10 Academy Batch 3: Week 6

Bank Institution Term Deposit Predictive Model

Overview

Business Need

You successfully finished up to your rigorous job interview process with Bank of Portugal as a machine learning researcher. The investment and portfolio department would want to be able to identify their customers who potentially would subscribe to their term deposits. As there has been heightened interest of marketing managers to carefully tune their directed campaigns to the rigorous selection of contacts, the goal of your employer is to find a model that can predict which future clients who would subscribe to their term deposit. Having such an effective predictive model can help increase their campaign efficiency as they would be able to identify customers who would subscribe to their term deposit and thereby direct their marketing efforts to them. This would help them better manage their resources (e.g human effort, phone calls, time)
The Bank of Portugal, therefore, collected a huge amount of data that includes customers profiles of those who have to subscribe to term deposits and the ones who did not subscribe to a term deposit. As their newly employed machine learning researcher, they want you to come up with a robust predictive model that would help them identify customers who would or would not subscribe to their term deposit in the future.
Your main goal as a machine learning researcher is to carry out data exploration, data cleaning, feature extraction, and developing robust machine learning algorithms that would aid them in the department.

Data and Features

The dataset should be downloaded from the UCI ML website and more details about the data can be read from the same website. From the website, you would find access to four datasets:

1. Bank-additional-full CSV with all examples

2. Bank-additional.csv with 10% of data examples
3. Bank-full.csv
4. Bank.csv with 10% of 17 inputs

NB: For this work, download the first CSV (bank_additional_full.csv) with all examples.

bank client data:

1 - **age** (numeric)

2 - **job** : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')

3 - **marital** : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)

4 - **education (categorical)**: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')

5 - **default:** has credit in default? (categorical: 'no','yes','unknown')

6 - **housing:** has housing loan? (categorical: 'no','yes','unknown')

7 - **loan:** has personal loan? (categorical: 'no','yes','unknown')

related with the last contact of the current campaign:

8 - **contact:** contact communication type (categorical: 'cellular','telephone')

9 - **month:** last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

10 - **day_of_week**: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

11 - **duration**: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

other attributes:

12 - **campaign:** number of contacts performed during this campaign and for this client (numeric, includes last contact)

13 - **pdays:** number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

14 - **previous:** number of contacts performed before this campaign and for this

client (numeric)

15 - **poutcome**: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

social and economic context attributes

16 - **emp.var.rate:** employment variation rate - quarterly indicator (numeric)

17 - **cons.price.idx:** consumer price index - monthly indicator (numeric)

18 - **cons.conf.idx:** consumer confidence index - monthly indicator (numeric)

19 - **euribor3m:** euribor 3 month rate - daily indicator (numeric)

20 - **nr.employed:** number of employees - quarterly indicator (numeric)

Output variable (desired target):

21 - **y** - has the client subscribed to a term deposit? (binary: 'yes','no')

Learning Outcomes

- Students would learn to use data visualization tools or software such as Tableau, Matplolib, Seaborn
- Understand the end to end pipeline in building a machine learning model
- Knowledge in handling **class imbalance** problems
- Gain practical knowledge on how to do a **label encoding vs one-hot encoding** improves model interpretation and performance.
- Fine tuning **hyperparameters** to enhance model performance
- Using Github for version control with recommended folder tree
- Documentation of results to both technical and non-technical audiences

Team

- Deborah Dormah Kanubala
- Yabebal Fantaye,

Key Dates

- Discussion on the case - 1130 Rwanda time on Monday 24 August 2020. Use #all-week6 to pre-ask questions.
- Interim Solution - 2000 Rwanda time on Tuesday 25 August 2020.
- Final Submission - 2000 Rwanda time on Saturday 29 August 2020

Grading for the week

There are 100 points available for the week.

20 points - community growth and peer support. This includes supporting other learners by answering questions (Slack), asking good questions (Slack), participating (not only attending) daily standups (GMeet), and sharing links and other learning resources with other learners.

25 points - presentation and reporting.

5 points - interim submission

5 - Requirements met, clear presentation

3 - Most requirements met, presentation acceptable

1 - Some effort made

20 points for the final submission. This is measured through:

- Clarity of graphs (5 points)
- Clarity of message (5 points)
- Professionalism/production value (free of spelling errors, use of same font, well-produced) (5 points)
- Balance between being 'full of information' and 'easy to understand' (5 points)

55 points - data analysis and coding

10 points - interim submission

The validity of recommendations made (5 points)

Quality of code (including readability) (5 points)

45 points - final submission

The validity of recommendations made (25 points)

Quality of code (20 points)

Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20

points to the overall score.

> **Visualization** - the quality of visualizations, understandability, skimmability, choice of visualization
> **Quality of code** - reliability, maintainability, efficiency, commenting - in future this will be CICD
> **An innovative approach to analysis** -using the latest algorithms, adding in research paper content and other innovative approaches
> **Writing and presentation** - clarity of written outputs, clarity of slides, overall production value
> **Most supportive in the community** - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Data Scientist toolbox.

Late Submission Policy

Our goal is to prepare successful learners for the work and submitting late when given enough notice, shouldn't be necessary.

For interim submissions, those submitted 1-6 hours late will receive a maximum of 50% of the total possible grade. Those submitted >6 hours late may receive feedback, but will not receive a grade.

For final submissions, those submitted 1-24 hours late, will receive a maximum of 50% of the total possible grade. Those submitted >24 hours late may receive feedback, but will not receive a grade.

When calculating the leaderboard score:

- From week 8 onwards, your two lowest weeks' scores will not be considered.

Instructions

The task is divided into the following objectives

- Data Visualization using Tableau, Matplotlib/ Seaborn
- Classification model for predicting term deposit
- A detailed and well-written report

Task 1 - Exploratory Data Analysis

This is an important aspect of the machine learning model development pipeline. In this task, you are expected to carry out detailed data visualization on the data. The insight you extract here will help in the feature engineering and model development stages.

Task 1.1 - Visualization using Tableau

From the data visualization, you should be able to identify columns with missing values, outliers, and be able to find the best methods to handle problems of these sorts. You can begin by exploring the following:

EDA to help you understand the data, and get insight about the data generation process

1. Univariate Analysis on all columns (You can make use of bar charts, pie charts in understanding the categorical columns. Plots such as histograms can be useful for continuous columns. Be creative and think about better graphs to use to communicate what is in the data)
2. Bivariate Analysis. Carry out some analysis to see how two columns are related. Given that this is a classification problem, cross-tabulation between most categorical columns and the target would be helpful. For numerical columns use boxplots to understand the data separation.

EDA to help you plan selecting appropriate algorithms

3. Identify the class imbalance - what do you do to improve?
4. From the bivariate analysis, can you identify potential columns that could be helpful in currently separating the target column?

EDA to help you identify important features and perform feature engineering

5. Check the correlation between some of the continuous columns.
6. Check to see how many columns have outliers. Outliers can skew statistical measures thereby providing misleading representation of your data. See reference to a guide in handling data with outliers. **NB:** It could be a great exercise to compare the model performance with outliers and without outliers.

Task 2 - Classification Model for Predicting Term Deposit

Task 2.1 - Preprocessing

1. Encode categorical variables - identify all categorical columns and use one-hot encoding to transform them into numerical columns.
2. Handle outliers.
3. Use your preferred scaler to rescale all numerical columns. You can read about using MInMaxScaler(), StandardScaler(). The main idea why this is done is that some variables are often measured at different scales and would not contribute equally to model fitting and this may lead the trained model to create some bias. Hence, in dealing with it is usually important to normalize.
4. Transform and Aggregate columns to create better features
5. Explore TSNE, autoencoders, and PCA dimensionality reductions techniques.

The objective of this challenge is to build a predictive model if a new bank client would subscribe to a term deposit or not. The following steps would be helpful to carry this task out but you are allowed to use your approach.

Task 2.2 - Train, Test Split

Split the data using 90% for training and 10% for testing

Task 2.3 - Select Machine Learning Models

Use cross validation to select the best three machine learning models for submission. Use the following cross validation techniques and comment on their differences as well as pros and cons.

- Stratified K-fold
- K-fold

Suggested model to explore - you must try the ones in bold:

- **Logistic regression model**
- **XGBoost**
- **Multilayer perceptron**
- SVM (rbf, poly, linear)

- Decision Trees & Randomforest

Suggested evaluation metrics to use:

- ROC
- F1 Score
- Accuracy, Precision, Recall

Comment on

- Why do you use k-fold training technique? Does Stratified K-fold improve your model? Why or why not?
- Why do certain models perform better than others for the data we have?
- Which evaluation metric is more appropriate for this project, and why?

Task 3 - A detailed and well written report

Write a Medium article summarizing the pre processing carried out on the data, the idea behind choosing the type of k-fold, model algorithm, and evaluation metric. Present a justification for choosing the three models. Your article should have a link to your code in github. Your code in Github should at least have three model functions (if you prefer, you can make module folders instead) and a README and requirements.txt files.

Your github folder should at least have the following:

- **README** - explaining the project, and a guide on how to run the code
- **Requirement.txt** - which python packages are needed to run your code
- **Main.py** - imports all the necessary classes and functions from other files and automates the process of pre-processing, model training, and model prediction.
- **Data.py** - contains all functions and classes you write to do the pre-processing
- **Model.py** - contains all functions and classes you write to generate your three models
- **notebooks/** - a folder that contains jupyter notebooks you use to develop your code

Interim Submission (Due Tuesday 25.08 20hr Rwanda time)

- Your employer wants a quick meeting after you've done a first quick pass of the data and wants to know whether further investigation is useful. To achieve this, summarize your findings from Exploratory Data Analysis (task 1) in a Tableau dashboard.

Feedback

You may not receive detailed comments on your interim submission, but will receive a grade.

Final Submission (Due Sat 29.08 20hr Rwanda time)

- Link to your Github code that includes all your code
- Submit a PDF of the article you wrote for a medium post along with a link to the article published online

References:

1. Introduction to Tableau
2. Outlier Detection Algorithms in Python
3. Multilayer Perceptron
4. SVM
5. Xgboost
6. RandomForest
7. https://arxiv.org/pdf/1503.06410.pdf