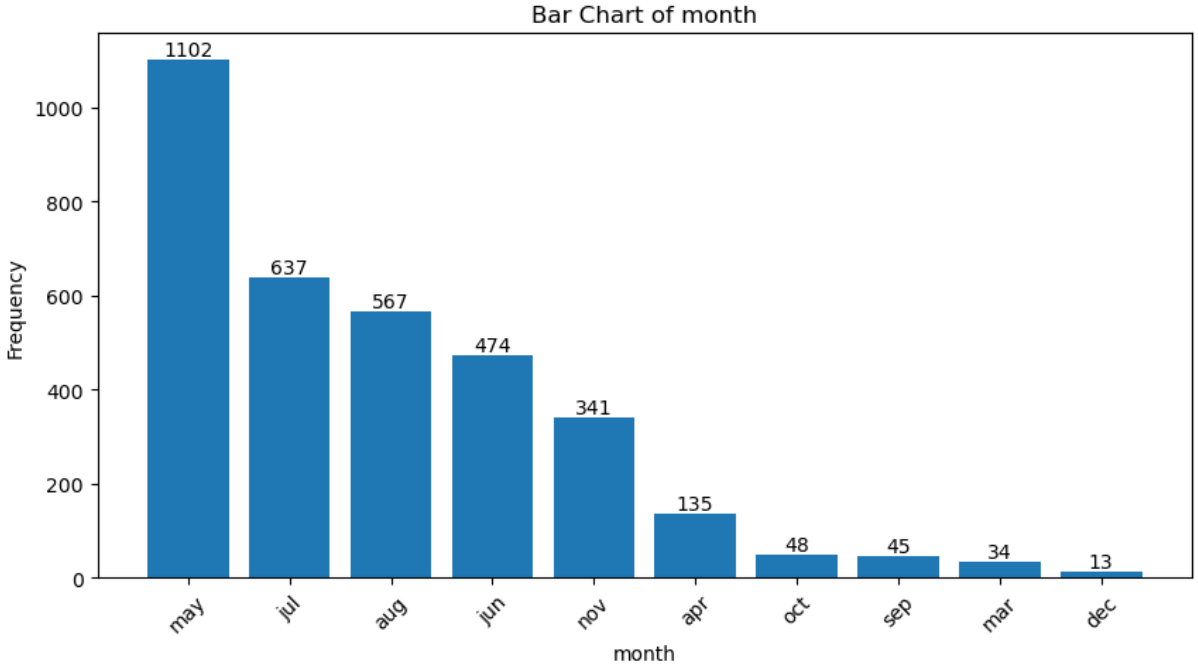

Figure 4: Bar graph of column month

In the month column, May has significantly more values with 1102 and December having the least with 13.

2.3 Bivariate Analysis

To prepare my data for the bivariate analysis, one-hot encoding was done to columns like marital and sale as they are columns with few unique values. Label encoding was done to columns like education and job as they had significant unique values. The encoding choices align with the findings of the paper (Poslavskaya & Korolev, 2023) which found that one-hot encoding performs well with binary columns or columns with low cardinality and label encoding is used for columns with high cardinality.

	age	job	marital_married	marital_single	marital_divorced	education	housing_yes	housing_no	loan_no	loan_yes	contact_cellular	contact_telephone	month
0	30	7	1	0	0	4	1	0	1	0	1	0	5
1	39	4	0	1	0	5	0	1	1	0	0	1	5
2	25	4	1	0	0	5	1	0	1	0	0	1	6
4	47	5	1	0	0	7	1	0	1	0	1	0	11
6	32	5	0	1	0	7	1	0	1	0	1	0	9
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4095	36	5	0	1	0	7	0	1	0	1	1	0	8
4096	33	4	1	0	0	5	0	1	1	0	0	1	5
4097	41	7	0	0	1	4	0	1	1	0	1	0	8
4098	34	3	0	1	0	7	1	0	1	0	1	0	8
4099	58	5	0	0	1	5	0	1	1	0	1	0	8

Table 3: Data frame representing the encoded sample

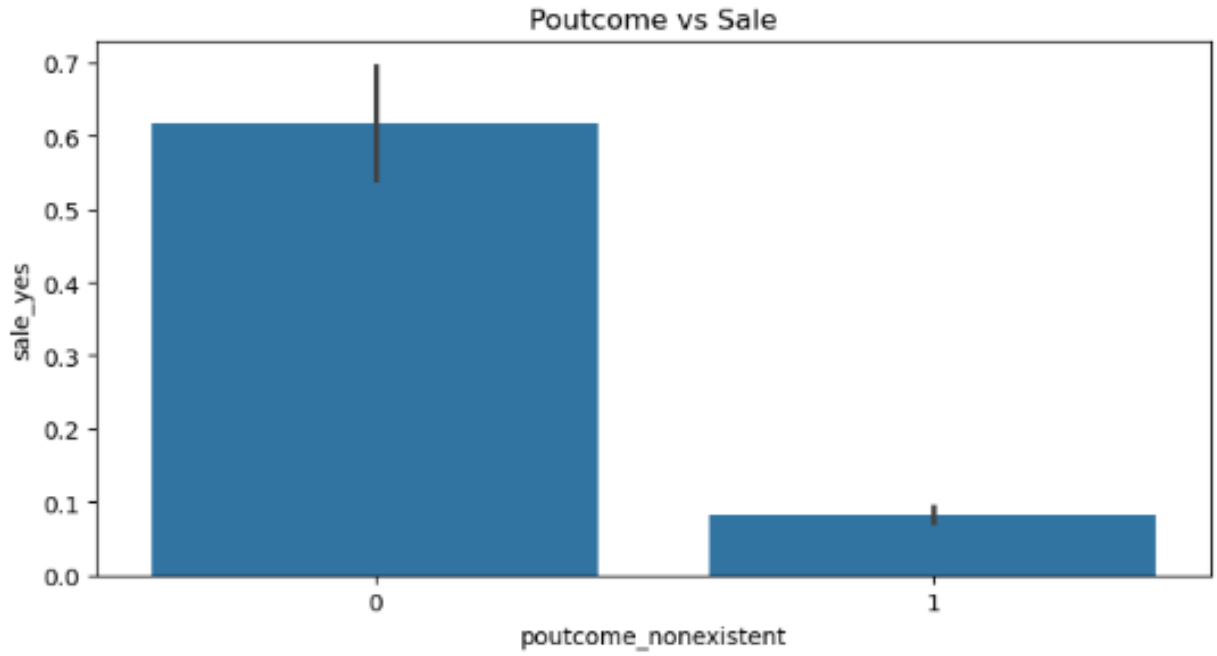

Figure 5: Bar graph comparing poutcome to sales

The encoding allows for findings like figure 5. Where I attempted to answer hypothesis 3, which was proved incorrect. As weather a client has not been successful or not in a previous campaign, the chance of a sale is 50% higher compared to a client who has never been contacted before.

I did not want to miss any other columns with potential correlation with a sale, so I performed a Chi-squared test and a Mann-Whitney U test. Where the reliability of the Chi-Square test is not significantly different between a 95% and 99% confidence interval, making it a surefire tool for hypothesis testing. (Falana et al, 2024) and the Mann-Whitney U Test ranks all variables, the test can determine the better-performing variable even when distributions differ significantly. (Price et al, 2022). With this, I was able to put together a list of strong correlation columns.

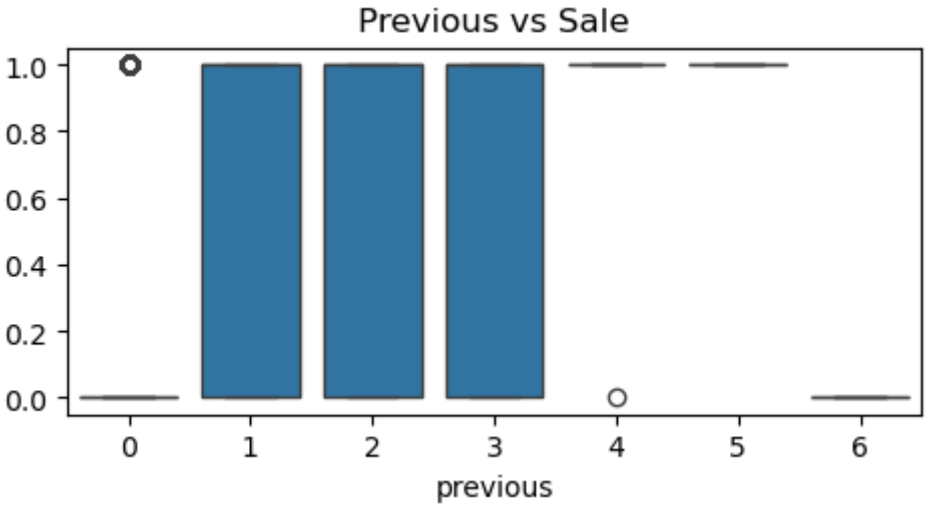

Figure 6: Bar graph comparing previous to sales

Having a p value < 0.0001, suggesting a strong relationship, plotting shows having previous campaigns for a client has a positive impact on sales. With 1-5 previous having a higher chance of sales than 0, but 6 having the least chance of sales.

3. Statistical Modelling

Model Selection and Feature Strategy

Logistic Regression was selected due to its simplicity and statistical grounding. It has been consistently shown to provide high precision and recall in structured datasets with interpretable coefficients. (Lin et al, 2024)

Generalized Linear Models (GLM) with a logit link were included as a generalization of logistic regression. They allow for flexibility in modelling the relationship between predictors and response variables. (Bar-Lev et al, 2024)

Decision Trees are capable of capturing nonlinear interactions and hierarchical feature importance. (Saroj & Anand, 2021). Their interpretability and visual clarity are beneficial in business use cases. (Dou and Meng, 2023)

Four feature sets were used:

- **Bank Client Information columns**

- **Marketing Campaign Information columns**

- **All columns**

- **Strong Correlation columns**

Standardization was applied to input features before Logistic Regression to address convergence issues and numerical instability. A maximum iteration count of 1000 ensured better convergence, following recommendations from Kumle et al. (2021) for iterative GLM estimation stability.

GLMs were fitted using the library statsmodels with a Binomial family and Logit link, following best practices for binary response models in structured data (Schwendinger et al, 2024).

Decision Trees were constrained to a depth of 5 to prevent overfitting while capturing meaningful feature interactions, a method supported by Lin et al. (2024).

Model Performance

	Type	Feature Set	Model	ACC	AUC
6		Marketing Info	GLM	0.908127	0.797709
7		Marketing Info	Logistic	0.908127	0.795769
0		All Features	GLM	0.909305	0.792549

Table 4: Data frame representing top 3 models

These findings mirror Saroj and Anand's (2021) results, where logistic regression outperformed decision trees. Marketing features were best on all models compared to other tests besides for decision tree on strong correlation. Confusion matrices reinforced this: bank client features led to zero true positives for both logistic and GLM models, whereas marketing features enabled models to recall up to 24–29% of positive cases.

This reinforces progress with GLM and marketing info. Optimizing the model to develop a statistical model that can accurately predict the result of a phone call to sell long term bank deposits.

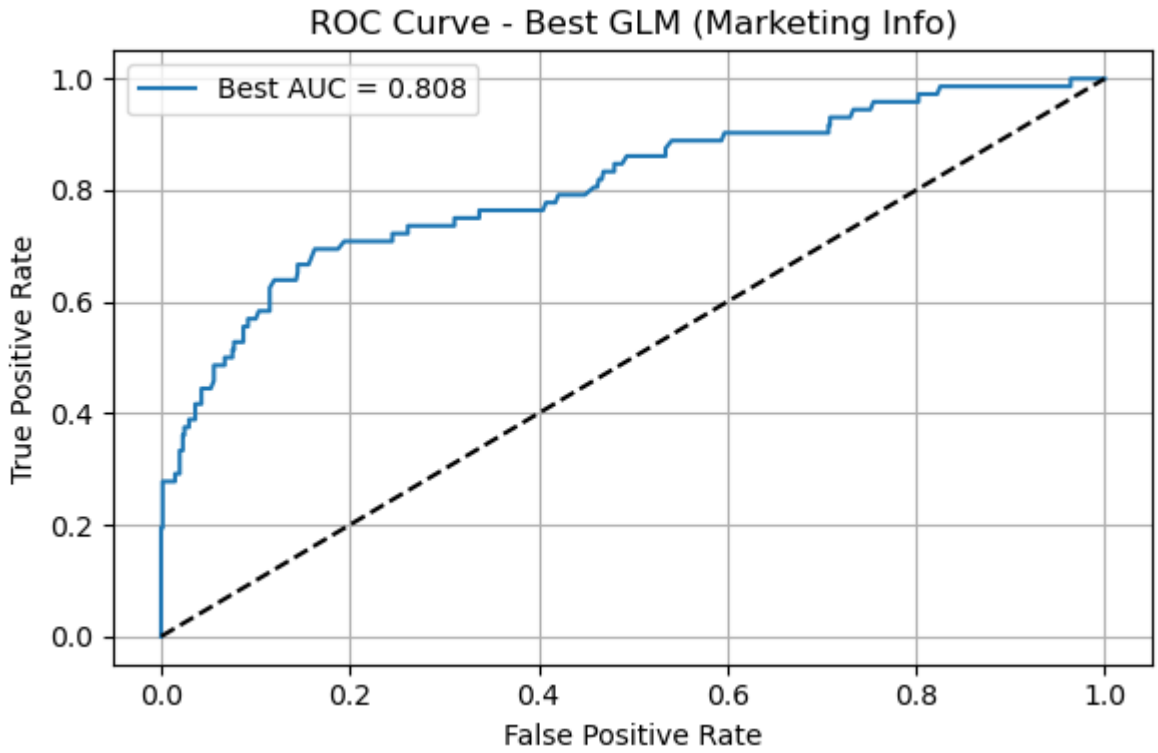

Figure 7: ROC curve of the best model

```python
# Load and prep
df = df_encoded.copy()
y = df['sale_yes']
df = df.drop(columns=['sale_no'])

marketing_cols = ['contact_cellular', 'contact_telephone', 'month', 'day_of_week', 'campaign',
                  'pdays', 'previous', 'poutcome_nonexistent', 'poutcome_success', 'poutcome_failure',
                  'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed']
X = df[marketing_cols]

# Split once
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2, random_state=42)

# Results store
results_list = []

# Loop over thresholds and balancing options
for balance in [True, False]:
    for threshold in np.arange(0.2, 1.5, 0.025):
        X_train_bal, y_train_bal = X_train.copy(), y_train.copy()

        if balance:
            smote = SMOTE(random_state=42)
            X_train_bal, y_train_bal = smote.fit_resample(X_train, y_train)

        # Scale
        scaler = StandardScaler()
        X_train_scaled = scaler.fit_transform(X_train_bal)
        X_test_scaled = scaler.transform(X_test)

        X_train_scaled = sm.add_constant(X_train_scaled)
        X_test_scaled = sm.add_constant(X_test_scaled)

        # Train GLM
        glm = sm.GLM(y_train_bal, X_train_scaled, family=sm.families.Binomial(link=sm.families.links.Logit()))
        glm_result = glm.fit()

        # Predict
        glm_probs = glm_result.predict(X_test_scaled)
        glm_preds = (glm_probs > threshold).astype(int)

        # Metrics
        auc = roc_auc_score(y_test, glm_probs)
        acc = accuracy_score(y_test, glm_preds)
        precision, recall, f1, _ = precision_recall_fscore_support(y_test, glm_preds, average='binary', zero_division=0)

        results_list.append({
            'Balance': balance,
            'Threshold': round(threshold, 2),
            'AUC': auc,
            'Accuracy': acc,
            'F1': f1,
            'Precision': precision,
            'Recall': recall,
            'Model': glm_result,
            'Scaler': scaler
        })
```

Figure 8: My code that creates the marketing column and loops through the thresholds

Testing various thresholds and found that threshold 0.92 had the best result. The best model was built using a GLM with a logit link, making it light weight and quick to deliver results for a fast-paced banking environment.

4. Conclusion

The model provides high accuracy and precision meaning when it says the client will be a sale, it is very likely to be correct.

The learning rate of the model (AUC) at 80% shows the model learns quicky. Which is required for real time models. High accuracy and precision is required so the model makes the right decisions, leading to an increase and efficiency in sales, the final models is on 92% and 94% respectively. The low recall score of 24% means the model does not catch all potential sales, but that is sacrificed for high precision. Increasing revenue on long term bank deposit sales, by providing efficiency and accuracy to any telemarketing team. As per Dou & Meng (2023), such prioritization strategies are crucial in resource-constrained business environments, where precision is often more valuable than coverage.

1607 words

References

Midway, S. (2022) Data Analysis in R. Avaliable from: https://bookdown.org/steve_midway/DAR/exploratory-data-analysis.html [Accessed 14 March 2025]

Dhanshetti, P. (2023) Techniques of Exploratory Data Analysis. Avaliable from: https://www.researchgate.net/publication/374674185_Techniques_of_Exploratory_Data_Analysis [Accessed 14 March 2025]

Arunkumar, R., Thambusamy, V. (2021) An Exploratory Data Analysis Process on Groundwater Quality Data. Avaliable from: https://www.researchgate.net/publication/348351115_An_Exploratory_Data_Analysis_Process_on_Groundwater_Quality_Data [Accessed 15 March 2025]

Theportugalnews.com. (2024). Retirement age to rise in 2026. [online] Available at: https://www.theportugalnews.com/news/2024-12-30/retirement-age-to-rise-in-2026/94548 [Accessed 18 March 2025]

Poslavskaya, E., Korolev, A. (2023). Encoding categorical data: Is there yet anything 'hotter' than one-hot encoding?. Avaliable at: https://arxiv.org/abs/2312.16930 [Accessed 20 March 2025]

Falana, S. A., Omaku, P., Familokun, O. T., & Musa, A. . S. (2024). Reliability of Chi-Squared Hypotheses Tests Conducted at 95% Confidence Level: Reliability of Chi-Squared Hypotheses Tests. Nasara Journal of Science and Technology, 11(1), 13–24. Avaliable from: https://nasarajournal.com.ng/index.php/NJST/article/view/64 [Accessed 20 March 2025]

Price, K., Kumar, A., Suganthan. P. (2022) Trial-Based Dominance Enables Non-Parametric Tests to Compare both the Speed and Accuracy of Stochastic Optimizers. Cornell University. Avaliable from: https://arxiv.org/abs/2212.09423 [Accessed 20 March 2025]

Lin, S.-J., Liu, C.-C., Tsai, D.M.T., Shih, Y.-H., Lin, C.-L. and Hsu, Y.-C. (2024). Prediction Models Using Decision Tree and Logistic Regression Method for Predicting Hospital Revisits in Peritoneal Dialysis Patients. Avaliable from: https://www.mdpi.com/2075-4418/14/6/620 [Accessed 21 March 2025]

Bar-Lev, S.K., Liu, X., Ridder, A. and Xiang, Z. (2024). Generalized Linear Model (GLM) Applications for the Exponential Dispersion Model Generated by the Landau Distribution. Avaliable at: https://www.mdpi.com/2227-7390/12/13/2021 [Accessed 21 March 2025]

Saroj, R.K. and Anand, M. (2021). Environmental factors prediction in preterm birth using comparison between logistic regression and decision tree methods: An exploratory analysis. Avaliable at: https://www.sciencedirect.com/science/article/pii/S2590291121001121 [Accessed 21 March 2025]

Dou, Y. and Meng, W. (2023). Comparative analysis of weka-based classification algorithms on medical diagnosis datasets. Avaliable at: https://journals.sagepub.com/doi/full/10.3233/THC-236034 [Accessed 21 March 2025]

Kumle, L., Võ, M.L.-H. and Draschkow, D. (2021). Estimating power in (generalized) linear mixed models: An open introduction and tutorial in R. Avaliable at: https://link.springer.com/article/10.3758/s13428-021-01546-0 [Accessed 21 March 2025]

Schwendinger, B., Schwendinger, F. and Vana, L. (2024). Holistic Generalized Linear Models. Avaliable at: https://www.jstatsoft.org/article/view/v108i07 [Accessed 21 March 2025]

Appendix

- All coding, analysis and notes are found in the accompanying code document Unit 8 - Programming Exercise.