ADVERTISEMENT CLICK THROUGH RATE

PROJECT REPORT

REPORT

1. Project definition

Predicting ad click-through rates is a massive scale machine learning problem that is central to the multi-billion dollar online advertising industry. The project was inspired by google AdSense. Google AdSense is a program run by Google that allows publishers in the Google Network of content sites to serve automatic text, image, video, or interactive media advertisements, that are targeted to site content and audience. They can generate revenue on either a per-click or per-impression basis. In this project I tried to create a model which try to predict the probability whether an ad will be clicked or not based on different features.

Online advertising is a multi-billion dollar industry that has served as one of the great success stories for machine learning. Sponsored advertising have relied heavily on the ability of learned models to predict ad click-through rates accurately. In this project I am planning to build a supervised learning model to find the probability whether an advertisement will be clicked. Here I will use classification (probabilistic classification) and will use predict_prob function (this is an additional method in the classifier which I will be using in order to predict the probability) so as to predict the probability whether an advertisement will be clicked or not. The inputs variables are mentioned below in the datasets and input section. The final model is expected to be useful for companies so that they could make necessary changes so as to make their advertisement useful.

The metrics used here is ROC-AUC score. ROC stands for receiver operating characteristic and AUC stands for area under the curve. ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary

classifier system (which in our case is click or not-click) as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. Here we are interested in finding the score so we use roc_auc_score function to calculate the area under the receiver operating characteristic curve. The area under the curve (AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative example. It measures the classifiers skill in **ranking** a set of patterns according to the degree to which they belong to the positive class, but without actually assigning patterns to classes so by computing the area under the roc curve, the curve information is summarized in one number.

2. Analysis

The dataset and inputs were obtained from hacker earth platform where the same problem was listed as a challenge. The dataset here we have is too large. It has about 10 million records which is not possible for me to train due to lack of resources so here I have just restructured the data for a single merchant that means now the record just have the data's of a single merchant. So I have basically done the feature selection of the merchant which reduced my dataset to an amount which is considerably smaller and enough for my machine to train

The following variables are given in the dataset:-

Variable	Description
ID	Unique ID
datetime	timestamp
siteid	website id
offerid	offer id(commission based offers)

merchant	merchant id’s
category	offer category
countrycode	country International code
browserid	browser used
devid	device used
click	target variable

I have given some descriptive statistics which describes our dataset to great extent. So what I have done here is taken a snip of the notebook of the statistics of the features.

	siteid	offerid	category	countrycode	browserid	devid	tweekday	thour
count	29920.000000	29920.000000	29920.0	29920.000000	29920.000000	29920.000000	29920.000000	29920.000000
mean	1590.506852	1251.756049	0.0	1.624866	3.529980	1.486731	2.785027	12.682587
std	1125.697155	739.255387	0.0	1.621032	2.920468	1.133531	1.750844	3.779531
min	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000
25%	588.000000	646.000000	0.0	1.000000	1.000000	1.000000	1.000000	10.000000
50%	1491.000000	1233.000000	0.0	1.000000	2.000000	1.000000	3.000000	13.000000
75%	2551.250000	1903.000000	0.0	3.000000	6.000000	3.000000	4.000000	16.000000
max	3757.000000	2528.000000	0.0	5.000000	11.000000	3.000000	6.000000	23.000000

Here in the variable datetime we have the times based in the month of January and the year 2017 so what I have done is converted this to a format readable by pandas and then extracted the weekday (Sunday=0, Monday=1 etc.) and the hrs (24 hr clock) as these data’s are only the ones which are relevant to us in order to train our model. I have also found that there are some variable which have blank data per record so I have just filled in those gaps with -999 for integer data types and “None” for the string datatypes. After doing all of these I splitted my dataset into features and target. Features contained all the variables except click (which now becomes our target variable), Id’s (As these

are unique), datetime (As we extracted the weekday and the hrs based on 24hr clock from these). Target just contains the click variable which has 0 for not click and 1 for click. After doing the splitting of dataset into features and target we check for skewness in our dataset columns which is done by using pandas skew method. Also an exploratory visualization is provided our dataset which is done by plotting the histograms of the different variables which confirms that our data is not skewed and there is no need for it to normalize.

I am using supervised learning for this model. I am treating it as probabilistic classification problem since we want to know the probability as to whether the advertisement will be clicked or not so for that I am using predict_proba function which gives the probabilities that whether an ad will be clicked or not. The algorithm used here is Logistic regression (As a classification model). **Logistic regression** estimates the probability that a characteristic is present (e.g. estimate probability of "success") given the values of explanatory variables, in this case a single categorical variable so to use that is best for this problem. Also the predict_proba function which will be used to estimate the probabilities is a part of logistic regression algorithm in sklearn. Just for information I also tried Decision tree models and SVM for this but that didn't worked out.

I tried to compare my model with the scores of the leaders of the challenge from where I have picked up this dataset but since I have reduced my dataset to a considerably smaller amount so the score are little bit higher so I guess I have done a good job in training the dataset.

3. Methodology

Here the dataset we have used is too large which is not possible for me to train due to lack of resources so what is that I have done some refinement and have only chosen the records of a particular merchant which eventually makes our dataset smaller. Here in the variable datetime we have the times based in the month of January and the year 2017 so what I have done is converted this to a format readable by pandas and then extracted the weekday (Sunday=0, Monday=1 etc.) and the hrs (24 hr clock) as these data's are only the ones which are relevant to us in order to train our model. I have also found that there are some variable which have blank data per record so I have just filled in those

gaps with -999 for integer data types and “None” for the string datatypes. After doing all of these I splitted my dataset into features and target. Features contained all the variables except click (which now becomes our target variable), Id’s (As these are unique), datetime (As we extracted the weekday and the hrs based on 24hr clock from these). Target just contains the click variable which has 0 for not click and 1 for click.

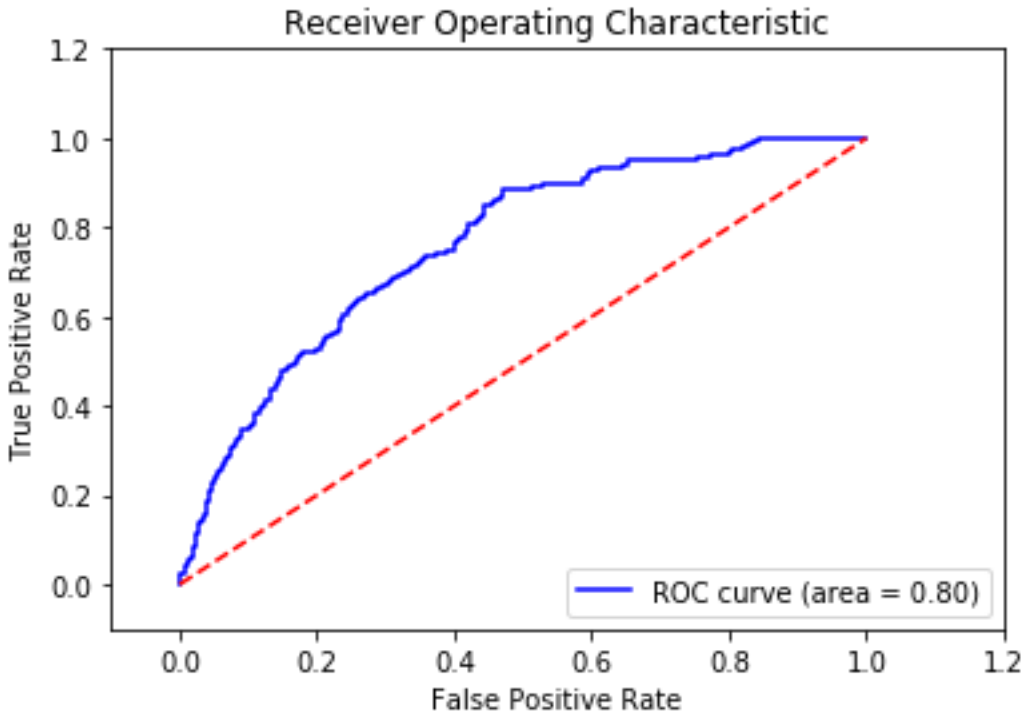
Initially I was using the whole dataset but that did not worked well as it was too large so when I was training that it was taking too much of time which eventually hanged my system and in the end what I got was nothing. Then I tried to do mini-batch training on the dataset but that also did not worked which eventually led the way to some refinements that’s why I choose only the records of a particular merchant and that worked well The only algorithm that was working well for smaller dataset is logistic regression as my model wants to find out the probability that whether an advertisement will be clicked or not. Others were just overfitting. Also initially I thought the metrics that will work well with this was log-loss metric but the benchmark was based on roc-auc score so I switched to roc-auc score as my metrics.

My model was giving 0.76 as the roc-auc score which after refinement and optimization gave me the score of 0.79.For doing that I used grid search cross validation as the technique to optimize the model. The parameter that was cross validated was C. The trade-off parameter of logistic regression that determines the strength of the regularization is called C, and higher values of C correspond to less regularization (where we can specify the regularization function).C is actually the Inverse of regularization strength (lambda). Here smaller values specify stronger regularization.

4. Results

This problem is a supervised learning model. The algorithm which was used here was logistic regression (as a classification model). After fitting the model I used predict_proba function as I wanted to evaluate the probability that whether an advertisement will be clicked or not and then I used roc-auc score to When

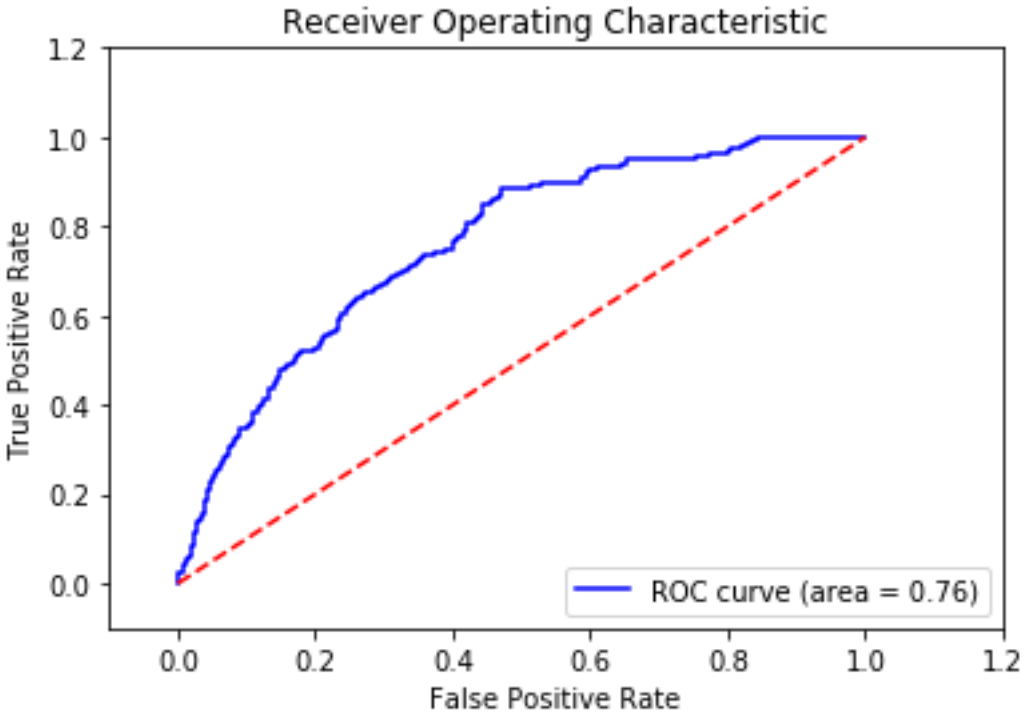
I compared that with my benchmark model I thought it to be good as since I am using the refined version of the dataset and so I was getting the score higher than the leaders of the competition. Then I performed grid search cross validation on my data and found out the score which was higher than the unoptimized model. The gird-search model was giving me a score of 0.79 which was good. The parameters that were tuned was C parameter which determines the strength of the regularization. Thus the accuracy increases which can also be justified from this graph.



The final model got a score of 0.80 (approx.) which was trained for a smaller dataset particularly for a single merchant but the benchmark model had a highest score of 0.69 but then they are training on a large dataset and as we always know the score tends to decrease on a large dataset so this justifies the solution proposed by me for this particular problem.

5. Conclusion

I have used roc-auc score as a metrics so as to evaluate my model. I this we plot a graph between false positive rate and true positive rate and we find the area under the graph. The graph which we got is shown here.



We can see here the blue curve is the Roc curve and the area which we got is the score.

So using the supervised learning model and the algorithm of logistic regression as described earlier I have achieved a great success in building up the model with a great accuracy. Although the size of the original dataset was too large which was not possible for me to train because of lack of resources so I have refined my dataset for a particular merchant which makes my dataset considerably smaller so as it now becomes good for my machine to train. The

size was that much big that even mini-batch training also failed to train this model.

Even though the model gave me promising results but in future I would try to implement my project on full dataset using the concepts of big data. Also I would like to attempt building this model using cloud services such as amazon web services or something like that. Also I found out something like lightgbm as an algorithm to build such kind of problems but I

6. Reference and citations

The dataset of this project was taken from HackerEarth platform.

https://www.hackerearth.com/challenge/competitive/machine-learningchallenge-3/problems/

The link to download the dataset is https://he-s3.s3.amazonaws.com/media/hackathon/machine-learningchallenge-3/predict-ad-clicks/205e1808-6-dataset.zip

The idea for this project was inspired by Google AdSense and necessary citations for this is referenced from this paper. https://www.eecs.tufts.edu/~dsculley/papers/ad-click-prediction.pdf

Different kaggle competitions were also taken a look at for different ideas related to the model.

https://www.kaggle.com/c/avazu-ctr-prediction
https://www.kaggle.com/c/avito-context-ad-clicks