Car Purchase Predictor (Something Interesting)

Ben Griffis, bgriffis@bellarmine.edu

ABSTRACT
For this project, I was assigned to perform an analysis on a dataset using Python and develop a classification model. The dataset I selected contained a mix of continuous and discrete variables that could potentially be used to predict another variable. Using Python, I conducted data exploration, built a logistic regression model, and evaluated its accuracy.

I. INTRODUCTION

I found the dataset on Kaggle.com. The dataset contained values including 'User ID', 'Gender', 'Age', 'AnnualSalary', and 'Purchased'. Using the logistic regression model, I chose to try and predict whether a customer purchased a car or not based on the factor given in the dataset.

II. BACKGROUND

A. Data Set Description

The dataset came from a user by the name GABRIEL SANTELLO that was uploaded on Kaggle.com. This dataset's purpose was to collect data about specific individuals with differing salaries, genders, and ages to determine whether they purchased a car or not. The customers decision to purchase a car was recorded in the "Purchased" column and was represented by either a "1" or "0", yes was represented by "1" and no was represented by "0". Being someone that's into business and specifically has a interest in cars I thought this was an interesting dataset to analyze.

Machine Learning Model

The machine learning model used in this project is logistic regression, which involves using mathematical calculations to synthesize data. It uses a combination of continuous and discrete variables to predict a categorical outcome. The model takes into account the real values of the continuous variables and the coded 0s and 1s of the discrete variables to develop a predictive model. After building the model, a classification report is generated to calculate its accuracy score, which is determined by weighing the precision and recall values. A higher accuracy score, closer to 1, indicates a better-performing model.

III. EXPLORATORY ANALYSIS

This data set contains 1,000 samples with 5 columns with both continuous and discrete data types. None of the columns had any missing values or unique values that had to be changed/adjusted/or imputed.

Table 1: Data Types

Variable Name	*Data Type*
V1 User ID	continuous
V2 Gender	discrete
V3 Age	continuous
V4 AnnualSalary	continuous
V5 Purchased	discrete

IV. METHODS

In this section, describe how you prepared the data for your model and performed multiple experiments using different parameters for the model.

A. Data Preparation

The original dataset only had 5 columns, so I felt there was no need to remove any of the columns. There was also no missing data, so nothing had to be prepared in that sense. The one thing that was changed in the dataset was assigning the purchased column to a new Y variable because that was the column that we were trying to predict with this model.

B. Experimental Design

Table X: Experiment Parameters

Experiment Number	**Parameters**
1	60/40 split for train and test set
2	80/20 split for train and test set

3	90/10 split for train and test set

C. Tools Used

The following tools were used for this analysis: Python v6.4.8 running the Anaconda 2.2.0 environment for Macintosh HD was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 1.4.2, NumPy 1.21.5, Matplotlib 3.5.1, Seaborn 0.11.2, Sklearn 0.0.post4

V. RESULTS

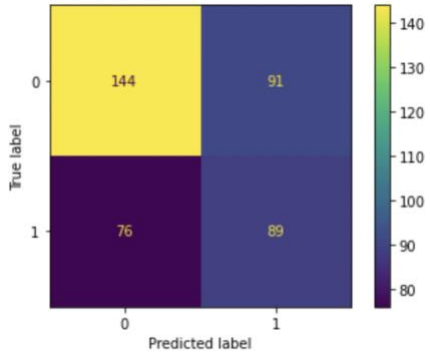
A. Classification Measures

Provide the classification measures for each experiment using a confusion matrix and classification report.

TEST 1

60/40 split for train/test set

Confusion Matrix Test 1



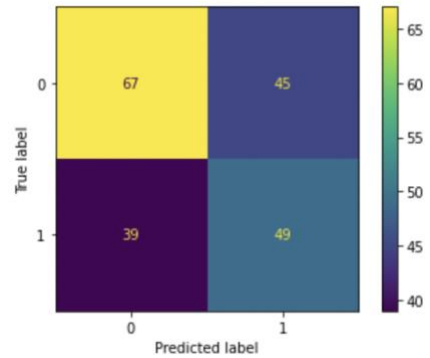
Classification Report Test 1

	precision	recall	f1-score	support
0	0.65	0.61	0.63	235
1	0.49	0.54	0.52	165
accuracy			0.58	400
macro avg	0.57	0.58	0.57	400
weighted avg	0.59	0.58	0.58	400

TEST 2

80/20 split for train/test set

Confusion Matrix Test 2



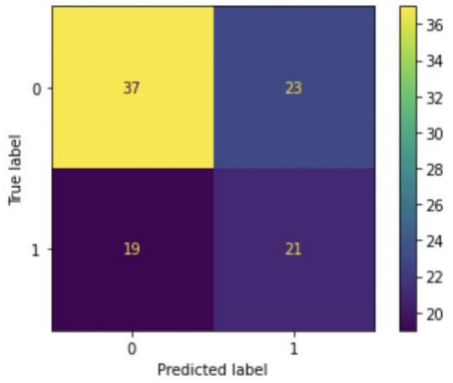
Classification Report Test 2

	precision	recall	f1-score	support
0	0.63	0.60	0.61	112
1	0.52	0.56	0.54	88
accuracy			0.58	200
macro avg	0.58	0.58	0.58	200
weighted avg	0.58	0.58	0.58	200

TEST 3

90/10 split for train/test set

Confusion Matrix Test 3



Classification Report Test 3

	precision	recall	f1-score	support
0	0.66	0.62	0.64	60
1	0.48	0.53	0.50	40
accuracy			0.58	100
macro avg	0.57	0.57	0.57	100
weighted avg	0.59	0.58	0.58	100

B. Discussion of Results

The baseline accuracy score of this dataset was 57.6% and after running the classification report it was observed to be 58%. We can say that the model performed slightly better than the baseline accuracy of 57.6%. Looking at the confusion matrix's we can see that in each test/experiment that the model predicted more True Negatives/Positives then False Negatives/Positives which is a what you want to see in a model. This basically means that it was more right on the predictions then it was wrong, but we can't ignore the accuracy.

Problems Encountered

I didn't run into too many problems, except for with the confusion matrix. When creating the confusion matrix in python I was getting an error with "disp.plot()" because I didn't assign the "Purchase" column to a Y variable. Besides that, another problem/confusing part I was running into was once again with the confusion matrix, I was struggling to read the matrix because "no" or 0 is on the top and "yes" or 1 is on the bottom along with on the other side "no" being on the left and "yes" being on the right. It wasn't a big deal it was just making it harder to read the graph personally.

C. Limitations of Implementation

Linear regression assumes that there is a linear relationship between the predictor variables and the target variable. However, in real-world datasets, this is sometimes not the case. For example, there may be nonlinear relationships between the features (such as age, income, or credit score) and the likelihood of purchasing a car. If the relationships between the variables are nonlinear, linear regression may not be an appropriate model.

D. Improvements/Future Work

Like said in the prompt previously linear regression might not be the best model to use for this data set so using another model would be something I'd look into. I'd also like there to be more data in this set it's only a 1,000-

person sample size, in general the more data you have the easier it is to make conclusions about the general population.

VI. CONCLUSION

Overall, a model was created and was technically better than the baseline accuracy score, though it was a very minimal improvement, it still outperformed the baseline. There were both more true positives and true negatives then false so that is a positive to look for in all models. The dataset wasn't the most extensive in nature, there were only 1,000 samples so for future works and if I were to want to make any general conclusions about the population, I would want more data to analyze. Regardless, 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, but these experiments didn't vary in results.