<u>Stanza coding challenges writeup</u>

Steps which I performed to decide the best method:

1) Reading the data in pandas dataframe
2) Created another column ‘cpm’ which is calculated as revenue/impressions*1000.
3) Converted data column to datetime and made it index of the dataframe.
4) Performed EDA.
   I Calculated total number of websites, rows.
   II Calculated mean and standard deviation of CPMs for all the websites.
   III Calculated and compared ranges of CPM for all the websites.
   IV Visualized histograms for some of the websites to check if CPM is normally distributed.
   V Count of days when impressions are made for all the websites.
   VI Checking for seasonality. Checked for various periods like week, months and quarters.

My Initial thought:

1) Setting some benchmark at the lower and upper side. If the value goes below the benchmark, it is underperforming and overperforming if it goes above upper limit

In this case, I was thinking of keeping the same limits for the whole data
The possible limits could be:
1) Mean+/-2 SD
2) Mean+/-1 SD
3) Median+/-1/2 SD
4) Some quartile +/- n SD

Conclusions after EDA:

1) The dataset contains data of 173 websites and more than 17000 rows for all the websites.

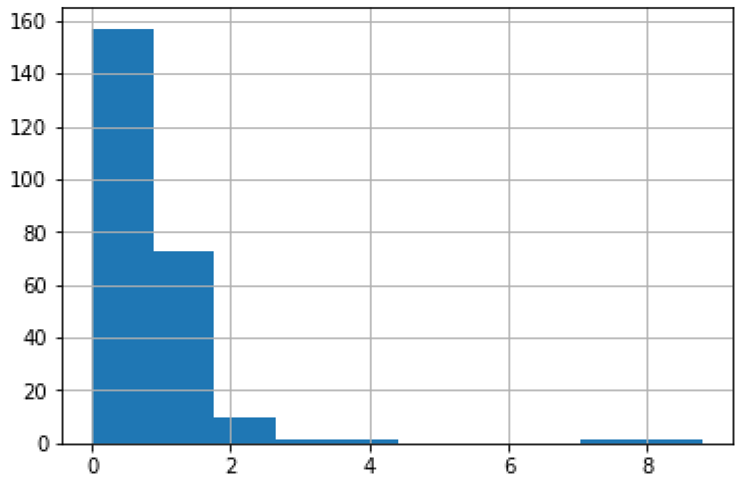
2) Mean is varying a lot for all the websites. We need to consider every website independently. We can't use same benchmark for all the websites

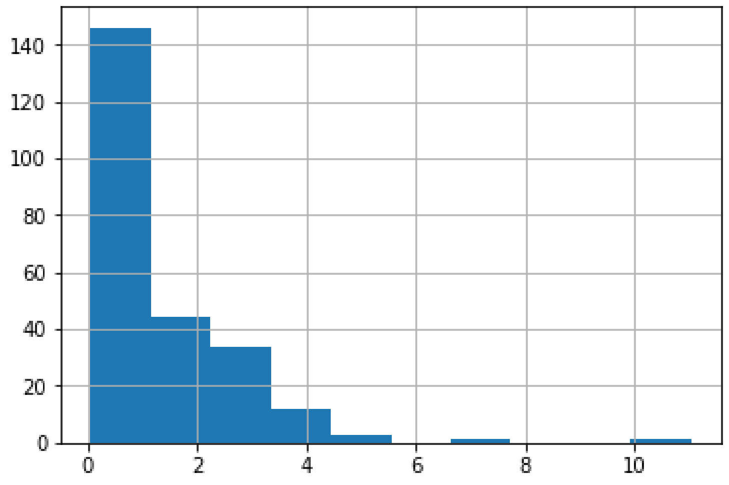
Mean+/-2SD is the most preferred benchmark. If the data is normally distributed, this range will cover more than 95% of data and only 2% of data from both the ends will be taken as an anomaly

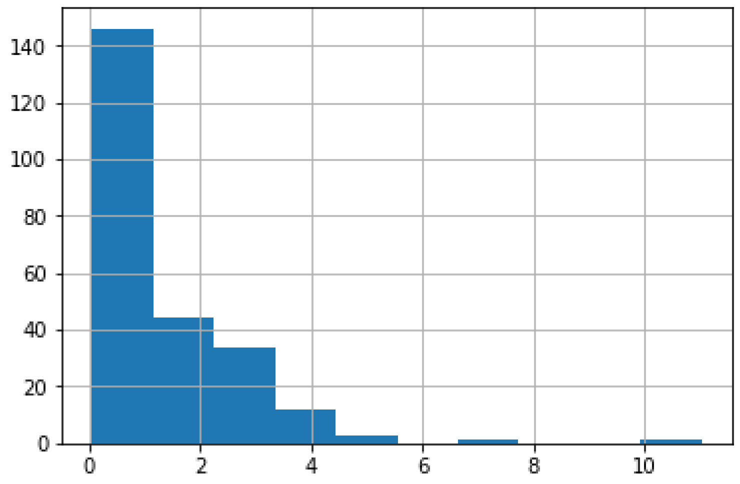
3)

After visualizing histograms for 3 different websites, we can conclude that data is not normally distributed. At least for some of the sites. So we can't use this generic method of standard deviation.

And so 2 SD is a bad idea for benchmark.







4) Also, after reviewing SD for the websites indivitually, I found out that SD is sometimes higher or almost equal to the mean. So, using -SD on the lower side will set -ve benchmark and we wont be able to filter out low abnormalities.

5) Also the websites have variety of number of entires in the entire year which made me decide that the same method will also not work for all the websites. Need to check if we can categorize websites based on their density and ranges and then set same method based on their categories.

6) The CPM also has wide ranges for some websites and very narrow ranges for some of the websites. I was thinking of categorizing the websites based on their counts and then setting same benchmarks for all the websites in the same category. But, that won't work either and I will have to treat every website differently.

7) It is observed that, CPM is highest on saturday or sunday for most of the websites. So, we can conclude that there is seasonality in the weeks.

8) CPM doesn't have any trend in the months. Also the variety is there because number of observations are very diverse along the months

9) On the website level, I observed that mean CPM is lower for the months january, july, april and october for some of the websites. (the same thing is stated in the problem statement as well) So, I think there is seasonality by the quarters.

10) So I concluded that the data has weekly seasonality and statistical approach won't work here.

Thoughts on possible methods:

Now, I had multiple options to find anomalies in the data.
1) Classification
2) Clustering
3) Time series

We can use single class classification in this case. Possible algorithms could be SVM, logistic regression, artificial neural networks which can provide non-linear boundaries on the normalized data. But, in this case, labels aren't provided and manually labeling 17000 entries is not feasible option.

We can use k-means clustering with k=2 or 3. Now, 3 clusters will be formed. The points which are present at the extreme ends and far from the centroid of normal(non-anomalous) datapoints will be detected and clustered differently. But, clustering algorithm won't detect the seasonalities. For example, at the start of every quarter, CPM tends to be low. But, clustering won't consider this fact and still cluster these low CPM points as the anomaly.

To consider the seasonality, best way is use time-series analysis/forecasting.

Possible options were:
1) AR
2) MA
3) ARMA
4) ARIMA
5) SARIMA
6) SARIMAX
6) TBATS

As we have concluded our model contains at least 1 seasonality (weekly) and so we will have to convert out our non-stationary data to stationary time series. Also these models won't handle seasonal data well. So, the remaining options are TBATS, SARIMA, SARIMAX

BATS model is Exponential Smoothing Method + Box-Cox Transformation + ARMA model for residuals. The Box-Cox Transformation here is for dealing with non-linear data and ARMA model for residuals can de-correlated the time series data. Alysha M.(2010) has proved that BATS model can improve the prediction performance compared to the simple Sate Space Model. However, BATS model does not do well when the the seaonality is complex and high frequency. So, Alysha M.(2011) propsed TBATS model which is BATS model + Trigonometric Seasonal. The trigonometric expression of seasonality terms can not only dramatically reduced the parameters of model when the frequencies of seaonalities are high but also give the model more flexibility to deal with complex seasonality. In a nutshell, this is how the story goes: Exponential Smoothing Method ⇒ State Space Model ⇒ BATS ⇒ TBATS.

1) TBATS:

I played around the data for seasonality terms,(tried s1=7 and s2=15,30,60,90). s2=90 gave the closed prediction for 2-3 websites. However, after reviewing the results I realized that the data isn't enough and not even 3 seasons for quarterly seasonality is present to detect seasonalities and predict multiple outputs. As, for most of the websites values are not present and so TBATS isn't working properly. AIC is also very high.

2) SARIMAX:

Now lets consider SARIMA. This model works well with single seasonality but can't handle multiple seasonalities. So, I had to consider SARIMAX. X denotes exogenous term. So, in terms for SARIMAX, multiple seasonalities are handled using exogenous dataset. So, one of the seasonality is handled by exogenous dataset using fourier transformation. Another seasonality is handled by SARIMA model.
After playing around multiple websites by varying number of fourier terms (upto 5), I observed that the model gave best predictions for some websites with 2 fourier terms.

Data before 04/01 is used for fitting the model. Data after that is used to test the model. (The data transformation part is excluded from the notebook to avoid confusion)
In this case, I predicted the future outcomes for test data. The future outcomes are expected CPM values considering seasonalities and trends in the CPM data over a year (actually dataset contains values less than the year). If the residuals between actual and expected values are larger than some benchmark, then for that day, data will be said abnormal.

Actual operations:
Lets do line by line analysis of the code Iterated over the dictionary which has site by site dataframes:

1) Filled the dates for which data isn't present by 0.

2) Divided the data for site k in train and test set. Data before 04/01 is used to train the model. The later data of 2 months is used to test the model fitted for the website.

3) If less than 20%(also tried different values here, selecting higher number here had rejected more websites and lesser number might lead to bad predictions. So tried to select the value which balanced the trade-off between accuracy and data) of the data contains the actual CPM values in the trainset or no data is present in the test set. That website isn't considered for the prediction, as insufficient data will lead to false negatives.

4) Residuals are calculated for the predicted CPM using SARIMAX. If the negative residual is smaller than some threshold, that entry is considered abnormally low and if positive residual is larger than some threshold, the entry is considered as abnormally high. Residuals are calculated by taking difference of actual value and expected(predicted) value.

5) After playing around different parameters and values, standard deviation alarmed higher anomalies the best. For lower anomalies the same threshold value was giving some false alarms and so threshold is increased to 2.5 times of standard deviation of the residual values. If the value in the test set is 0 (No revenue is generated for the day), residual is ignored. Otherwise the model will send false alarms of lower abnormality in that case as well.

6) In some cases, predicted values are negative (might be because too many values are missing in the train set and model is weak for those websites).

7) The values which are recorded by the system are writen in the file. In this case, different model is used for all the websites and the same benchmark term is used which generalized almost all websites.

Prediction can be bad as less than 3 quarterly seasons were present to fit the model. It can be improved if more data was present and more seasonalities could also be considered/detected. False negatives are avoided by setting larger threshold limit.

Output analysis:
1) After running all the cells, it will output a file ‘notification.csv’ which will give analysis of the alerts given for every website along with date, actual CPM and expected CPM. The percentage drop/growth will help you justify why the particular alert is required.

2) hoosierhuddle is giving most number of lower abnormalities. It is sign of probably weak model. For 32 values, the predicted value is larger by 2.5 times of SD out of total of 61 values. It is possible that in the training set CPM values are much larger than the test set observations. Can be improved by using more data or setting higher threshold. However, setting different threshold for all the websites is also not efficient.

3) dotapicker is overperforming the most during last 2 months of data. For 19 days, actual value is smaller by 1 SD of total residuals for the websites data.

Performance analysis:
Mean CPM per day is considered to calculate performance of the site in terms of revenue.

4) Top 10 sites which are overperforming in terms of average daily CPM values (days when CPM is not present are not considered here)

lakersnation	96.919101
justinbieber	40.399356
celticquicknews	33.329765
bluemoon-mcfc	19.165804
opendota	17.283611
autoweek	14.899640
fswbucs	13.051070
elophant	10.931453
videocelts	9.714749
scfc2	8.343301

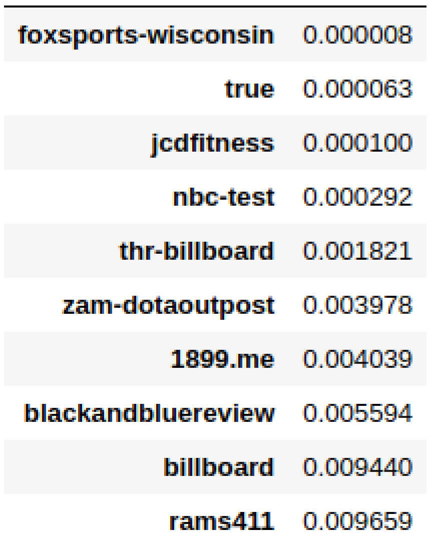
5) Top 10 sites which are underperforming in terms of average daily CPM values (days when CPM is not present are not considered here)

foxsports-wisconsin	0.000286
true	0.010187
jcdfitness	0.032279
zam-wowhead	0.035910
zam-dotaoutpost	0.038934
zam-hearthhead	0.080079
zam-esoui	0.083276
zam-csgoutpost	0.085767
1899.me	0.093188
nbc-test	0.094298

6) Top 10 sites which are overperforming in terms of average daily CPM values (days when CPM is also present are not considered here)

lakersnation	66.613128
celticquicknews	13.414456
bluemoon-mcfc	7.595117
autoweek	7.519014
videocelts	6.616857
opendota	5.779040
mynhltraderumors	5.332797
justinbieber	5.128092
dodgerblue	4.529269
dotafire	3.566299

7) Top 10 sites which are underperforming in terms of average daily CPM values (days when CPM is also present are not considered here)



foxsports-wisconsin	0.000008
true	0.000063
jcdfitness	0.000100
nbc-test	0.000292
thr-billboard	0.001821
zam-dotaoutpost	0.003978
1899.me	0.004039
blackandbluereview	0.005594
billboard	0.009440
rams411	0.009659

How to run the file:
1) Notebook file:
You will need Ipython and jupyter notebook to run the ipynb file. That file contains detailed analysis on the dataset. The step by step results of every EDA step and also descriptions and thoughts after every step. You will also need to download tbats package. You can download that using:
pip install tbats
For linux/mac.

Other required packages are:
1) pandas – for dataframe operations
2) numpy
3) matplotlib – graphs
4) pmdarima – to run SARIMAX model
Run using : *jupyter notebook*

And Browse to the folder where files are downloaded open the stanza.ipynb.
Keep notebook and input datafile in the same folder.

2) Python file:
This file doesn’t contain the analysis results. It only reads the data and builds the SARIMAX model along with the expected outputs.
Run using : *python stanza.py*