Mini Project 1

Samuel.J 2001177

Åbo Akademi

Machine Learning 2024

4.4.2024

Mini Project 1

In this project I predicted the result for a dataset about different marketing campaigns for a banking institute using machine learning. I used Google collab as my IDE (“Google Colab”, n.d.), Python as the main programming language (“Python”, n.d.), Sickit learn as the machine learning algorithm library (“Scikit Learn”, n.d.), Pandas as the library for editing and cleaning the data (“Pandas”), and Matplotlib as the library for plotting the data (“Matplotlib”). I used two different methods(Algorithms) for the training of my machines, logistic regression and random forest classifier.

Data processing

In this machine learning project, I focused on several data processing steps to prepare the bank-additional-full.csv dataset for analysis:

Categorical Data Cleaning: I standardized categories in multiple columns like job, marital, and education against predefined lists, marking any outliers as 'unknown'.

Numeric Conversion: I converted columns that should be numeric (campaign, pdays, etc.) into numeric types, coercing errors to NaN to gracefully handle non-numeric inputs.

Output Variable Transformation: I transformed the output variable y from a yes/no format to a boolean format, making it more straightforward to use in binary classification models.

Missing Values Check: I conducted a comprehensive check for missing values to identify and plan for any missing data in the dataset.

Data Type Verification: Finally, I verified the data types of all columns to ensure each was correctly typed, preventing potential issues in data manipulation and model training. These steps were essential to ensure the dataset was clean, consistent, and fully prepared for the next stages of analysis and modeling, laying the foundation for accurate predictive modeling.

Modeling

In my machine learning project, I focused on Logistic Regression and Random Forest Classifier due to their suitability for binary classification problems.

Logistic Regression:

Was my first choice for its simplicity and interpretability. I prepared the data through a pipeline that included imputation, scaling, and encoding, ensuring it was ready for model training. This model served as a good baseline with decent accuracy.

Random Forest Classifier:

Was chosen next for its robustness and ability to handle complex patterns without overfitting. Employing a similar preprocessing approach, this model showed an improvement in accuracy over Logistic Regression.

Improving Performance:

I enhanced model performance by fine-tuning preprocessing steps and hyperparameters. Adjustments in hyperparameters, such as max_iter for Logistic Regression and n_estimators for Random Forest, alongside meticulous data preprocessing, were key strategies.

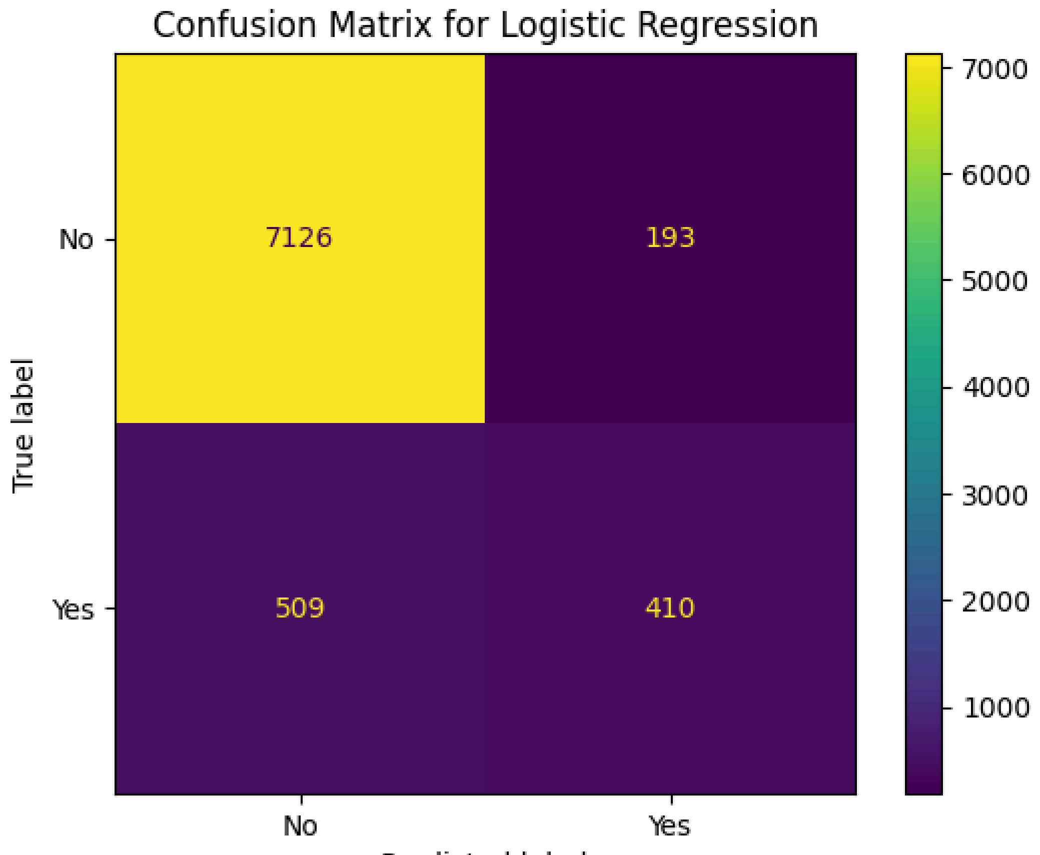
Final Accuracy:

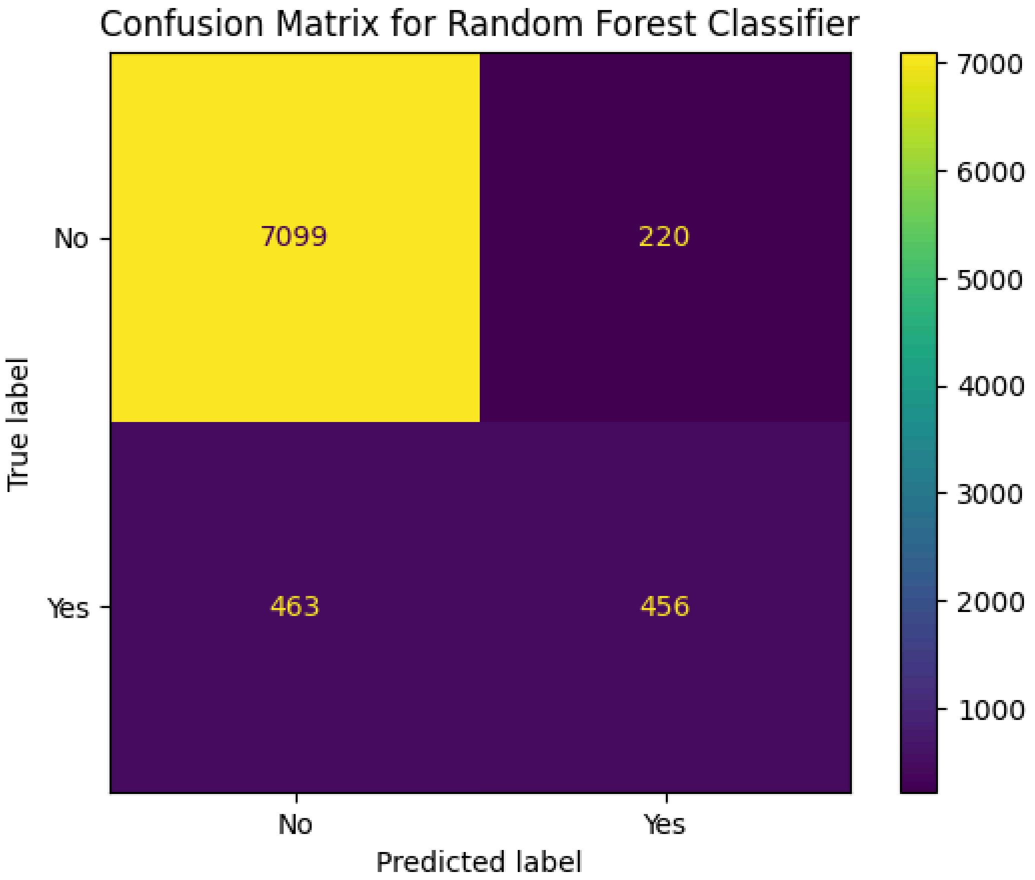
The initial models provided promising starts, but achieving the target accuracy required iterative refinement. By adjusting model parameters and improving data quality, I was able to meet the desired accuracy levels, particularly with the Random Forest model, which benefited significantly from this process.

This experience highlighted the importance of algorithm selection, data preparation, and hyperparameter tuning in reaching optimal model performance.

Results

The random forest classifier was a better performing algorithm in general it got 92% accuracy in its predictions 1% more than the Logistic Regression model with only 91% accuracy. The Random Forest Classifier model was also better when it came to getting the positive predictions right and in this use case its much more useful to have some extra yes answers than by accident removing potential customers. You can see these results in the images below.




Confusion Matrix for Random Forest Classifier
True label
No
Yes
7099
220
463
456
No
Yes
Predicted label
7000
6000
5000
4000
3000
2000
1000

References

Google Colab. (n.d.). colab.google. Retrieved February 4, 2024, from https://colab.google/

Matplotlib. (n.d.). Matplotlib — Visualization with Python. Retrieved February 4, 2024, from https://matplotlib.org/

Pandas. (n.d.). pandas - Python Data Analysis Library. Retrieved February 4, 2024, from https://pandas.pydata.org/

Python. (n.d.). Python.org. https://www.python.org/

Scikit Learn. (n.d.). scikit-learn: machine learning in Python — scikit-learn 1.4.0 documentation. Retrieved February 4, 2024, from https://scikit-learn.org/stable/