Show me the Money

Bailey Korfhage, bkorfhage2@bellarmine.edu

ABSTRACT

The task of this project was to analyze a data set using Python and complete a classification model. The data set I chose required a variety of continuous and discrete variables that could potentially predict another variable. I used Python to explore the data, create the logistic regression model, and check its accuracy.

I. INTRODUCTION

I obtained the dataset from Kaggle.com. The dataset contained values including age, marital status, education, balance, housing, loan, duration, and y. Using the logistic regression model, I chose to try and predict whether a bank customer bought a term deposit based on the factor given in the dataset.

II. BACKGROUND

A. Data Set Description

The dataset came from a user by the name Aslan Ahmendov that was uploaded on Kaggle.com. This dataset's purpose was to collect data about contact with bank customers including if they have housing, loans with the bank, and duration of contact over the phone. The customer was offered a term deposit and that information was recorded in the 'y' column. As a student studying finance, the banking industry is very interesting to me, and the engagement with customers is important to any business.

B. Machine Learning Model

The logistic regression model is an approach for data synthesis using mathematical figures. It attempts to use multiple variables at the same time to predict a qualitative variable. The model takes the actual values of the continuous variables and the coded 0s and 1s of the discrete variables to make a prediction model. A classification report is then run to get an accuracy score, a score that is calculated by taking the weighted calculation of precision and recall. The closer this accuracy score is to 1, the better the model.

III. EXPLORATORY ANALYSIS

This data set contains 45,210 samples with 8 columns with both continuous and discrete data types. Two variables, age and balance, had missing values. The two plots are shown in the Appendix. Figure 1 was the plot for age and Figure 2 was the plot for balance. Both were right skewed and required the median to be used for filling in the values.

Table 1: Data Types

Variable Name	*Data Type*
age	continuous
marital	discrete
education	discrete
balance	continuous
housing	discrete
loan	discrete
duration	continuous

IV. METHODS

A. Data Preparation

The original dataset had 18 columns. Many of these columns were related to the premise of contacting the customer. However, I wanted this model to be more about the relationship between the bank and the customer, not the statistics on the contact attempts. I felt this was necessary for tidying up the data and creating a nicer model overall.

B. Experimental Design

Table X: Experiment Parameters

Experiment Number	Parameters
1	60/40 split for train and test
2	80/20 split for train and test
3	90/10 split for train and test

C. Tools Used

The following tools were used for this analysis: Python v6.4.8 running the Anaconda 2.2.0 environment for HP Laptop computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 1.4.2, Numpy 1.21.05, Matplotlib 3.5.1, Seaborn 0.11.2, Sklearn 1.0.2.

Python libraries: pandas, sklearn, numpy, seaborn

- Pandas allowed me to import the data set and to convert the categorical variables into a numeric pattern.
- Sklearn was the major tool used for Logistic regression. It split the data into the training and testing set. It also set up the model for logistic regression and the classification report.
- Numpy was used for filling in the missing values.
- Seaborn was used to plot the variables with missing data to determine which measure of centrality was appropriate.

V. RESULTS

A. Classification Measures

See Appendix for visual representations of the confusion matrix and classification report.

B. Discussion of Results

The model was a good prediction for customers who purchased term deposits at the bank. All three reports had an accuracy score above .88. Unfortunately, of the two false accurate predictions, the false negative was more frequent than the false positive, which is not preferable.

C. Problems Encountered

As mentioned before, the dataset was originally double its size. I would have preferred to use a different dataset but finding one with a result built into the dataset was very difficult. In addition, when applying the model, Python is very intuitive for calculating the logistic regression. This makes it difficult to explain the model in mathematic terms.

D. Limitations of Implementation

The accuracy score indicates that the model is good. However, the score was higher when the test size was larger, which is not a good indicator. It means that as the model had less information, it was able to predict better, but it was not able to be applied in context.

E. Improvements/Future Work

I think that it would be interesting to take the original dataset and extract only the variables related to contact, the ones I extracted originally. This might make the model more fitting because the connection is human.

VI. CONCLUSION

Overall, the model and the effort behind it proved to be successful. Despite the unfavorable count of false negatives, the accuracy score is an indicator of a good model. The dataset was quite extensive in nature, so I decided to cut nonessential variables. I used Python to explore, split, and train the data for the logistic regression. There were three experiments with different splits for training and testing. These experiments slightly varied in results but had preferable accuracy scores.

VII. APPENDIX

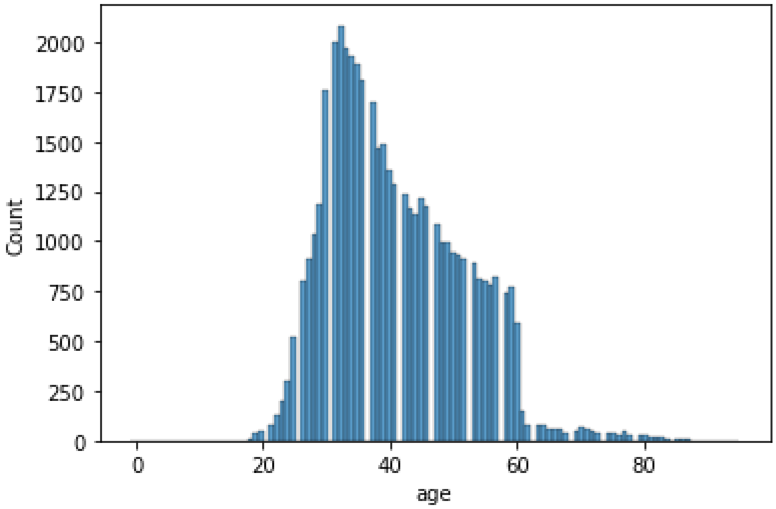


Figure 1

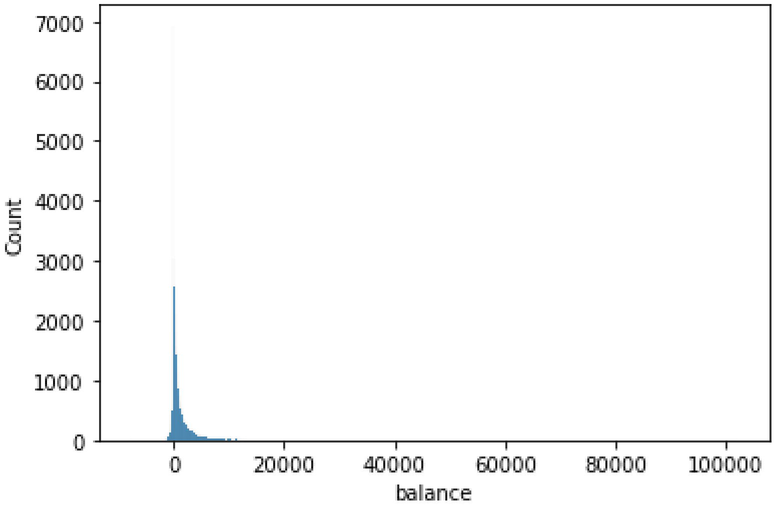
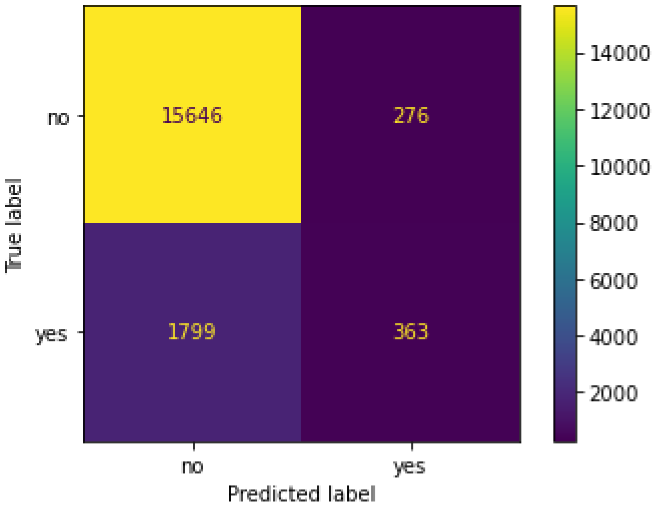
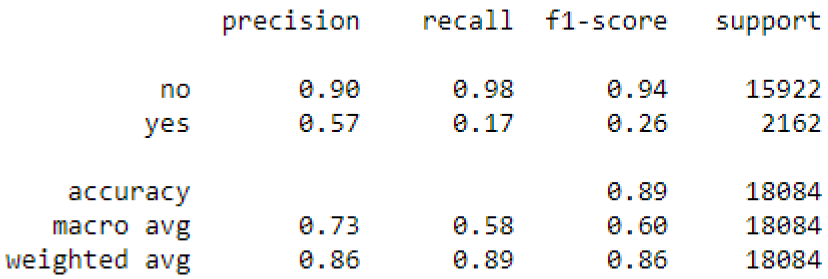


Figure 2

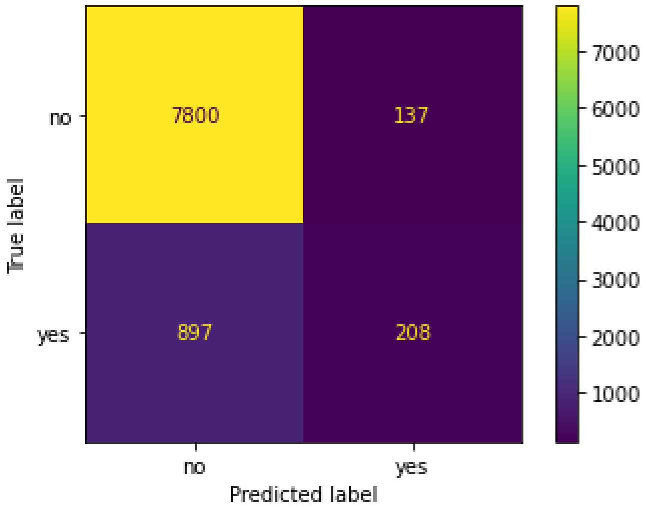


Confusion Matrix: Experiment 1

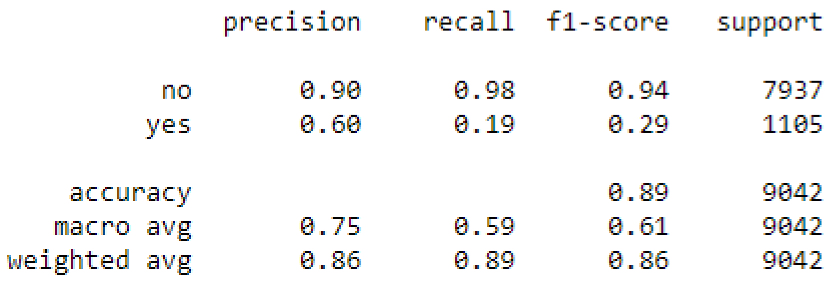


```
              precision    recall  f1-score   support

          no       0.90      0.98      0.94     15922
         yes       0.57      0.17      0.26      2162

    accuracy                           0.89     18084
   macro avg       0.73      0.58      0.60     18084
weighted avg       0.86      0.89      0.86     18084
```

Classification Report: Experiment 1

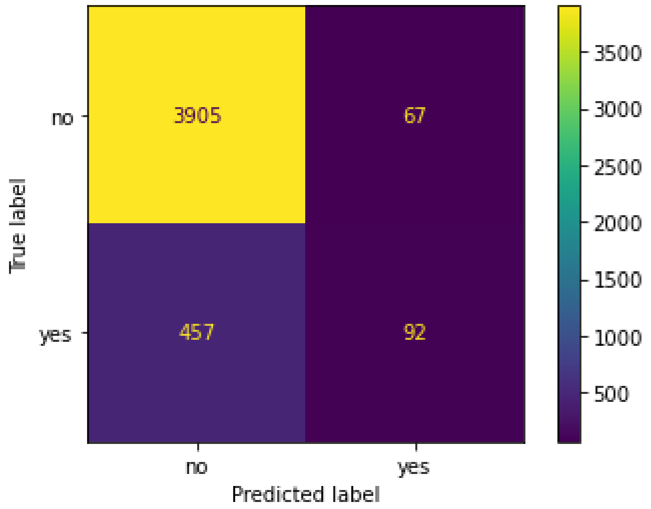


Confusion Matrix: Experiment 2

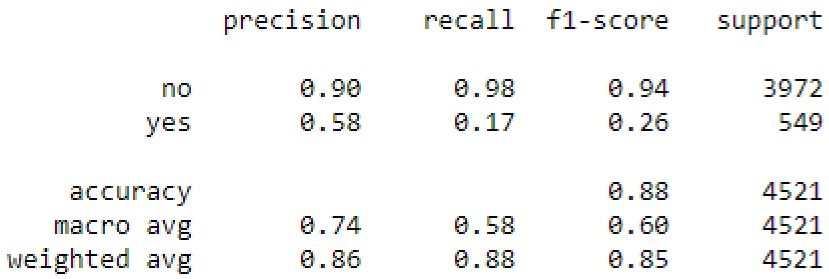


```
              precision    recall  f1-score   support

          no       0.90      0.98      0.94      7937
         yes       0.60      0.19      0.29      1105

    accuracy                           0.89      9042
   macro avg       0.75      0.59      0.61      9042
weighted avg       0.86      0.89      0.86      9042
```

Classification Report: Experiment 2



Confusion Matrix: Experiment 3



```
              precision    recall  f1-score   support

          no       0.90      0.98      0.94      3972
         yes       0.58      0.17      0.26       549

    accuracy                           0.88      4521
   macro avg       0.74      0.58      0.60      4521
weighted avg       0.86      0.88      0.85      4521
```

Classification Report: Experiment 3