Machine Learning Engineer Nanodegree

Capstone Project – Starbucks Offers

David Morcuende Cantador

January 3rd, 2020

Proposal

Domain Background

This dataset provided by **Starbucks** contains simulated data about how their customers interact with the mobile app. Sometimes, an offer is sent to the users, this offers can be an **advertisement** for a new product or an offer such as a **discount** or **BOGO** (buy one get one).

Not all users receive the same offers, that might be one problem to solve with this dataset.

This dataset has a subsample of all the app **transactions**, some **users** and it **demographic information** and general **information about the offers**.

The offers have a **validity period**, so if a user doesn't complete the offer by this due date, the offer is invalid, this might be done to **encourage the customers to buy products** in that period of time.

The transactional data shows how the users interact with the app, with entries about if an offer has been informed, if it has been seen and if it has been completed, and also information about if the customer has bought something and the amount of money they have spent.

Keep in mind as well that someone using the app might make a purchase through the app without having received an offer or seen an offer.

Example

An user gets an offer that is a discount of 5 dollars which needs 15 dollars to be completed, and has a validity of 7 days. It means that a user must spent 15 dollars in less than 7 days since the reception of the offer.

However, a user can complete an offer even without having seen it, if a customer gets the offer explained above, but never saw it and spend 15 dollars the offer was completed but the customer was not influenced by the offer.

Cleaning

The user influence is important and to get correct solutions the completed offers must be real-completed offers, so the user has to received, seen and spend money in that order, because **if the user spends the money not being influenced by the offer, another offer should be considered for that user.**

This makes data cleaning especially important and tricky.

Problem Statement

You can solve the problem in three different ways depends on what are you looking for: **multi-label, multiclass** or as a **binary classifier**.

1. <u>multiclass</u>: in multi-class problems the classes are mutually exclusive
2. <u>multi-label</u>: in multi-label problems each label represents a different classification task, but the tasks are somehow related (so there is a benefit in tackling them together rather than separately)
3. <u>binary classifier</u>: is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to).

<u>1)</u>
For the multiclass problem the goal is to get the offer that will give the maximum benefit to the company for a user

<u>2)</u>
For the multi-label problem, the goal is to get the offers that a certain user will complete

<u>3)</u>
For the binary classifier problem, the goal is to answer if a certain user will complete an offer or not.

Classifier	Problems
Custom ANN	1,2
XGboost	3
Naive Bayes, BinaryRelevance	1
Naive Bayes, LabelPowerset	1
Naive Bayes, MLkNN	1
Naive Bayes, ClassifierChain	1

Datasets and Inputs

The data is contained in three files:

- portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)
- profile.json - demographic data for each customer
- transcript.json - records for transactions, offers received, offers viewed, and offers completed

Here is the schema and explanation of each variable in the files:

portfolio.json

- id (string) - offer id
- offer_type (string) - type of offer ie BOGO, discount, informational
- difficulty (int) - minimum required spend to complete an offer
- reward (int) - reward given for completing an offer
- duration (int) - time for offer to be open, in days
- channels (list of strings)

profile.json

- age (int) - age of the customer
- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
- id (str) - customer id
- income (float) - customer's income

transcript.json

- event (str) - record description (ie transaction, offer received, offer viewed, etc.)
- person (str) - customer id
- time (int) - time in hours since start of test. The data begins at time t=0
- value - (dict of strings) - either an offer id or transaction amount depending on the record

Solution Statement

Custom ANN

Artificial neural networks are built of simple elements called neurons, which take in a real value, multiply it by a weight, and run it through a non-linear activation function. By constructing multiple layers of neurons, each of which receives part of the input variables, and then passes on its results to the next layers, the network can learn very complex functions.

Its strength is its ability to dynamically create complex prediction functions and **emulate human thinking** (as this problem is about how humans complete offers), in a way that no other algorithm can. There are many classification problems for which neural networks have yielded the best results.

Naive Bayes Classifier

It calculates the probability that each of the features of a data point (the input variables) exists in each of the target classes. It then selects the category for which the probabilities are maximal. The model is based on an assumption (which is often not true) that the features are conditionally independent.

Naive Bayes is **surprisingly accurate for a large set of problem**s, scalable to very large data sets. But it **has problems where categories may be overlapping** or there are unknown categories so the set of categories selected must be exhaustive.

The naïve Bayes classifier has been implemented with:

- BinaryRelevance: Performs classification per label.
- ClassifierChain: A multi-label model that arranges binary classifiers into a chain.
- LabelPowerset: Transform multi-label problem to a multi-class problem.
- MLkNN: kNN classification method adapted for multi-label classification.

Benchmark Model

<u>BCELoss (For the ANN)</u>

Creates a criterion that measures the Binary Cross Entropy between the target and the output:

The unreduced loss can be described as:

$$\ell(x, y) = L = l1, \dots, lN\top, ln = -wn[yn \cdot logxn + (1 - yn) \cdot log(1 - xn)],$$

where N is the batch size. If reduction is not none, then this is used for measuring the error of a reconstruction in for example an auto-encoder. Note that the targets yy should be numbers between 0 and 1.

$$\ell(x, y) = \begin{cases} \text{mean}(L), & \text{if reduction} = \text{'mean';} \\ \text{sum}(L), & \text{if reduction} = \text{'sum'.} \end{cases}$$

<u>Precision</u>

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

<u>Recall</u>

Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

<u>F1-score</u>

F1 Score is the weighted average of Precision and Recall.

Evaluation Metrics

(approx. 1-2 paragraphs)

In this section, propose at least one evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model. The evaluation metric(s) you propose should be appropriate given the context of the data, the problem statement, and the intended solution. Describe how the evaluation metric(s) are derived and provide an example of their mathematical representations (if applicable). Complex evaluation metrics should be clearly defined and quantifiable (can be expressed in mathematical or logical terms).

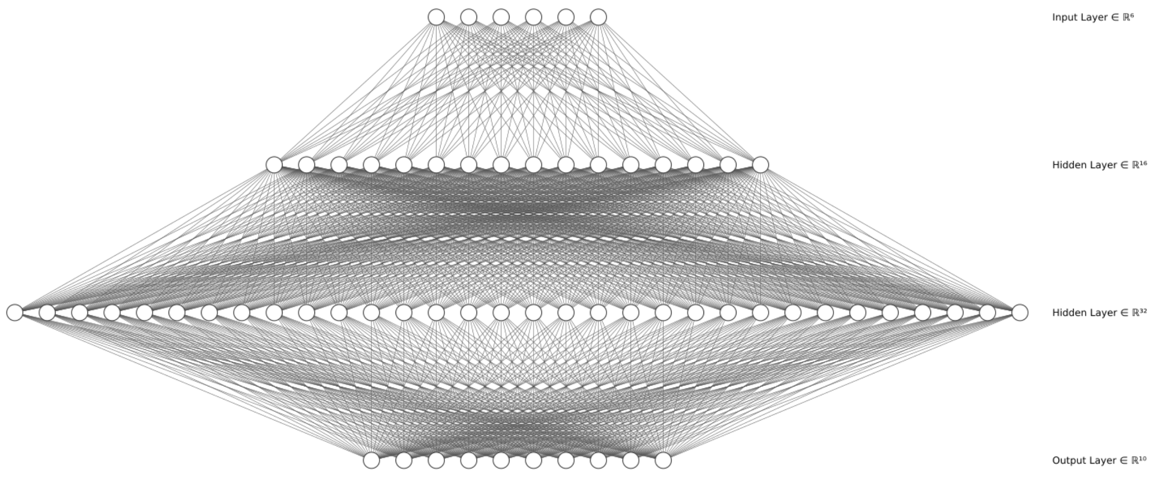
Project Design

The architectures are split into the different classification problems.

Multi-label

Custom ANN (pytorch implementation):

6 inputs 2 hidden layers (16,32) and 10 oputputs as there is 10 different offers



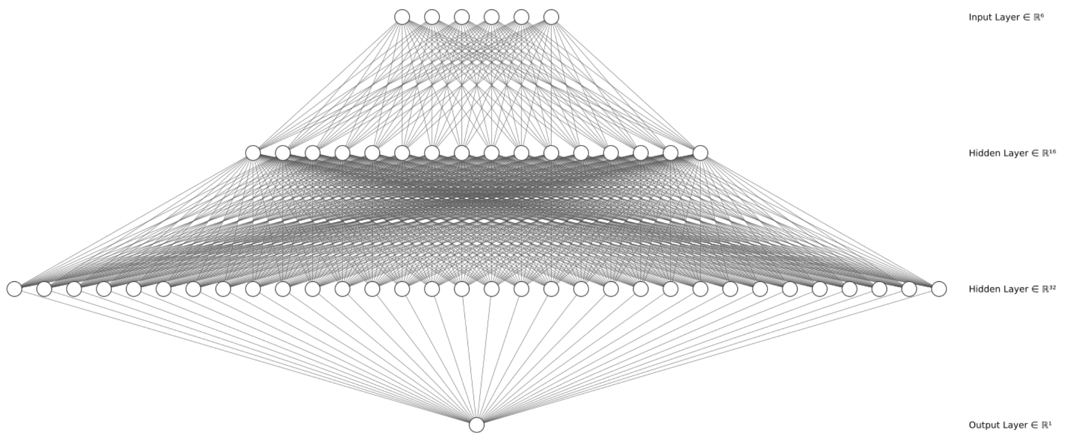
Scikit

Naive Bayes + BinaryRelevance
Naive Bayes + LabelPowerset
Naive Bayes + MLkNN
Naive Bayes + ClassifierChain

<u>Multiclass</u>

Custom ANN (pytorch implementation)

6 inputs 2 hidden layers (16,32) and 1 outputs the top rewarded offer by user


Input Layer $\in \mathbb{R}^6$
Hidden Layer $\in \mathbb{R}^{16}$
Hidden Layer $\in \mathbb{R}^{32}$
Output Layer $\in \mathbb{R}^1$

Xgboost Classifier

- Objective → multi:softmax

<u>Binary Clasifier</u>

Xgboost Classifier

- Objective → binary:logistic