Contents

Machine Learning Engineer Nanodegree

Capstone Proposal

Haitham Alhad Hyder

March 30th, 2019

Proposal

Domain Background

The Titanic data set is famous data set that is part of an introductory Kaggle competition.

> **Citation:** Kaggle. (n.d.). Titanic: Machine Learning from Disaster. Retrieved from https://www.kaggle.com/c/titanic/overview

It is about the infamous RMS Titanic, an "unsinkable" ship that made the headlines on April 15, 1912. Out of the 2224 passengers and crew, only 1502 of them survived. Although it comes down to chance if a person survives, the data shows that some groups of people have a higher chance of surviving. To solve the problem of identifying those who have a higher chance of survival we will have to come up with a machine learning model that can gives us the probability of someone surviving or not.

I am motivated to finish this project, since I want to get involved with Kaggle. A journey of a thousand miles starts with a single step and also begins at the door.

Problem Statement

What we want our Machine learning model to identify is the pattern that yields the survival of a passenger or crew member. The output of our model would be a probability of survival and therefore a possible solution would be building a logistic model.

To quantify the problem we would have a test data set that would help us evaluate how well our model is doing by looking at the precision and recall numbers of our model.

Data sets and Inputs

The data set is obtained from Kaggle's Titanic competition. We will only make use of the train data set since we don't know the labels of the test data set and we want to be able to evaluate our models ourselves.

> **Citation:** Kaggle. (n.d.). Titanic: Machine Learning from Disaster. Retrieved from https://www.kaggle.com/c/titanic/overview

The data contains information on the amount of fair someone paid, the number of parents or children involved with them as well as other characteristics. The labels are only provided in the *train* data set.

The different characteristics of the people on board will help us generate a model that identifies that patterns that increase chances of survival.

Finally, when we deploy the model, we can send in characteristics of a person and get their chance of survival.

Solution Statement

The solution that we will implement are two different models, an XGBoost logistic model as well as a PyTorch neural network. Since our expected output is binary then getting a probability between 0 and 1 can easily be rounded of to a binary value.

In the deployed web app that we can input characteristics of a person, the output would not be binary but a probability of survival.

Benchmark Model

The benchmark model to use would be a statistical model. One would be a SkLearn decision tree model. The decision tree would not lead to a probability but would give us the binary values if a person survives or not. We can therefore compare this model with our proposed solution by rounding of the probabilities returned by the solution model. By comparing the rate of false positives and false negatives, we will be able to identify which model does a better job.

Evaluation Metrics

The main quantity that can help us evaluate our models is the accuracy of the models. Unlike in fraud detection, the rate of false positives and false negatives is not as important but it also does help us in picking a better model. Although, accuracy is the best metric in this situation.

Accuracy is the percentage of values the model labelled correctly.

Project Design

First, taking a look at the data is important and getting to know the features. Some of the columns are not as important in our prediction such as the id number of a person and therefore, we can avoid using such columns in our model.

To aid in choosing important features we can find the correlations of the numerical columns and identify those that are correlated. If the correlation is high, we can avoid using such columns as well.

Next, we will need to clean our data to avoid having Null values.

Finally, we build our models and deploy them for testing. After we pick the best model, we shut down and delete the others.

Next, deploying our model with an API endpoint that communicates with a webpage by sending out the probability of a person surviving upon receiving the characteristics of a person.
