Machine Learning – TechScape Ecommerce Project Report

Names and student numbers

**Abstract**

1. **Introduction**

Our challenge was to build a machine learning model for TechScape, an ecommerce company selling consumer products and services designed for detoxing from technology and improving balance with the digital space. We were provided a dataset of approximately 10,000 observations of customer interactions on the TechScape website, and if they bought a product of not. We also had a dataset of about 2300 unlabeled observations. Our goal was to build the best ML model to identify customers that will buy products.

Data provided included the date of the visit, information on where in the website the user visited, and the type of operating system and browser the user was working on.

1. **Techniques**

Introduction of any new techniques used:

* Power transformation (should be done after train-test split so you do not leech information into your validation dataset)

1. **Methodology**
   1. Logical Checks

Making sure the Access ID was unique, that duration and pages were constant (both were either above of below 0), and

* 1. Data exploration

On initial inspection, the dataset contained no null values, but did have outliers in a number of variables (see figure 1 for boxplots). There were 15 features and the target variable was “buy”, indicating whether or not the customer bought a product in the end.

It was also apparent that we were dealing with imbalanced data, as the “buy” variable only had an 18% positivity rate in the raw data.

We also analyzed the boxplots of the numerical data after separating buy=1 and buy=0, to visually see any differences in the distributions/outliers of each group. The only obvious differences in the median value of the ExitRate rate and of the PageValue variables.

* 1. Outliers

Split the data first, to avoid a falsely high f1 score in the validation data. Initial outlier detection was done using a number of different techniques.

First by IQR method, by removing all observations above X number of IQR ranges for all numeric variables. We tested this method with 1.5-5 IQRs, all of which led to lower f1 scores on trials with both NB classifiers and logistic regressions, meaning we were removing valuable data from our train data.

Next, by visual inspection of boxplots. The filters that ended up resulting in the best scores for our final variables were as follows:

(include table of the ranges for only the variables we kept in the end)

* 1. Missing values
  2. Feature engineering and transformation

Feature Engineering:

* Month

Dummy Variables: ‘OS’, ‘Browser’, ‘Type\_of\_traffic’, ‘Type\_of\_visitor’, ‘Month’

Looking at the distributions of our metric features, we decided to test different transformations, including minmax, scaler, log, and power transform, the best outcome was power transform.

* 1. Feature selection
     1. Correlation

We ran a correlation matrix on the non-target variables, and found high correlation between **'AccountMng\_Pages'**and **'AccountMng\_Duration', 'FAQ\_Pages'**and **'FAQ\_Duration', 'Product\_Pages'** and **'Product\_Duration', and 'GoogleAnalytics\_BounceRate'**and **'GoogleAnalytics\_ExitRate'.** We tried the lasso technique which determined that pages was more valuable for the first three pairs, and bounce rate over exit rate, so the latter variable was dropped for each of the above pairs.

* + 1. Chi-Squared

Dropped Country, which was not significant.

* + 1. RFE: Numeric: Power transformed page value, Categorical: type of traffic 12
    2. Lasso

balanced class, with all dummies, and the above two.

* 1. Data partition and Balancing

Over and undersampling resulting in overfitting

* 1. Testing Algorithms
     1. For each: why, parameters tested, results

1. **Results** (model we used and how we got there)

Because of the imbalance of data, we were getting many false positives in the confusion matrix. To help solve this we used a new probability threshold that was artificially higher than .5, which is shown to help deal with false positives. (source needed).

* 1. Model testing
  2. Boosting/stacking
  3. Grid search

1. **Conclusion**

Can we draw conclusions from this model and the methods we used?

1. **Figures**
2. **References**