Tel-Aviv University

The Iby and Aladar Fleischman Faculty of Engineering

Digital Sciences for High-Tech

Sessions of users on E-Commerce

Introduction to Machine Learning (0560182401)

Lecturer: Dor Bank.

Student names: Dana Dahan, ID: 316468859. Yonatan Klyiner, ID: 314996588.

Year of study: Second year, Bachelor's degree in Science (BSc).

Date of Submission: 09/06/2022.

Chapter One: The Project Process

Part One – Exploration: In this section, we wanted to get as much information about the data as possible for efficient pre-processing. For example - how many features are there in the data and what is their type, how many missing values are there in each feature, and is there a correlation between features.

We used all sorts of functions that helped us read and understand the data. At first, we used the "describe" function to see the arithmetic parameters of each of the features: mean, standard deviation, etc. It helped us get a better idea of the values in each feature. We then looked at "shape", the number of observations, and the number of features in the data. Using the "info" function we saw the type of each feature and the number of its missing values. We decided to define two lists of numerical and categorical variables that will help us later in the project. Then we used "msno" to plot the number of missing values. This is how we discovered which features can be removed from the data due to many missing values.

After that, we checked the data distributions. In Machine Learning, data satisfying Normal Distribution is beneficial for model building. In the plots, we have seen some features that are normally distributed and those that are close to being distributed normally. We then plotted categorical variables to see their values and their distribution. It helps us to get a better knowledge about categorical features.

Part Two – Preprocessing: In the next section, we have made a great effort to process the data into a beneficial form that will suit the models and species of the dataset. We focused on a few things:

- Distinguish the set of categorical and numerical features - Throughout the pre-processing we kept the division of the categorical and numerical features into two lists and updated them throughout the processing. These lists are important so that we know who the numerical features are and who the categorical ones are so that we know how to address them later. We asked who a categorical variable in his being (represents categories and not numeric values) should be changed to such, and same for the numeric variables, we needed to make some transformations (full explanation in code comments).

- Removing features that are not relevant to the prediction of making a purchase: We have chosen to remove features that are not relevant to the data or that have many missing values. We also chose to remove highly correlated variables, for any variable that has a correlation higher than 0.8 with another variable, we chose to remove one of them to lower the model complexity.

- Filling in missing values to run models without "NA" values: We decided to fill in the missing values in a systematic way, if a variable is numeric, we filled in the missing value by a median and if it is categorical, we filled in the missing value using the most common value. We decided to fill in missing values after we converted numeric variables in their being of the type of object to int-type variables so that filling in the missing values would be relevant. However, there were also categorical variables that we later consolidated and edited, which we chose to fill in before the transformation. Because these features are categorical in nature, we will convert them to numeric variables for the models but fill in their missing values by being categorical. We came across continuous bug while filling missing values incorrectly, we identified the bug when it came up as an outlier.

- Removing Outliers: In this part, we located at which values we have the most extreme samples, that do not represent the true distribution with the help of the boxplots. and with a specific function according to the boxplot, we removed the unrepresented outliers from the train. We removed outliers from train set only to learn the best we can on the data, we chose to keep outliers in the test set to avoid samples removal.

- Normalization: Normalization gives equal weights/importance to each variable so that no single variable steers model performance in one direction just because they are bigger numbers. Because the data has variable scales and the technique, we used "StandardScaler" to normalize the values in the data set.

- Split to train and test- We have divided the data set of "Train" to "train" and "validation" for feature selection and models. We saved the original "train" set for the CV we will perform on the models afterward.

- Feature Selection- In this part, we reduce the number of features in different ways. First of all, we reduced the features that have a high amount of ‘NA’ because of their lack of real information telling. Second of all, we reduced features with a high correlation to each other,

because they give the same information. Thirdly, we ran two feature selection algorithms: PCA and forward selection, and checked which one gave the best MSE result. They gave pretty much the same result so we needed to choose a different reason to prefer one algorithm over the other, and we choose the number of features that have been reduced as our new parameters, in order to reduce overfitting and the algorithm with the least amount of features was forward selection so we chose him. After that, we ran the PCA algorithm on the data that has been reduced already with the forward algorithm in order to get a better result.

Part three – Modelling: In this section, we performed 5 models: Logistic Regression, Naïve Bayes Classifier, KNN, Multi-Layer Perceptron (ANN), and Support Vectors Machine.

We first set up two functions that test the AUC score, one is designed to test the score on the training set and the set of validation, and the other function is designed to test the AUC score using the cv method.

We then performed models one after the other, with each model choosing hyperparameters in two methods - either using Grid Search which brought us the best parameters according to the AUC score in the CV method, or using a list where we defined potential hyperparameters and chose the parameter that brought us the best score.

After finding hyperparameters, we re-trained each model with the parameters we found and then plotted the Confusion Matrix that illustrated the differences between the training and validation.

Finally, in each model we called the functions we defined for AUC and checked what score we got on train and validation to detect overfitting, and then we ran the function that tests AUC with CV to see what the final score is for each model (in a way closest to the real world).

Feature importance - when it comes to working on identifying the feature importance, we had a hard time because none of the models we worked on have a built-in function that provides this information, and when we tried to build such a function we could not identify what the remaining features were because we converted it to Numpy array that represents the features as numbers and we could not identify the features.

Part four – Model evaluations: In this section, we evaluated the models according to the AUC score using the cross-validation method. In the end, we released a plot that shows the

average AUC score of each of the models to illustrate the quality of the models against each other. Among other things, after each model, we presented a plot that shows the AUC score we received in each of the folds we performed.

Part five – Predictions: In this section, we took the model that brought us the best score – the MLP model and checked what the predictions of this model are on the test dataset. We converted the predictions to a Data frame and exported them to a CSV file as requested.

Chapter Two: Executive Summary

In our project, we performed an analysis of information from various features that indicate the behavior of the user who visits the site. we wanted to predict whether a user will make a purchase or not. At first, we got different values in the data – some of the values were number types and some of them were object types, while four features were anonymous columns. Our mission was to bring the data values to be optimized the feature ideas. It was also important to handle outliers and missing values and make normalization for modeling. After that, we wanted to identify the best model in terms of fitting (not overfitting or underfitting) and in term of the AUC score. After the learning process, it is possible to understand what are the factors that most influence the chances of purchase.

Chapter Three: Summary

Summary about the different models: In the Naive Bayes model, we got the lowest score, we are assuming the model is not ideal for our data set. in the KNN model we got overfitting, this is mainly reason why we did not choose it as the beat model. In logistic regression and support vector classifier we got similar scores, both are not overfitted but yet, we got a better model. The best model we got is the Multi-Layer perceptron, it has the highest AUC score and it’s not overfitted.

Appendix- Responsibility and division of labor:

Dana's part : import libraries, data loading, organize code duplicates into functions, code comments, and this very report. Part one - exploration: data distribution, correlations between features, check and plot missing values. Part two - preprocessing: create and update categoric and numeric lists, fill missing values, normalization on data, create x and y NumPy arrays, and split them into train and validation sets. Part three - modeling: all models - KNN, MLP, SVM, NB, LR; Including the selection of hyperparameters, plotting confusion matrix, and calculating AUC's score on train and validation separately. Part four - evaluation: AUC score functions, the final plot of AUC score on all models. Part five - predictions: pipeline, predictions on test and creation of CSV file of the predicted probabilities.

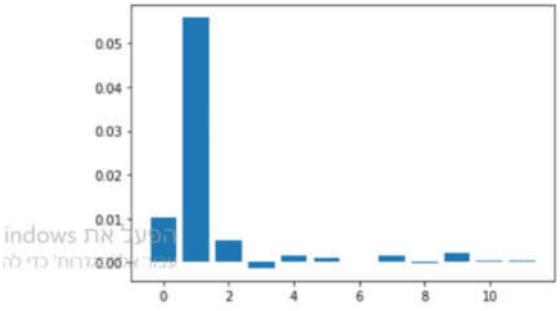
Jonatan's part : Part one - exploration: checking correlation between different features, correlation Between numeric features and purchase because this is the item, we want to check what affects him the most. Part two - preprocessing: remove outliers, features reduction, data transformation to dummies variables. Part three - modeling: plotting ROC curve CV on each one of the models. Also tried to make feature importance as we mention before. Part four - evaluation: evaluating each ROC curve CV on each one of the models.

Additional appendices

Note that if we had more time to work on the project, we would improve the following:

In terms of order and organization - We would organize the functions in a more orderly way, and go over their efficiency, readability, and functionality. We would edit the colors of the plots in a certain line to make them more aesthetically pleasing to the eye. In terms of exploration - We would examine more correlations between variables to better understand data behavior. In terms of modelling – exploring more about the scores of the modelling, why we got overfitting in KNN model, why NB has the lowest score, and what is the reason MLP was the best model for data. We also would like to find out what is the feature importance for at least one model, we added below our try for feature importance. In terms of model evaluation - Confirm the AUC score using system functions and not a function we built.

```
Feature: 3, Score: -0.00131
Feature: 4, Score: 0.00161
Feature: 5, Score: 0.00088
Feature: 6, Score: 0.00021
Feature: 7, Score: 0.00143
Feature: 8, Score: -0.00027
Feature: 9, Score: 0.00210
Feature: 10, Score: 0.00039
Feature: 11, Score: 0.00027
```



<- <u>Feature importance for NB model</u>: we did not have the time to identify the feature names, so we decided to remove it from the project .