Report

Óliver Partida

May 13, 2021

1 Data preprocessing

Neural networks input are numbers which encode our data. The final goal of this step is to code our data using number that the

- Drop irrelevant columns. Remove **ID** and **TIMESTAMP** columns from the training set. I believe these two fields are not relevant for the classification task. **TIMESTAMP** is the date the flight was booked and there are only two dates in the set.

- Missing values. Remove from training and test data sets all rows that have columns with empty fields. The column **DEVICE** has several rows with empty values.

- Transform dates to numerical value. The columns **DEPARTURE** and **ARRIVAL** have several different formats like **01/July** and **01-may** We have to transform the month name to its numerical value and remove the characters **-** and **/**.

- Convert **DISTANCE** column to float. Replace **,** for **.**.

- Encode categorical columns. For the columns **WEBSITE**, **DEVICE**, **HAUL TYPE**, **TRIP TYPE**, **PRODUCT** we use hot encoding. In this encoding each category is transformed into a new column. The category that any particular row had will be represented now by a 1 in the column assigned to that category and zeros in all the others. For the columns **SMS** and **TRAIN** we replace the values **TRUE/FALSE** with **1/0**.

- Scale the data. We scale the columns **DISTANCE**, **DEPARTURE** and **ARRIVAL** to be in the range 0,1.

- Finally we transform the targets from **TRUE/FALSE** to **1/0**.

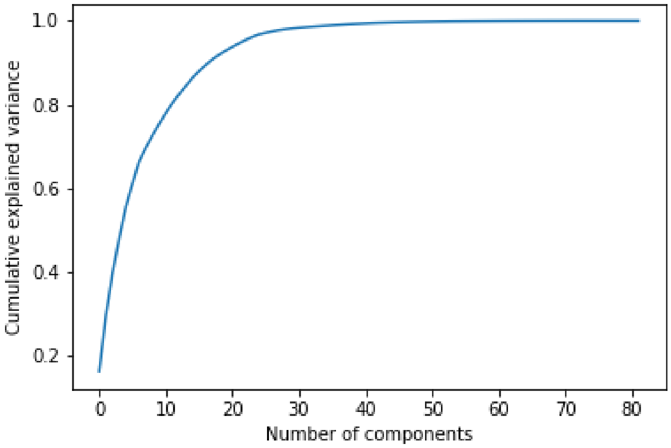

Figure 1: Data variance explained by components

2 Dimensionality reduction

We have 82 features. In order to avoid the curse of dimensionality we attempt a principal component analysis to eliminate correlated features. In figure 1 we see that around 30 components suffice to explain most of the variance in the data.

3 NN training

We tried a *NN* with one input layer with 30 neurons, three hidden layers with 100, 50 and 20 neurons respectively and *relu* activation function and finally an output layer with 1 neuron and *sigmoid* activation function. The loss function is *binary cross entropy*. The *NN* is trained in 200 epochs with 2500 samples in each batch. In figures 2 and 3 we see the evolution of the NN accuracy and loss on the training set in blue and validation set in red. Although in the training set the accuracy increases and the loss decreases in the validation set the accuracy starts at some point to decrease and the loss to increase which means the algorithm is overfitting the data.

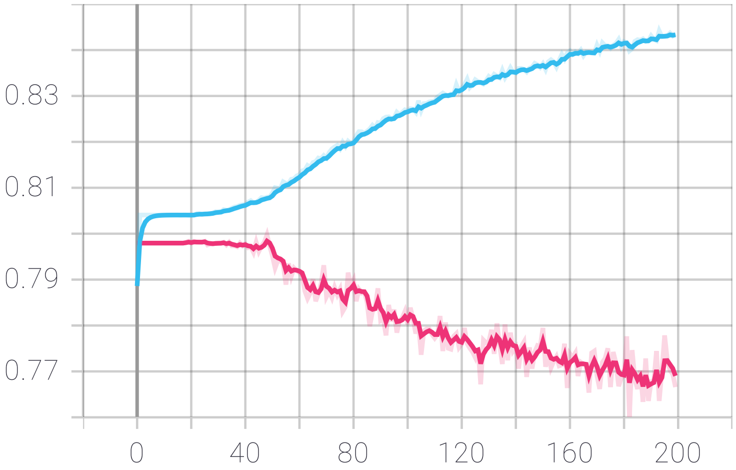

Figure 2: Epoch accuracy

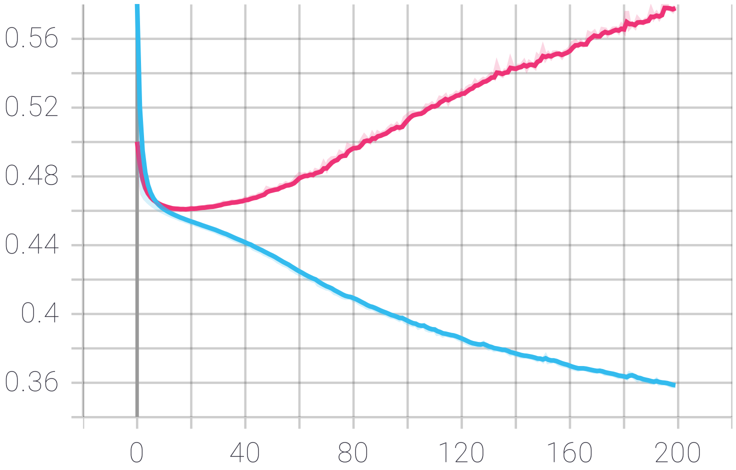

Figure 3: Epoch loss

4 Conclusions

Data preprocessing is a very important step to be able to train machine learning algorithms which can learn and extract patterns from the data. I did not have time to perform model selection to find the best NN architecture. The NN predicts 3162 samples that will buy extra baggage out of 30000 test samples. In 50000 samples in the training set there were 9799 samples that buy extra baggage. In we assume the test set have similar distribution the expected number of samples in the test set that would buy extra baggage is $0.6 \times 9799 = 5879$.