Berliner Hochschule für Technik (BHT)

Course Name: Data Science Platform

Project Title: Predict Term Deposit Using Dataiku

Submission Date: 18 January 2024

Submitted By

Ahmed Dider Rahat	Tania Sultana
ID: 916146	ID: 931031

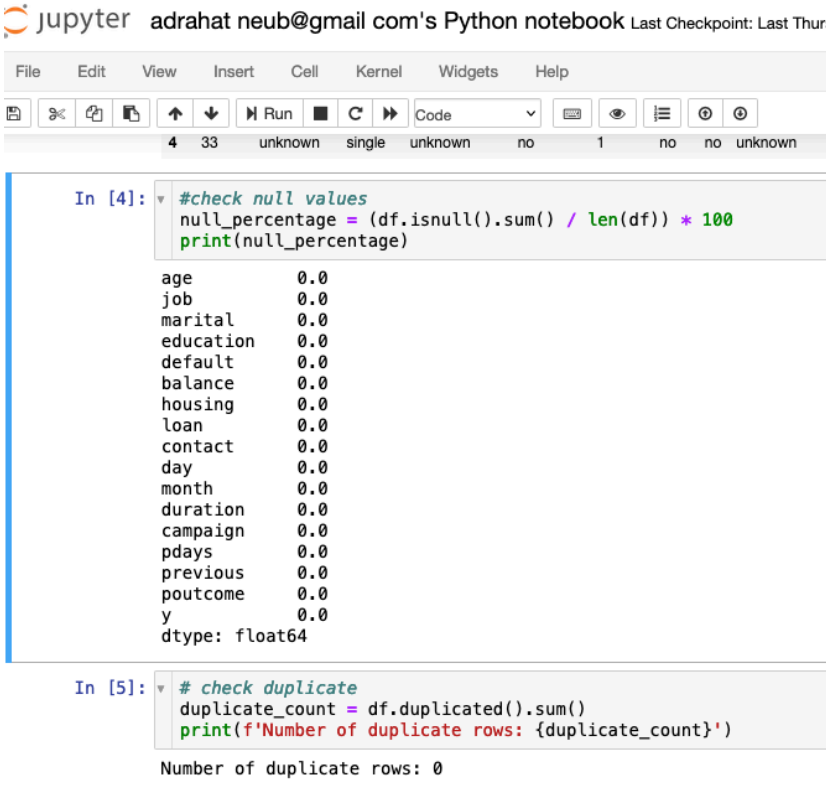
Abstract

This project, "Predict Term Deposit Using Dataiku," aimed to utilize Dataiku's data science platform to predict whether clients of a banking institution would subscribe to a term deposit. It involved univariate analysis of individual features to understand their impact and the creation of a structured workflow for data processing and model evaluation. The dataset was split into training (70%) and testing (30%) sets to train and assess three different machine learning models: Logistic Regression, Support Vector Machine (SVM), and Random Forest. Model performance was evaluated using metrics like Accuracy, Precision, Recall, F1-Score, and ROC AUC both on training and test data.

Introduction of Data

This dataset is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (variable y).

1. Check Null values and Duplicate Entries


jupyter
adrahat neub@gmail com's Python notebook Last Checkpoint: Last Thur
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
4   33   unknown   single   unknown   no   1   no   no   unknown
```

In [4]:
```
#check null values
null_percentage = (df.isnull().sum() / len(df)) * 100
print(null_percentage)
```

```
age          0.0
job          0.0
marital      0.0
education    0.0
default      0.0
balance      0.0
housing      0.0
loan         0.0
contact      0.0
day          0.0
month        0.0
duration     0.0
campaign     0.0
pdays        0.0
previous     0.0
poutcome     0.0
y            0.0
dtype: float64
```

In [5]:
```
# check duplicate
duplicate_count = df.duplicated().sum()
print(f'Number of duplicate rows: {duplicate_count}')
```

```
Number of duplicate rows: 0
```

Fig-01: Null and duplicate entry analysis

2. Univariate Analysis

1. **Variable-Age:** The age of the client is distributed from 18 to 95 and the mean is 40.94. The most frequent age group is (30-40) and almost 40% of the total data is dense in this age group.

2. **Variable-Job:** The type of job of the client. The largest segments are blue-collar workers (21.53%), management professionals (20.92%), and technicians (16.8%). Smaller proportions of the dataset include retirees, students, and unemployed individuals, reflecting the campaigns' broader demographic reach. The presence of an 'unknown' category at 0.64% suggests some gaps in the job information collected.

3. **Variable-Marital:** The marital status of the client. Among them 60% of the clients are married, 28% are single, and 12% are divorced.

4. **Variable-Education:** The level of education of the client. The majority of clients, 51%, have a secondary level of education, followed by 29% with tertiary education, 15% with primary education, and 4% with an unknown level of education.

5. **Variable-Default:** Indicates whether the client has credit in default. This information reflects the credit default status of clients. The majority, 98%, do not have credit in default, while 2% have defaulted on their credit.

6. **Variable-Balance:** The mean annual balance is approximately €1362.27, with a median value of €448. The dataset demonstrates variability, underscored by a standard deviation of €3044.77.

7. **Variable-Housing:** This dataset provides insights into the distribution of housing loans among the client demographic. Notably, a significant majority, accounting for approximately 56% of clients, holds housing loans, while 44% of clients are without such financial obligations.

8. **Variable-Loan:** This data indicates the prevalence of personal loans among clients. A substantial majority, approximately 84%, do not have a personal loan, while 16% of clients have opted for a personal loan.

9. **Variable-Contact:** This data outlines the communication types used during the campaign. Notably, 65% of contacts were made through cellular communication, 29% were categorized as unknown, and 6% involved telephone communication.

10. **Variable-Day:** The last contact day of the month. The 20th day of the month contains most of the last call, which holds 6% of the total calls.

11. **Month:** This data illustrates the distribution of the last contact month of the year in the dataset. Notably, May has the highest frequency at 30%, followed by July at 15%, and August at 14%.

12. **Variable-Duration:** The duration of the last contact, in seconds. Most of the calls last between 3-5 minutes (180-300 seconds). Yet, the duration is not known before a call is performed.

13. **Variable-Campaign:** On average, 2.76 contacts were performed, with a median of 2. The dataset displays variability, as evidenced by a standard deviation of 3.10. The range of contacts spans from a minimum of 1 to a maximum of 63.

14. **Variable-Previous:** This dataset outlines the frequency of contacts made before the current campaign for individual clients. The predominant segment, comprising approximately 82%, had no prior contacts. About 6% had one prior contact, 5% had two prior contacts, and 3% had three prior contacts.

15. **Variable-Poutcome:** This dataset provides insights into the outcomes of the previous marketing campaign. Notably, the majority, accounting for approximately 82%, had an unknown outcome. Failures constituted 11% of cases, while 4% were categorized as "other" outcomes. Successful outcomes were observed in 3% of cases.

16. **Predicted Variable-(y):** This data represents the target variable indicating whether clients subscribed to a term deposit. The majority, approximately 88%, did not subscribe, while 12% did subscribe.

Machine Learning Workflow

The machine learning workflow is divided into some core tasks. Starting from data preparation to model evaluation, we perform several subtasks. The complete workflow is given below:

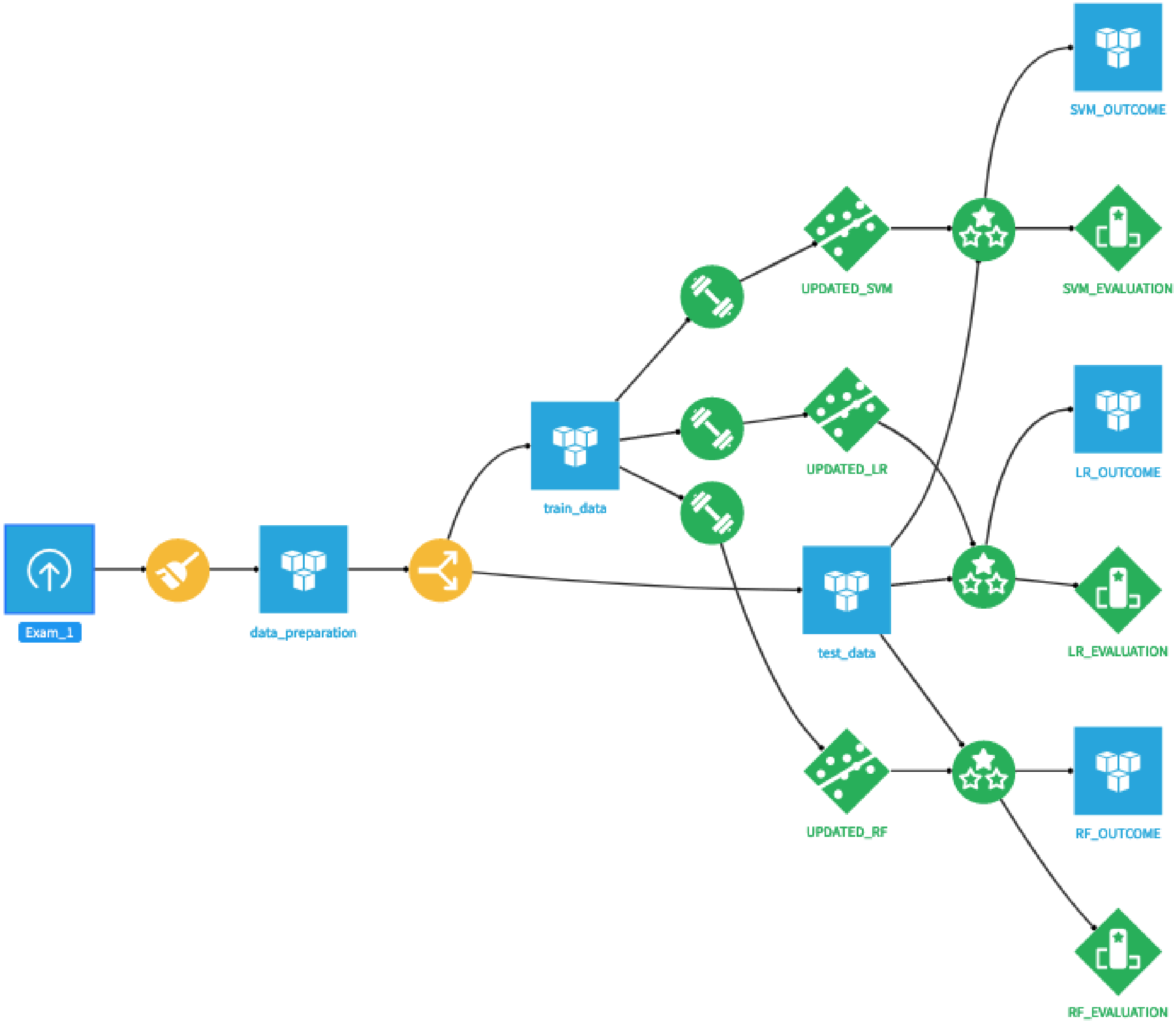


Fig-02: Machine Learning Workflow

Step 1: Data Preparation

The data preparation steps of the project:

1. As our data do not contain any null entries or duplicate entries, we don't need to clean the data.
2. For our better understanding, we rename some of the columns: y → predicted_value, marital → marital_status, default → credit_default.
3. Remove the duration from the model, as we don't know the duration before the call.

Step 2: Train/Test Split

For our machine learning training and evaluation, we split the dataset into two subsets: train data (70%), and test data (30%). The train data is used for model training and hyper-parameter tuning, and test data is only used to evaluate the model's performance. The training set contains 31,648 rows whereas the test set contains 13,563 rows. We randomly partition the data but still, the proportion of predicted values remains similar.

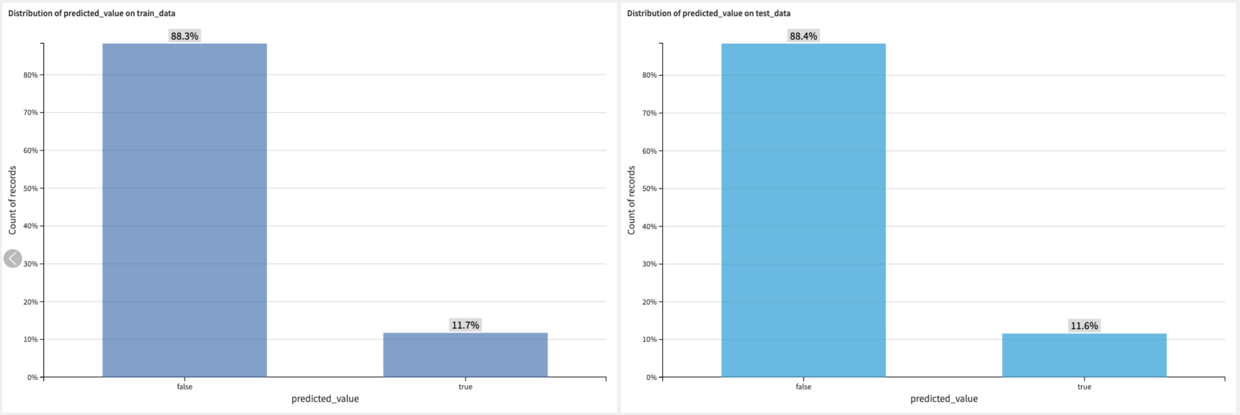


Fig-03: Comparison of each class in training and test set

Step 3: Machine Training

We have chosen three machine-learning models for our project.

1. Logistics Regression.
2. Support Vector Machine.
3. Random Forest.

Model-1: Logistic Regression

Logistic regression is a statistical method for predicting binary outcomes by using a logistic function to model a binary dependent variable based on one or more independent variables.

Feature Importance

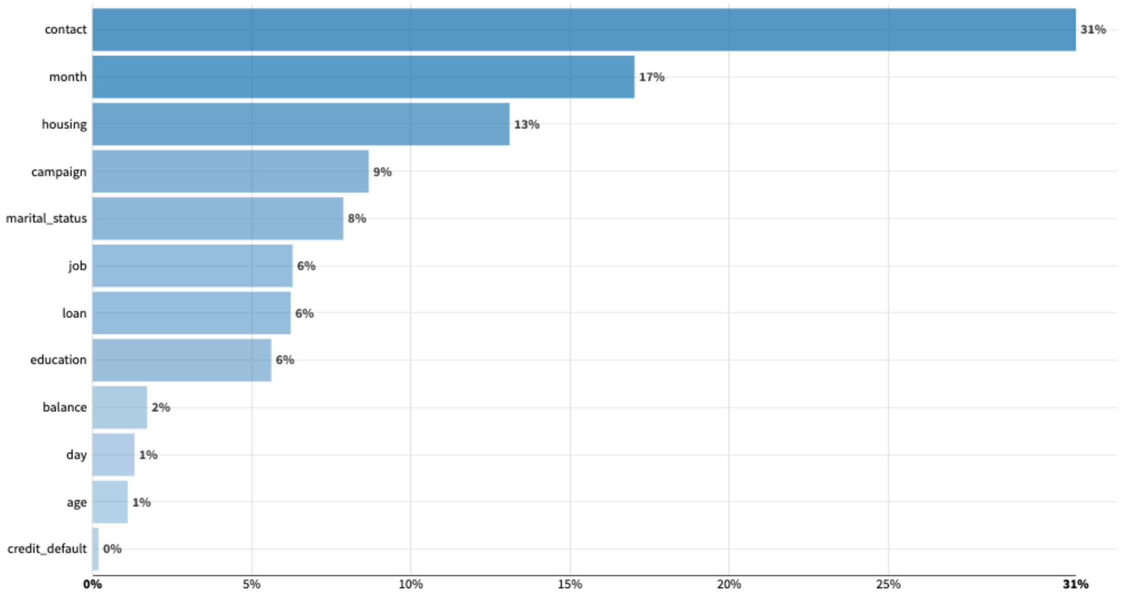


Fig-04: Feature Importance in Logistic Regression

Hyperparameter Optimization

We use k-fold (5-fold) cross-validation to optimize the hyperparameter C. In logistic regression, the hyperparameter C represents the inverse of regularization strength.

A smaller value of C will increase the regularization strength, which will create simpler models but can potentially underfit the data. It does this by penalizing the sum of squared coefficients, making them smaller. A larger value of C means less regularization, allowing the coefficients to be larger, which can capture more complexity in the data but may also lead to overfitting.

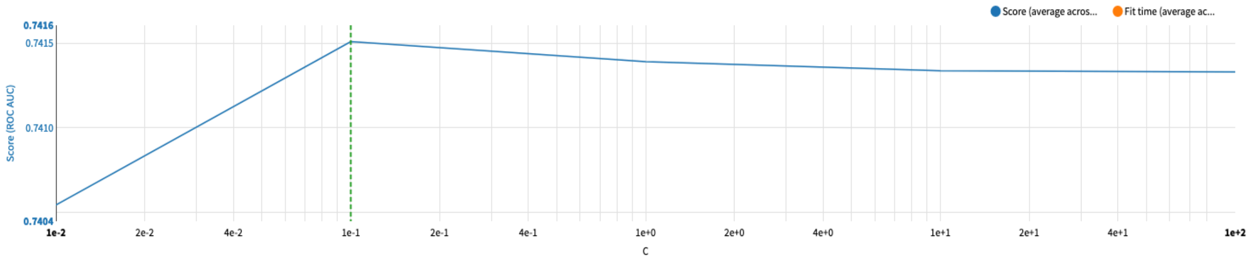
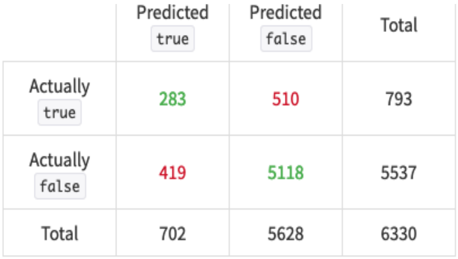


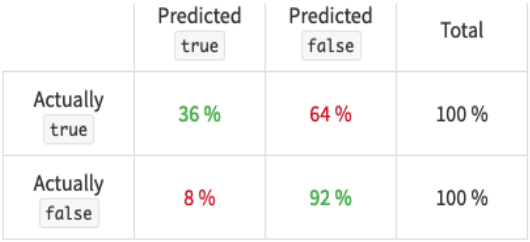
Fig-05: Value of C Vs. ROC AUC score

Results of the Training set

We got the ROC AUC = 0.742 for the optimal model.



	Predicted true	Predicted false	Total
Actually true	283	510	793
Actually false	419	5118	5537
Total	702	5628	6330



	Predicted true	Predicted false	Total
Actually true	36 %	64 %	100 %
Actually false	8 %	92 %	100 %

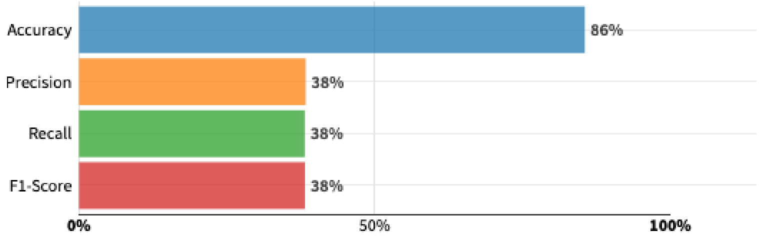


Fig-06: Training set results of Logistic Regression

Summary:

1. The accuracy of the true class is 36% and the false class is 92%.
2. As the false class contains more value, the total accuracy of the model becomes 86%.
3. The recall is 38%, which means the model correctly identifies 38% of all actual positives.
4. The value of precision refers to the model identifying only 38% of the correct true level whereas the rest 62% is predicted as true but not “actual” true.

Model-2: Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression tasks. It finds the best boundary (hyperplane) that separates classes of data in a high-dimensional space, intending to maximize the margin between the data points of different classes.

Feature importance

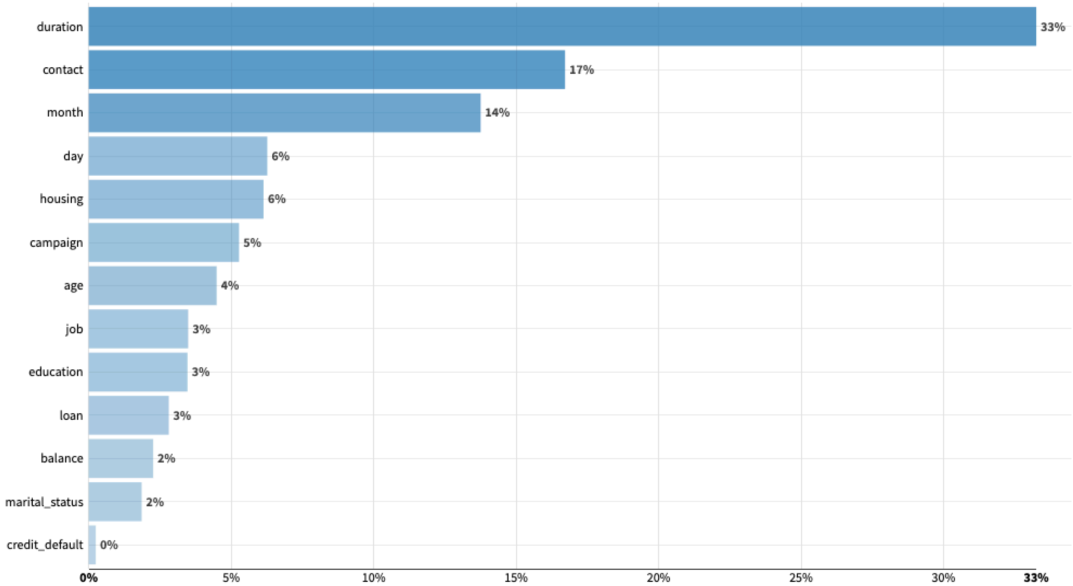


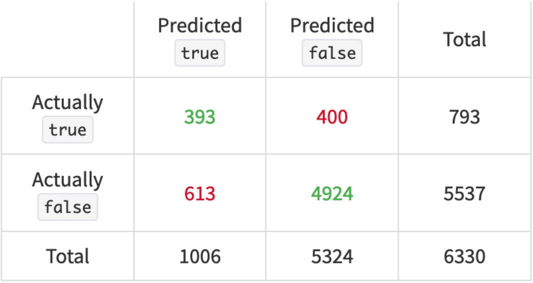
Fig-07: Feature importance in Support Vector Machine

Model Information

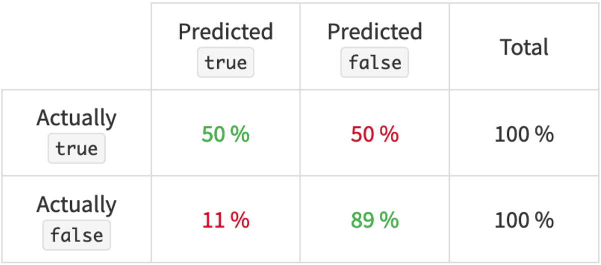
The algorithm uses an SVM classifier, the value of C is 1, and the stopping tolerance is 0.001.

Results of Training set

We got the ROC AUC = 0.765 for the optimal model.



	Predicted `true`	Predicted `false`	Total
Actually `true`	393	400	793
Actually `false`	613	4924	5537
Total	1006	5324	6330



	Predicted `true`	Predicted `false`	Total
Actually `true`	50 %	50 %	100 %
Actually `false`	11 %	89 %	100 %

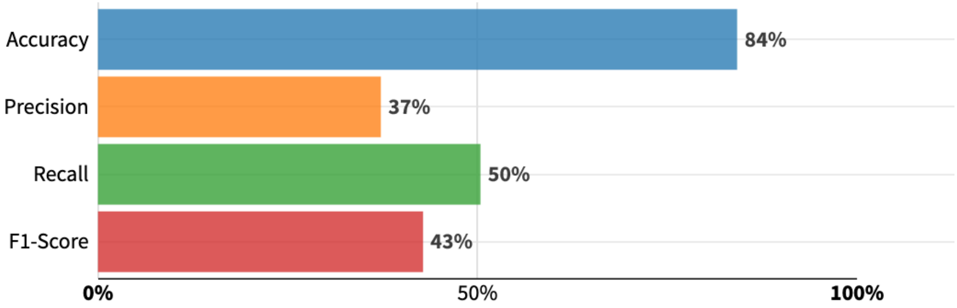


Fig-08: Training set results of Support Vector Machine

Summary:

1. The accuracy of the true class is 50% and the false class is 89%.
2. As the false class contains more value, the total accuracy of the model becomes 84%.
3. The recall is 50%, suggesting that the model correctly identifies 50% of all actual positives.
4. The value of precision refers to the model identifying only 37% of the correct true level where the remaining 43% is predicted as true but not “actual” true.

Model-3: Random Forest

Random Forest is an ensemble machine learning algorithm that builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Feature importance

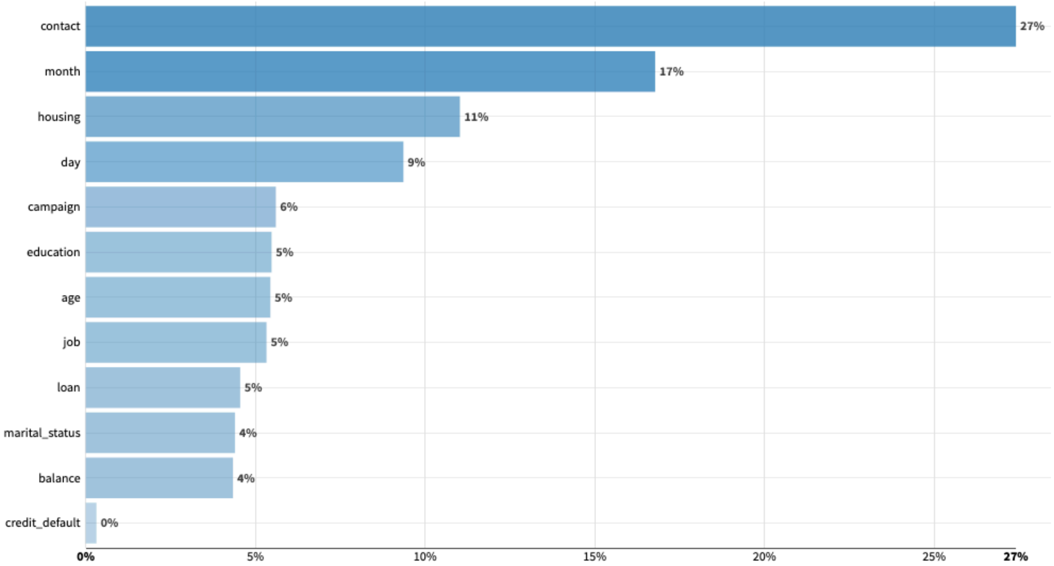


Fig-09: Feature importance in Random Forest

Hyperparameter Optimization

We use k-fold (5-fold) cross-validation to optimize the hyperparameter max_depth. There is a positive trend indicating that as the max_depth increases, the ROC AUC score also increases. This suggests that allowing the trees to grow deeper (up to a certain point) helps the model to better capture the patterns in the data. The plot has a vertical dashed line at max_depth 14, which is the optimal value found by the grid search in terms of ROC AUC score.

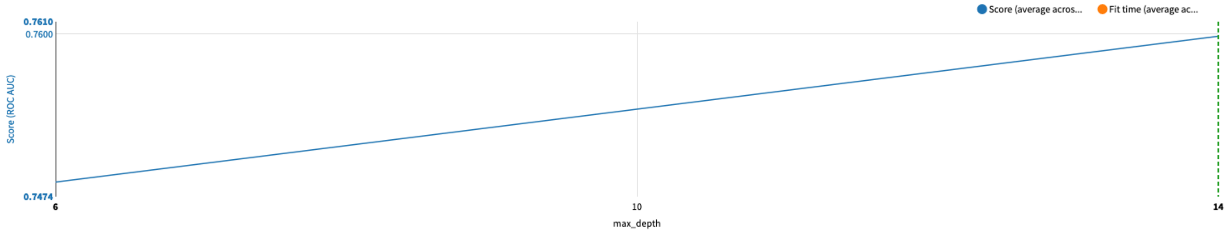
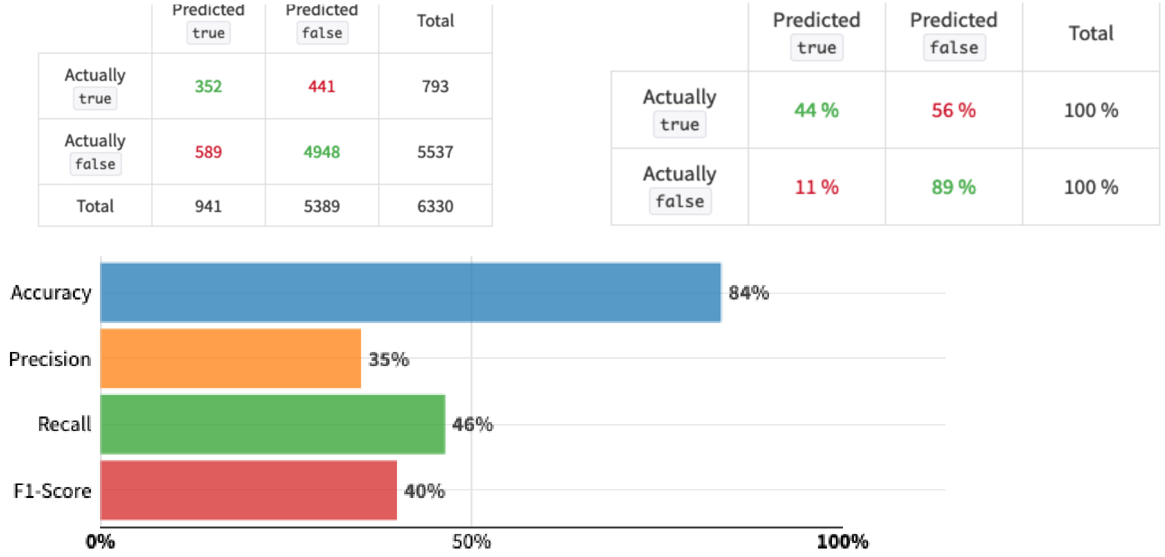


Fig-10: Value of max_depth Vs. ROC AUC score

Results of the Training set

We got the ROC AUC = 0.761 for the optimal model (max_depth = 14).



	Predicted true	Predicted false	Total
Actually true	352	441	793
Actually false	589	4948	5537
Total	941	5389	6330

	Predicted true	Predicted false	Total
Actually true	44 %	56 %	100 %
Actually false	11 %	89 %	100 %

Fig-11: Training set results of random forest

Summary:

1. The accuracy of the true class is 44% and the false class is 89%.
2. As the false class contains more value, the total accuracy of the model becomes 84%.
3. The recall is 46%, suggesting that the model correctly identifies 46% of all actual positives.
4. The value of precision refers to the model identifying only 35% of the correct true level whereas the rest 65% is predicted as true but not “actual” true.

Model Comparison

Model summary of training data:

	Logistic Regression	Support Vector Machine	Random Forest
Accuracy	**86%**	84%	84%
Precision	**38%**	37%	35%
Recall	38%	**50%**	46%
F1-Score	38%	**43%**	40%
ROC AUC	74.2%	**76.5%**	76.1%

From the above training summary table, we can conclude that the **accuracy** and **precision** of **Logistic Regression** are highest where the **Recall**, **F1-Score,** and **ROC AUC score** are best for the **Support Vector Machine**.

Step 4: Result Analysis

Model summary of test data:

	Logistic Regression	Support Vector Machine	Random Forest
Accuracy	**86%**	84.8%	83.5%
Precision	**39.4%**	38.5%	35%
Recall	39.4%	**52.3%**	49.1%
F1-Score	39.4%	**44.3%**	40.9%
ROC AUC	76.1%	**78%**	**78%**

Like the training dataset, the best **accuracy** and **precision** results were found using **Logistic Regression**. On the other hand, the **Recall** and **F-1 score** is highest in **SVM**. Lastly, the **ROC AUC** value is maximum for both **SVM** and **Random Forest**.

Conclusion

The comparative analysis of the three models indicated that the Support Vector Machine (SVM) displayed a notable improvement with the highest ROC AUC of 78% on test data, suggesting its strong predictive power. Although Logistic Regression maintained consistent accuracy, the precision and recall for SVM and Random Forest indicated a more balanced trade-off between identifying positive cases and the accuracy of those identifications.