Machine Learning Engineer Nanodegree

Capstone Proposal

Manish Kumar
October 10th, 2019

Proposal

Domain Background

Movie industry has many different facets that makes this industry unique in itself. While doing internet searches on this topic I came across this article published in *nytimes.com [Ref.1]* says that (during 2008 recession) when most of the industries are struggling for survival, movie industry has been witnessing unprecedented box office surge. The article cites movie going as an escape from other troubles that is going on in peoples life.

One more important aspects is that movie industry is very money intensive. A lot of money is at stake from the creation till the box office release, hence it becomes a subject of study and research to see what are the factors that makes a movies successful or failure. A paper published by Simonoff & Sparrow *[Ref.2]* discusses various predictor variables that can affect the gross revenue of a movie e.g Genre of the film, MPAA(Motion Picture Association of America) rating, origin country of the movie, star cast of the movie, production budget of the movie etc.

On personal front being a movie buff this problem domain becomes natural choice for me. It always fascinates me to know stats about movies and intrigue me to know how they are correlated and can be used to predict future trends. Also, I investigated a similar TMDB dataset as part of my Data Analyst Nanodegree and learnt a lot by doing that project and also made curious to apply machine learning concepts on these datasets and if possible to predict values accurately. These some of the points concludes my personal motivation for choosing this particular problem domain.

Problem Statement

This project problem statement is part of the *kaggle competition(Kaggle.com)*. In this competition, we're presented with metadata on over 7,000 past films from The Movie Database to try and predict their overall worldwide box office revenue. Data points provided include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries *[Ref.3]*.

Datasets and Inputs

This dataset has been collected from TMDB. In this dataset, we are provided with 7398 movies and a variety of metadata obtained from The Movie Database (TMDB). Movies are labeled with id. Data points include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries.We are predicting the worldwide revenue for 4398 movies in the test file *[Ref.3]*.

Solution Statement

In this project the task is to predict target variable(Revenue) using independent variables like budget, genres, popularity, runtime, release date, cast, crew etc. This problem comes under supervised learning as we have labeled data.

Further, This Supervised learning problem is about predicting the target (variable) so I need to build a regression analysis model. I need to train the model on the training data, then testing the model performance using test data and finally predicting the revenue for unknown data. In Scikit-learn(The package I chosen to use) package provides many regression algorithms. By taking into account the nature of data I decided to use SGDregressor algoirithm for this problem.

Benchmark Model

For benchmark model, I will use one of the simplest linear regression algorithm. This benchmark model will work as beseline for the problem.

```
from sklearn.linear_model import LinearRegression
reg= LinearRegression().fit(X,y) (where X is feature variables and y is target var
iable)
reg.Score(X,y)
```

Evaluation Metrics

I chosen R-squared score as the evaluation metrics of this project(a regression analysis). R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination. R-squared value always lies between 0 and 1 *[Ref.4]*.

The most general definition of the coefficient of determination is

$$(1 - \frac{SS_{res}}{SS_{tot}})$$

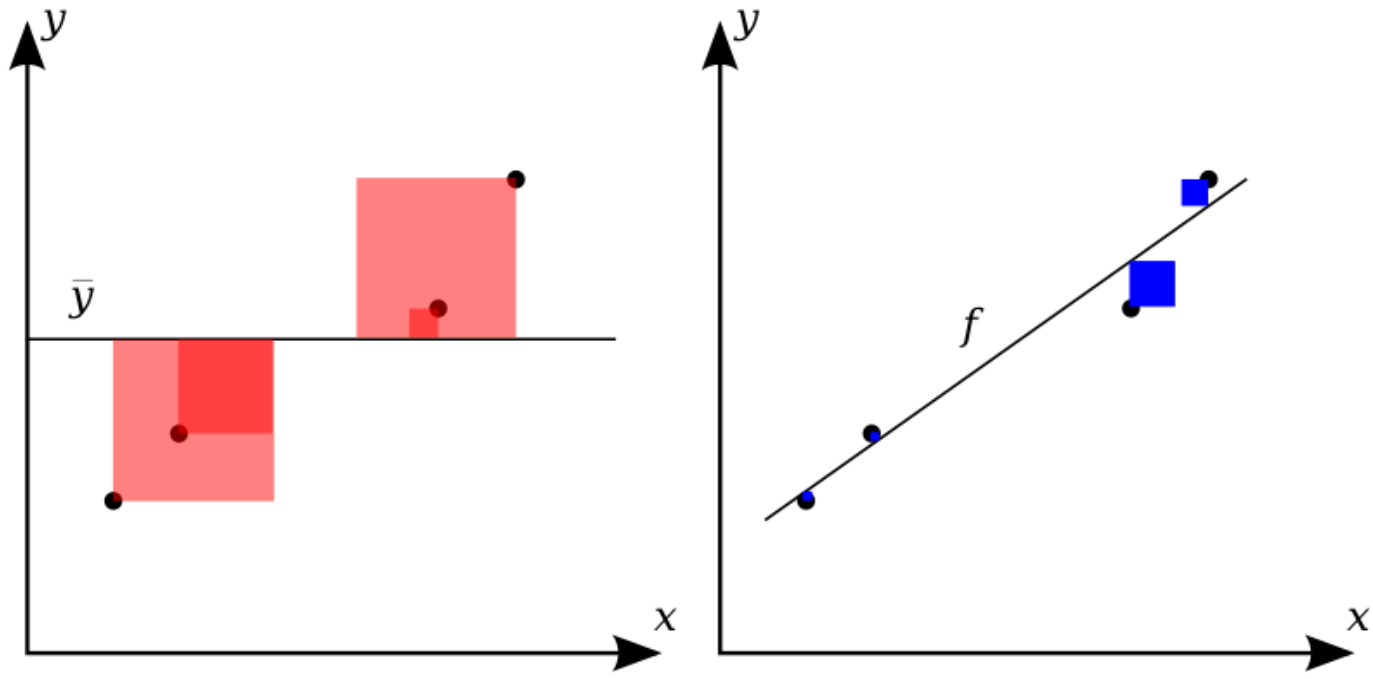
, where

$$SS_{tot} = \sum_{i=0}^{n} (y_i - y^{-})^2$$

is total sum of squares(proportional to the variance of the data) and

$$SS_{res} = \sum_{i=0}^{n} (y_i - f_i)^2$$

is the sum of squares of residuals *[Ref.5]*.



(By Orzetto - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=11398293 (https://commons.wikimedia.org/w/index.php?curid=11398293))

A model with R-squared value of 0 means that the model is behaving like a naive predictor model(predict average value all the time) and definitely not a good one. However, a value of 1 indicated the model is predicting the target variable accurately. Keeping this criteria in mind my aim in this project to build a regression model with a R-squared score close to 1.

Project Design

My workflow for this project comprises following steps:

1. Data Exploration:

This very first step towards the model building. To Know the data in depth is the key of any model building process. In this particular step I will import the raw datafile and an exploratory analysis will be done using pandas libraries and other visualization techniques. The main purpose of this step is to find out missing values, duplicates, extraneous columns(e.g. id, homepage, original_title, overview etc) and correlation between different columns (This help in identifying important features of the given dataset). Identifying numeric and non-numeric columns is also an important task in this step.

2. Data Preprocessing:

This step is to prepare the data for the training of the model. Generally, Supervised machine learning algorithms are most suited to work with numeric data. Hence, I need to convert some columns(e.g genres) using one hot encoding method. I also need to fix other anomalies discovered during the data exploration step. Once all these things are done my data will be ready to feed to the model for the training and testing.

3. Initial Model Implementation:

For this project to solve the underlying problem I need to build a regression model. The software package I am using is Scikit-learn as it provides a vast varieties of machine learning algorithms. For benchmark model, I will use one of the simplest linear regression algorithm. This benchmark model will work as beseline for the problem.

```
from sklearn.linear_model import LinearRegression
reg= LinearRegression().fit(X,y) (where X is feature variables and y is target var
iable)
```

As for the proposed solution for this project I chosen SGDregressor(Optimization of Linear Regression) algorithm seeing the nature of the dataset.

```
from sklearn import linear_model
reg=linear_model.SGDRegressor(max_iter=500)
reg.fit(X,y)
```

4. Model Evaluation:

As discussed in the previous section, I have chosen R-squared score as the evaluation metric for the model that I developed in the previous step. In sklearn, we have libraries using which r2_score can be calculated very easily.

```
from sklearn.metrics import r2_score
score=r2_score(y_true,y_predict)
```

By comparing this score value I can evaluate the model and further hyper parameter tuning can be done to improve the r2_score of the model. This parameter tuning we are going to do in the section.

5. Model Tuning:

Hyperparameter tuning is one of the important aspects of model building process. Gradient descent is an optimization method to minimize the cost function. Here cost function is

$$SS_{res} = \sum_{i=0}^{n} (y_i - f_i)^2$$

. We can use gradient descent to find the values of the model's parameter that minimizes the value of the cost function.

An important hyperparameter of gradient descent is the learning rate. If the learning rate is small enough, the cost function will decrease with iteration until gradient descent has converged on the optimal parameters.Stochastic Gradient Descent(SGD) ia a variant of gradient descent algorithm. In scikit-learn SGDregressor is an implementation of SGD can be used even for regression problems with large number of features *[Ref.6]*.

Other important hyperparameter in SGDregressor are loss function, max_iter, epsilon, tol(stopping criteria) etc. These hyperparmaters need to be tuned to get the best performance that built in previous section.

6. Final Model Evaluation:

This step is evaluation of the final model that I developed after hyperparamter tuning. This is where I am supposed to feed some unknown data to the model and see the output.

References:

[1] In Downturn, Americans Flock to the Movies (Feb 28, 2009) (https://www.nytimes.com/2009/03/01/movies/01films.html (https://www.nytimes.com/2009/03/01/movies/01films.html))

[2] Simonoff, Jeffrey S. and Sparrow, Ilana R., Predicting movie grosses: Winners and losers, blockbusters and sleepers, 1999 (https://archive.nyu.edu/handle/2451/14752 (https://archive.nyu.edu/handle/2451/14752))

[3] TMDB Box Office Prediction, Can you predict a movie's worldwide box office revenue? (https://www.kaggle.com/c/tmdb-box-office-prediction (https://www.kaggle.com/c/tmdb-box-office-prediction))

[4] Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?, 30 may 2013, (https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit (https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit))

[5] https://en.wikipedia.org/wiki/Coefficient_of_determination (https://en.wikipedia.org/wiki/Coefficient_of_determination)

[6] Mastering Machine Learning with scikit-learn, 2014, Gavin Hackeling.