Neil Pirch

Brian Hare

CompSci 461

9 December 2018

Churn Prediction with TensorFlow

In an effort to measure customer retention, companies with subscribing customers use the measure of churn, a percentage of customers who leave a company in a given period of time. By analyzing historical data on churn, a company may be able to predict future churn, as well as mitigate factors that influence churn. The goal of this project is to use machine learning to predict churn in customers of a telecommunications company.

Each row in the dataset represents a single customer. Columns in the dataset include whether a customer has “churned”, services they subscribe to, amount of monthly charges, and demographic data.

A TensorFlow environment was started in Anaconda, and the python script was built and run in Spyder. The dataset is read in using pandas.read\_csv, then specific fields are manipulated. The TotalCharges column is converted to a numeric value, and some null fields are filled by calculations using MonthlyCharges and Tenure columns. The SeniorCitizen value is converted to an object, and Churn ‘yes’ and ‘no’ values are converted to 1 and 0.

The independent variable set X consists of all columns except customerID (which is a unique identifier of each customer, and should not influence churn) and Churn. The dependent variable set Y contains the Churn values.

Next, the dataset is split into three sets: training, test, and validation. This is done in two steps. First, 15% of the X set is split into the test set. Of the remaining 85% of the original dataset, 17.647% is split into validation, equal to 15% of the total.

A linear classification model is used. Two classes are supplied, Churn: Yes or No. In batches of 100 lines, the model trains then tests and validates. One sample run resulted in the following data:

precision recall f1-score support

No 0.82 0.92 0.87 758

Yes 0.70 0.50 0.58 299

micro avg 0.80 0.80 0.80 1057

macro avg 0.76 0.71 0.73 1057

weighted avg 0.79 0.80 0.79 1057

Test set accuracy: 0.798

Finally, the script loops through the variables generated by the model and outputs them with their weights. According to this set of results, the variable with the strongest correlation to churn was TotalCharges:

linear/linear\_model/TotalCharges/weights/part\_0/Ftrl [[3.1735463e+13]]