BLG 456E Learning From Data Term Project Report

Onat ŞAHİN

Burak BEKCİ

Abstract

In this project, we created a machine learning application to predict whether a purchased item is returned or not. For this task, we cleaned the given training data and created features of our own. Then we also converted the test data to our format. With these data, we tried different machine learning algorithms to predict the data and chose the best one.

I. INTRODUCTION

For this project, we are asked to predict whether an item will be returned or not using its information. We are also given a training data. For this task, we needed to develop a machine learning system. However, given data was a very noisy. Therefore, we first cleaned the data and adjusted out features. Then, we proceeded to try different machine learning algorithms.

Kaggle Names: Onat Sahin, Burak Bekci

Team Name: 150150127_150150129

Score / Rank: 2123 / 10

II. DATA SET USED

Since the training data given to us was very noisy, we had to clean the data and create out features. First of all, we removed the data with unknown features, which are indicated by the "?" character. We used order date and delivery date information to calculate the delivery time and included this as a feature. From date of birth, we calculated age and mapped the ages to different values depending on its value. Ages between 0 and 10 are mapped to 0, ages between 10 and 20 are mapped to 1, and so on. Ages above 80 are mapped to 8. We used creation date to calculate the age of the item using today's date. We included price as a feature without changing it. For the rest of the information given, using the training data we calculated the probability of return for every value the feature took. We included these probabilities as features. We also applied these operations to the test data. However, there were some customer ids which were in the test data but not in the training data. Since we calculated the probability of return for customer ids in the training set and predict according to these probabilities, we predicted the probability of unknown customer ids as 0.5.

III. METHODS USED

We wrote our whole code in Python 3. We have used Python 3 because it supports many libraries such as; Pandas[1], Numpy[2] and Sklearn[3]. We used Pandas library to calculate the probability of items to be returned according to a feature. Numpy used in matrix operations and storing our data. Lastly, we used Sklearn's models to make our predictions. Models that we have used in Sklearn are Linear Logistic Regression, Random Forest Classifier, Neural Networks and Decision Tree. Moreover, we used some basic libraries of Python such as datetime to make calendar operations.

IV. RESULTS

Firstly, we tried a basic model. We used the train and test set that are described in the II. Section and we normalized that data before training our model. We normalized the data by subtracting the min of instances' value of that feature from each instances and dividing this value by the term max – min value of that feature. For the predictions we used Linear Logistic Regression model of Sklearn library. This model gave around 12 thousand points. We discovered that most of our probabilities in the range of [0.4,0.7]. After that discovery, we only used the probability values 0 or 1 and our score improved to 9 thousand. As a second model we used Random Forests. This model gave a score around 8 thousand in the same train and test sets we have used before. After this point, we changed our features. We calculated probabilities that are given in the section II. We applied same operations to test set and made predictions with Random Forest, Neural Network and Decision Tree models. Among these models, Decision Tree gave the best score which is 2123.00000. While trying models, rather than submitting all submissions we used an algorithm to evaluate our submissions. This algorithm counts the number probabilities in each range of (0,0.3), (0.3,0.5), (0.5,0.7), (0.7,0.8), (0.8,1). We expect from a submission to give high score if it has many values in the range of (0,0.3) and (0.8,1).

V. CONCLUSIONS

We conclude that, normalization improves predictions. Data should compress in the range 0 to 1. Furthermore, Decision Tree and Random Forest are more suitable for our model because they gave better scores. Last but not least, using probability values rather than mapped integer values gave much better results.

REFERENCES

[1] Pandas version 0.23.0 May 2018. 2008-2012, AQR Capital Management, LLC, Lambda Foundry, Inc. and PyData Development Team.

[2] Numpy 2005-2018. NumPy developers. NumFocus.
[3] Scikit-Learn version 0.19.1. 2007 - 2017, scikit-learn developers.