Name: Sanjay Challal

Email address: sanjay.challal@gmail.com

Contact number: +91-9538408985

Anydesk address: **437 691 393**

Years of Work Experience: **5 years**

Date: **21th Sep 2020**

Self Case Study -1: Recruit Restaurant Visitor Forecasting

"After you have completed the document, please submit it in the classroom in the pdf format."

Please check this video before you get started:
https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg

Overview

Running a thriving local restaurant isn't always as charming as first impressions appear. There are often all sorts of unexpected troubles popping up that could hurt business.

One common predicament is that restaurants need to know how many customers to expect each day to effectively purchase ingredients and schedule staff members. This forecast isn't easy to make because many unpredictable factors affect restaurant attendance, like weather and local competition. It's even harder for newer restaurants with little historical data.

Recruit Holdings has unique access to key datasets that could make automated future customer prediction possible. Specifically, Recruit Holdings owns Hot Pepper Gourmet (a restaurant review service), AirREGI (a restaurant point of sales service), and Restaurant Board (reservation log management software).

The idea here is to use the reservation and visitation data to predict the number of visitors to the restaurant for future dates. This information will help restaurants be much more efficient and allow them to focus on creating an enjoyable dining experience for their customers.

Research-Papers/Solutions/Architectures/Kernels

*** Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. it is mandatory to write a brief description about that paper. Without understanding of the resource please don't mention it***

1. Why use RMSLE over RMSE in this visitor number prediction?

RMSLE has this unique feature of penalizing underprediction compared to overprediction, which is important in this problem since we don't want the restaurants to be underprepared especially in case of small restaurants. Being over prepared has less effects on the business.

few other advantages:

- RMSLE is unaffected by the outlier values while RMSE does

- RMSLE focuses on the relational ratio difference between prediction and actual values while RMSE focuses on the difference between their magnitudes

Source: *https://medium.com/analytics-vidhya/root-mean-square-log-error-rmse-vs-rmlse-935c6cc1802a*

2. How to model time series data using the additive models.

Different additive models and their Intuition.

Source: *https://yashuseth.blog/2018/01/19/time-series-analysis-forecasting-modelling-arima/*

3. Kaggle Meetup: Recruit Restaurant Visitor Forecasting:

Overview of Problem statement, features that are useful in prediction, winner(1st & 2nd) solution features, Insights on modelling using Arima, boosting, deep learning.

Source: https://www.youtube.com/watch?v=6IlLC4M3dMo

4. 8th Place Kaggle Solution:

1. The air visit data here is resampled by day, so that missing dates are filled with zero visits.
2. Calendar information is used to find if the next/prev day of the current day is holiday. This is useful, since it can boost/lower the visitor number.
3. The store information is retrieved from https://www.kaggle.com/huntermcgushion/rrv-weather-data store which has weather data of the area.
4. Average precipitation and average temperatures from all weather stations are the two main features extracted from weather data.
5. Visitor count(target) has a lot of outliers especially around new years eve which are high count. Assuming the visit count per restaurant follow normal distribution, any value beyond 2.4 times the std deviation is capped to 2.4 times the std deviation

    I.e values > 2.4 * visitCount.std() = max(values < 2.4 * visitCount.std())

6. Day of the month is used as a feature to mark the date getting paid. Usually paid days follow with outdoor dining.

7. Exponential weighted Means (EVM) is used to find the trend in the time series data on numerical features. Here the alpha(weight) value is determined by the optimization.
8. The simple features such as, mean, median, std, variance, count, max and minimum are added.
9. lightgbm is used for modelling, achieving RMSLE 0.50775

Source *https://github.com/MaxHalford/kaggle-recruit-restaurant/blob/master/Solution.ipynb*

5. 6th place kaggle solution:

1. The dataset is from official data, and weather-data is used from another source.
2. The gap hour between reservation time and reservation made is used as a feature. This again is divided in 5 categories as gap less that 12 hrs, 12-36, 37-59, 60-85, 85+
3. The mean, median, max, min of visitors to restaurants grouped by working and non working days is taken as a feature.
4. The mean of visitor count grouped by monthly is taken as a feature.
5. Similarly, the mean of visitor count grouped by weekly
6. Temperature and precipitation from weather data is taken as a feature.
7. XGboost is used for modelling, achieving RMSLE 0.50710

Source

https://github.com/anki1909/Recruit-Restaurant-Visitor-Forecasting

First Cut Approach

*** Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** ***

*** When you are doing the basic EDA and building the First Cut Approach you should not refer any blogs or papers ***

From the given dataset, we AIR data, which has store information, reservation information, and visits information. Along with this we have HPG reservation information and store information.

Using all this we will calculate following features,

1. Merge Air visit data with store data. This way we get location information with respect to each store. Using this we deduce two more features No of restaurants in the same location, No of same genre restaurants in the same location.
2. Merge Air visit data and Air/HPG reservation data. From this we deduce two more features, No of reservations done for visit date using AIR software and HPG software.
3. Merge AIR visit data and date information. From this we deduce the trend in visit. The holiday flag has an impact on restaurant visits. We also use the day to deduce the working or non working day feature. Non working day higher trend in visits.
4. We add if next/prev day is a holiday as a feature. This also has a significant impact on visits.
5. We can see that the visits are seasonal across weeks. As the weekend nears the visits see a rise.

6. We also see that visits are very high in December, assuming the new year celebrations.
7. There are some high magnitude visits per day (70 - 100). However the average visits in the range 20 to 30. So using CI we can remove the higher visits data, as most of the restaurants are small.
8. Then using covariance matrix, remove the collinear features.
9. Remove outliers using Interquartile method
10. Build Knn-regressor, xgboost, lightboost models using these features, and repeat feature engineering to improve.

Notes when you build your final notebook:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument "X" (a single data points i.e 1*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
   a. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
   b. so in your final notebook, you need to pass only those two values
   c. def final(X):
      ```
              preprocess data i.e data cleaning, filling missing values etc
              compute features based on this X
              use pre trained model
              return predicted outputs
      final([time, location])
      ```
   d. in the instructions, we have mentioned two functions one with original values and one without it
   e. final([time, location]) # in this function you need to return the predictions, no need to compute the metric
   f. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y_predict)
4. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data

5. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
6. Check this live session: https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models