**VYSHNAV M T**

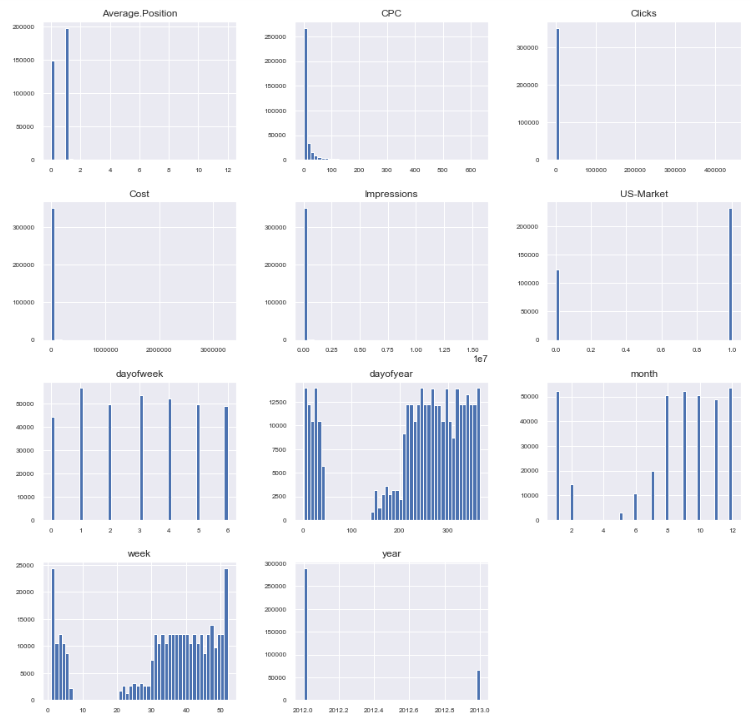
Email id: [**Vyshnav94.mec@gmail.com**](mailto:Vyshnav94.mec@gmail.com)

Skype id: **vyshnav94.mec**

**Technical Evaluation task**

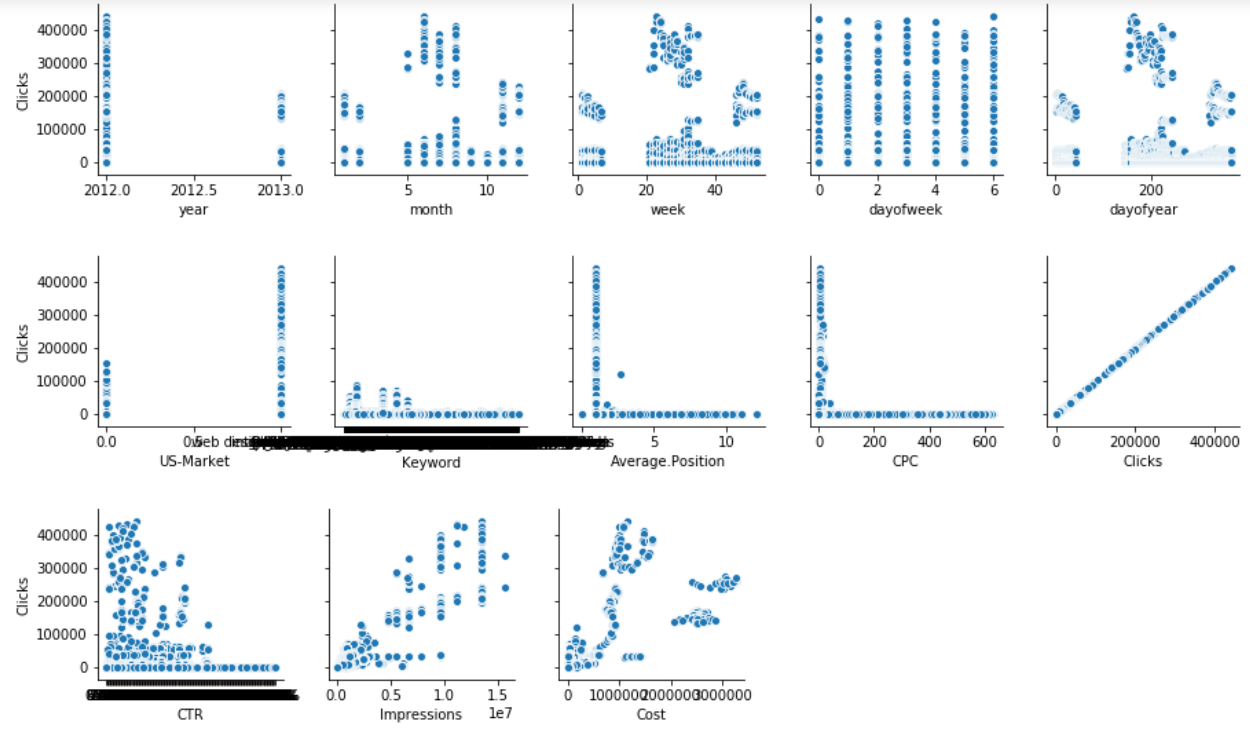
**Questions:**

1. **How did you manipulate the data, and why? Illustrate your answer with plots.**

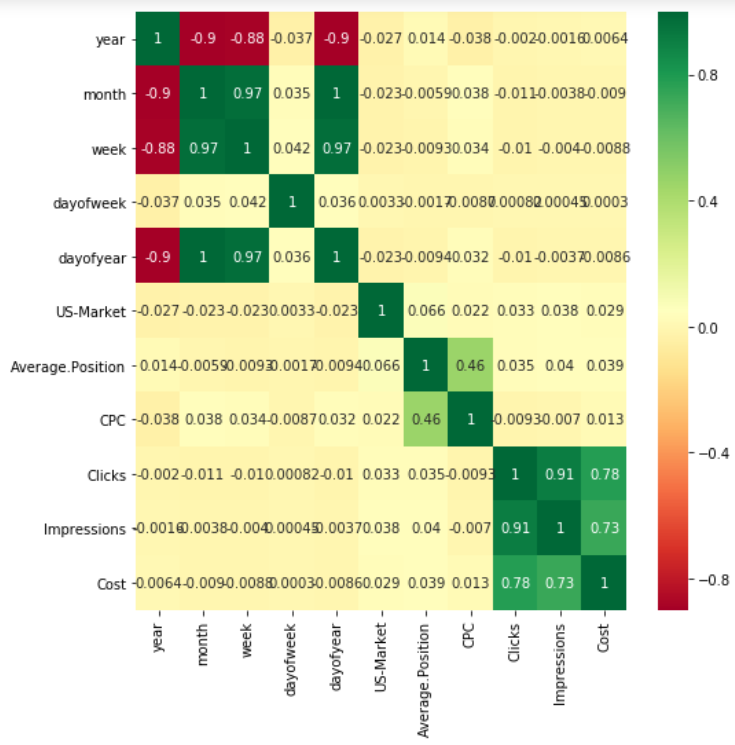


Initially I have used **describe method** to have intuition about number of non-missing values, mean, standard deviation, range, median, 0.25 and 0.75 quartiles. Then I plotted the **histogram plot** of each independent features to get the distribution of each features. It was observed that number of clicks = ‘0’ were more when compared to other numbers. Further, clicks were more in the year 2012 than in 2013. Also when considered month wise, most of the clicks were during months of August – December. If we see day of week, the highest clicks have happened on Tuesdays. And most of the clicks were from the US market.

Now to identify the feature importance of the numerical features, I have done a pair plot between the target variable=’clicks’ and all other features. It was observed that features such as Cost, Impressions, CTR, CPC, Week, day of year has some linear relationship with the target variable.



Inorder to further verify the imporatant features, I have plotted a correlation matrix with heatmap. It is observed that only two of the features impressions and cost were highly correlated with target variable. We can also see strong negative correlations in the plotted heatmap.



1. **How did you perform NLP, if any?**

**Text preprocessing/cleaning was done on the Keywords:**

Initially all numbers, punctuation marks, symbols were removed from the keywords using regular expressions. Then the keywords were lowercased and tokenized. After tokenizing the keywords, stop words were removed using NLTK stop words.

**Word Embedding’s applied on the cleaned keywords:**

After cleaning the data, each words in the keyword needs to be converted into word vectors, in order for the machine learning model to be trained, as algorithms cannot process the plain text or strings in its raw form. Word embedding’s are vectors that captures the semantic and contextual information of words. Here I have used a Fasttext based word embedding which gave vector of dimension 300 for each words in the keyword. Thus to generate a single vector for complete keyword, average of each word vectors in a keyword were taken.

1. **How did you model the problem, and why?**

In the given problem, the number of clicks is the target variable which is to be predicted by the trained model, when Date, market, keywords and CPC were given features. Since the number of clicks have positive continuous values, I have modelled the problem as a regression problem. There were many ‘Nan’ values present in the data, which were dropped instead of doing some imputations on the data and replacing Nan values. Because we all of the features were Nan for some particular keywords and so we are not sure how many clicks, impression or cost could be for these keywords. Aslo, since the data is huge, it doesn’t affect even if we drop these fewer data’s. Now, since date on which the keyword has been searched could have important features, I have extracted features like year, month, week, dayofweek, and dayofyear from the date appended to the data. Categorical values were converted to dummy variables.

As per the problem statement, it was mentioned that given keywords and CPC for each market (US/UK) at the date of 14/2/2013, model has to predict the number of clicks. So I have modelled the problem by taking two different sets of features:

1. X = keywords, market, and cpc
2. X = year, month, week, dayofweek, dayofyear, keywords, market, and cpc

Target variable was set as Y=number of clicks.

I have used classical machine learning and deep learning model such **as Random forest regression, Xgboost regression, 1D-CNN and GRU** models to train and evaluate the given data. The given data is split into train and test data by sorting it by date. All data from year 2012 were taken as train data and all data from year 2013 were taken as test data.

1. **How did you evaluate your model? What were the results of the evaluation?**

To evaluate the model I have used metrics such as Mean absolute error, Mean squared error and R2 score, where R-squared measures the goodness of the fit of the regression model. It basically gives percentage the variance in the dependent variable that independent variables explain collectively.

Results for two set of features:

1. X = keywords, market, and cpc

**Random forest Regression:**

Coefficient of determination R^2 of the prediction: 0.914177804049537

Mean squared error: 297606.09

Test Variance score: 0.80

mean\_absolute\_error: 74.95

**XGboost Regression:**

Coefficient of determination R^2 of the prediction. 0.5800091699348091

Mean squared error: 393330.94

Test Variance score: 0.74

mean\_absolute\_error: 213.09

**1-DCNN: (70 epochs)**

Mean squared error: 183220.71

Test Variance score: 0.88

mean\_absolute\_error: 61.83

**GRU: (20 epochs)**

Mean squared error: 125502.27

R2 score: 0.92

mean\_absolute\_error: 70.50

1. X = year, month, week, dayofweek, dayofyear, keywords, market, and cpc

**Random forest Regression:**

coefficient of determination R^2 of the prediction.: 0.9976723650491951

Mean squared error: 1008241.56

Test Variance score: 0.33

mean\_absolute\_error: 175.60

**XGboost Regression:**

coefficient of determination R^2 of the prediction.: 0.6653407930753578

Mean squared error: 519430.67

Test Variance score: 0.65

mean\_absolute\_error: 224.09

**1-DCNN: (30 epochs)**

Mean squared error: 9360270.80

R2 score: -5.25

mean\_absolute\_error: 2336.26

**GRU: (30 epochs)**

Mean squared error: 1535167.11

R2 score: -0.02

mean\_absolute\_error: 193.50

1. **If you had extra time, what would you do next?**

I would have performed some hyper parameter tuning on the trained models to get good results. Also, I would have created models using other pre-trained word embedding algorithms such as Bert, Google sentence encoder and Google universal encoder models. These are models trained on huge corpus of multilingual data and uses character level encoding. The advantage of using these kind of models is that, even if new keywords comes in the test data, these models will be able to give word embedding’s. Further I would have also tried some other machine learning and deep learning algorithms to get better results.