Machine Learning Engineer Nanodegree

Capstone Proposal

Harold Giovanny García Rodriguez

Domain Background

Companies that make an investment in marketing want to get as much profit as possible, so if the customer profile is not considered for the marketing campaigns the profit will decrease, a solution to this problem is the targeted marketing using machine learning techniques. The main idea of the targeted marketing is that we can obtain target groups of potential customers who have a high probability of acquiring the product or service that is being offered, to avoid wasting money offering it to people who are most likely not interested. A simple example of a common machine learning workflow in a targeted marketing project can be find in [1], in which the author wants to predict if a customer is going to respond to an offer and take an initial dataset with 29 features and through the analysis of the data, dropping low information or constant columns, applying Box-Cox transformations, creating new features through a feature engineering process, reducing the dimensionality of some correlated features with PCA, and removing the outliers, the data was ready to train the model, the author chose to trained a logistic regression model getting a ROC score of 98%, which is almost a perfect score but with most realistic data the performance could still be very good and help boosting the profit, but maybe not as good as a 98%.

Problem Statement

It is important that each economic and operational effort made by the company to promote its offers has the highest possible return. That's why the main goal of this project is to use the data to identify which groups of people are most responsive to each type of offer, and thus be able to send offers in a more personalized level. This can be made through a binary classifier model that predicts whether or not someone is going to respond to an offer.

Datasets and Inputs

The data that I am going to use for this project is divided in three JSON files, and was proportioned by Udacity's Machine Learning Nanodegree:

- portfolio.json - containing offer ids and metadata 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

Below is the detailed description of each of the variables contained in the JSON 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

This data has all the information needed to get the patterns of how people respond to an offer, and I can use it to train the model after having done some preprocessing to it.

Solution Statement

The solution proposed to the problem consist in making a binary classifier model, to do this I am going to analyze and select the most relevant features with high predicting power to train several models, and then test it to get the metrics that will allow me to choose the model with the best performance. The output of the final model is going to be a prediction that indicates whether or not someone is going to respond to an offer.

Benchmark Model

The benchmark model that has a relation with the domain of the current project was developed in the Towards Data Science article titled “Machine Learning Classification with Python for Direct Marketing”. The main goal of this project was predicting the customer response to a bank’s telemarketing campaign that offers a deposit contract to the client. To accomplish this goal, the author trains different classification models like Support Vector Machine, Decision Tree, Random Forest and Logistic Regression, and after evaluating the resulting models with the F1-score metric and a confusion matrix, the one with the best performance was the Random Forest Classifier with an F1-score of 0.92.[2]

Evaluation Metrics

Considering that the final model will be a classification model, for evaluating it I am going to use three metrics: Precision, recall and the F1-Score, these are explained in more detail below:

- Precision and Recall:
  In this context these are two numbers which together are used to evaluate the performance of a binary classification model [3], based on the original and predicted condition (positive or negative), thus determining the values of true positives (true

condition predicted as true), true negatives (false condition predicted as false), false positives (false condition predicted as true) and false negatives (true condition predicted as false).
Mathematically, precision and recall can be computed as follows:

Precision attempts to answer the following question: What proportion of positive identifications was actually correct? [4]

$$Precision = \frac{\sum True\ Positive}{\sum Predicted\ Condition\ Positive} = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall attempts to answer the following question: What proportion of actual positives was identified correctly? [4].

$$Recall = \frac{\sum True\ Positive}{\sum Original\ Condition\ Positive} = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- F1-Score:
  The F-score is a measure of a test's accuracy. It is calculated from the precision and recall of the test. It is the harmonic mean of the precision and recall, it is important to know that the highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero [5]. The mathematical definition of the F1-Score is:

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2 * \frac{precision * recall}{precision + recall}$$

Project Design

The steps I am going to follow to develop the project are described below:

1. Getting the data:
   The data is provided by Udacity’s Machine Learning Engineer Nanodegree Capstone Project section, the datasets were previously described.
2. EDA (Exploratory Data Analysis):
   - After reading the JSON files into a jupyter notebook using Pandas I’m going to use the customers ID and offers ID to merge the information from the three datasets into an additional Dataframe to have a base data to the project.
   - I am going to clean the null values of the data, either by dropping the rows or columns or doing some imputation as appropriate.
   - Then it is important to transform the categorical features to a numerical ones, so they can be an input to our model, this can be done with dummy variables or a label encoder.
   - After this I will explore how balanced are the target classes, if there are balanced I could continue with the modeling step, but if they are not, I could use some technique to do over-sampling or under-sampling to balance them.
3. Modeling:
   - At first I will split the data into training and testing sets.

- Then I am going to train multiple classification models like random forest, XGBoost, logistic regression, support vector machine, among others, and evaluate their performance with previously unseen data in the test set, using the metrics explained in the Evaluation Metrics section, and choose the model with the best performance to continue the process.
- The hyperparameters of this final model will be tuned with Grid Search to obtain the best possible combination of parameters.

4. Evaluating the final model:
   The optimized model will be evaluated with the test set using the same metrics that will be used in the previous step to get the final performance metric and take the decision if it's good enough or if it needs to be tuned again. If it is good enough we have a model that allows us to predicts whether or not someone is going to respond to an offer.

Bibliography

[1] R. Saldanha, "Targeted Marketing with Machine Learning," 2020. [Online]. Available: https://ai.plainenglish.io/targeted-marketing-with-machine-learning-38de28162483.

[2] S. Medvedev, "Machine Learning Classification with Python for Direct Marketing," 2019. [Online]. Available: https://towardsdatascience.com/machine-learning-classification-with-python-for-direct-marketing-2da27906ddac.

[3] T. Wood, "Precision and Recall." [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall.

[4] Google, "Classification: Precision and Recall." [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall.

[5] "F-score." [Online]. Available: https://en.wikipedia.org/wiki/F-score.