PREDICTING FUTURE VISITORS IN A RESTAURANT

Project Report

Submitted for

Internship program to

FYRA INSIGHTS

Guided by,

Mr. Sopan Shewale

Submitted by,

Samruddhi Yadav

MIT | Academy of Engineering

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

CONTENTS

Abstract:

For efficient and economical operation, restaurant owners need to accurately estimate the number of future customers. In this project, an approach to predict how many future visitors will go to a restaurant using supervised learning. The included data involves restaurant information, historical visits and historical reservations. With features constructed from the data, our approach generates predictions by performing machine learning operations. Evaluation of approach is done using large-scale real world datasets from two restaurant booking websites. The evaluation results show the effectiveness of the approach, as well as useful insights for future work.

Problem Statement:

The restaurant industry is totally unpredicted when it comes to understand customer behaviour. Making food ready for number customers which they don't know how many may visit on certain day, certain time may end up making losses to the restaurant owners. If they know exactly how many customers visit, it will help them to cook food accordingly and help them give better experience to the customers.

The current generation is totally driven by data - if we can use some machine learning or artificial intelligence tools based on the historical data about restaurant customers, we should be able to predict approximately number of visitors. That will help restaurants prepare appropriate amount of food, helping them convert losses into profits.

In this project - we are attempting to use Machine Algorithms and develop a System which can help predict number of visitors to a restaurant.

Feasibility study:

Feasibility and risk analysis are related in many ways .if project risk is great, the feasibility of producing quality software is required. It is one of the system analysis usually involves a thorough assessment of the operational, financial and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints .A feasibility study should be relatively cheap and done at the earliest possible time. Depending on the study,the decision is made whether to go ahead with a more detailed analysis.

Introduction

In service industry, an increasing number of business owners improve the quality of service and reduce cost using bigdata techniques. Based on historical data of business, predictive models of machine learning can estimate future moves of customers.

[1] Especially, forecasting the number of future visitors can help restaurant owners

[2] Make the best operations

[3] To maximize the revenue.

With an accurate visitor forecasting model, restaurant owners can prepare suitable amount of ingredients that exactly satisfy future visitors. In addition, restaurant owners can also schedule a suitable number of staff that can serve these visitors.

Despite various visitor forecasting techniques for other purposes such as national tourism and hotel demand in the literature, little is known for restaurant owners to estimate the number of future visitors. New customers may outnumber old customers in some restaurants of tourist destinations. Hence, it is necessary to develop a new method that can predict the total number of future visitors to a restaurant on a specific day.

To bridge this gap, in this project an approach to forecast how many future visitors will go to a restaurant using big data and supervised learning. Our approach collects big data involving restaurant information, historical visits and historical reservations. Regular restaurants can easily collect such big data on their own without any complex computing infrastructure (e.g., third-party cloud computing services). From such big data together with time information, we constructed four groups of features correspondingly.

Related Work

Predicting Future Visitors

Forecasting the number of future visitors is a meaningful task in service industry. Researchers have proposed various techniques to predicting the volume of future visitors using big data in different areas. For instance, search engine data can be used to forecast the number of future tourists to a certain popular destination in China. A strong correlation between the search queries and visitor numbers was found. Similarly, web traffic data in an area can be used to predict local hotel demand. With the popularity of devices connected to the Internet, this prediction relies on the fact that more travellers generate more traffic data. However, obtaining web search data and web traffic data is not a trivial task. Especially, many tourist destinations do not have the network connection, e.g., high mountains and small islands. Hence, many techniques predict the number of future visitors using historical visitor numbers only, e.g., predicting visitors to a city using past tourist traffic. Also, the characteristics of a location can be used for the prediction, e.g., predicting customers' revisit to a restaurant using the attributes of the restaurant. Moreover, the geographical influence is important factor in predicting future visitors going to Points of-Interests. Although various techniques are proposed for many areas to predicting the number of future visitors in service industry, little is known for restaurant owners to forecast the number of future visitors using big data.

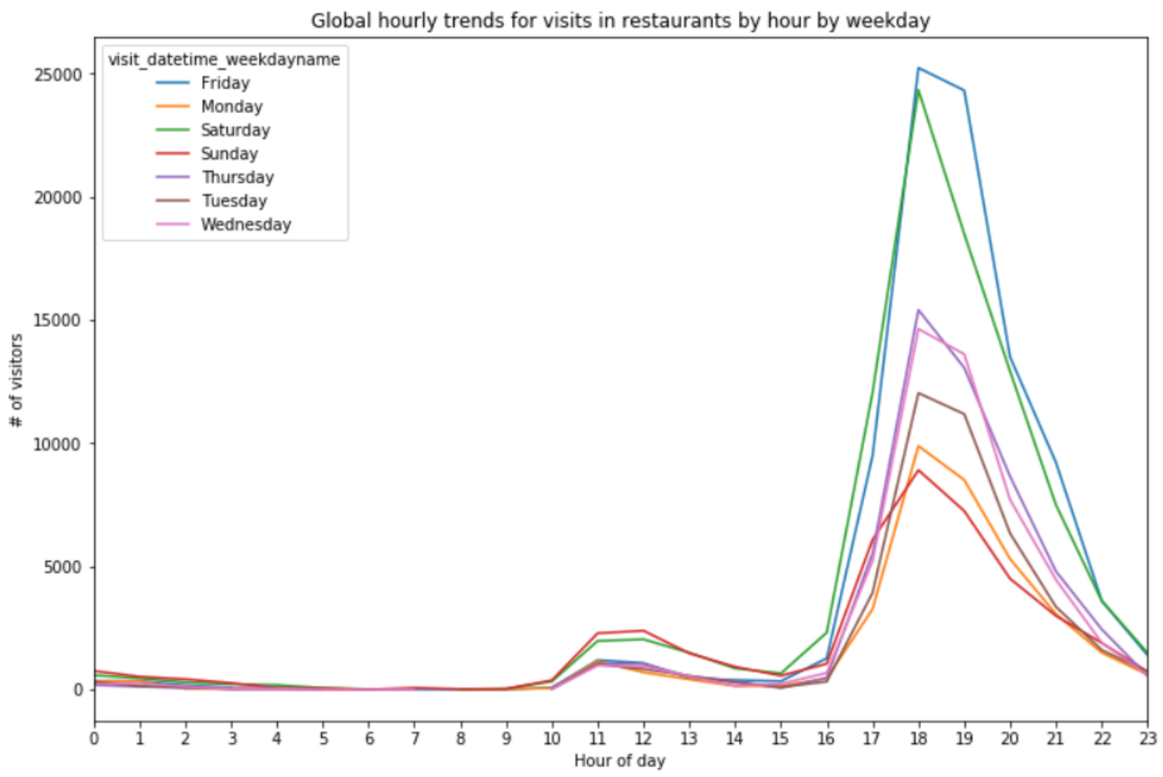
Implementation

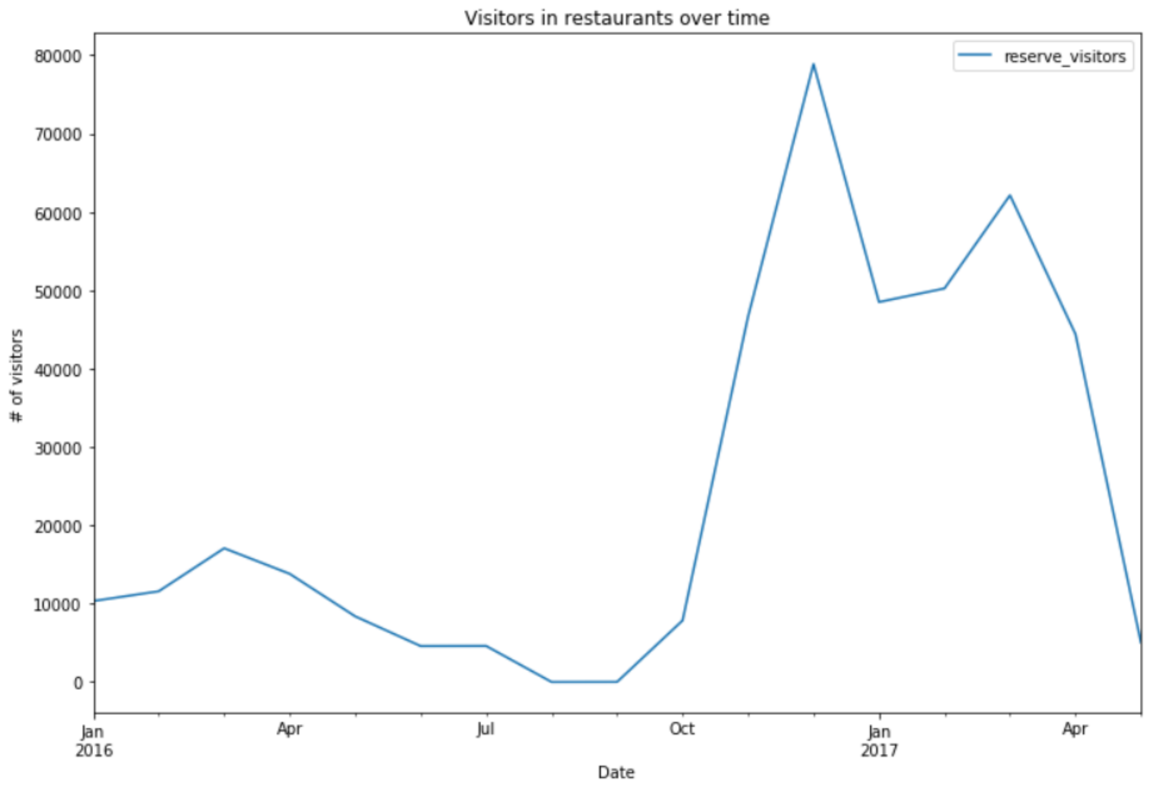
We implemented our approach using Python 3.6. The computer to run the program is a regular HP laptop with Intel Core i5-5300 CPU. The computer has 8GB physical memory. The program does not require GPU for computation.

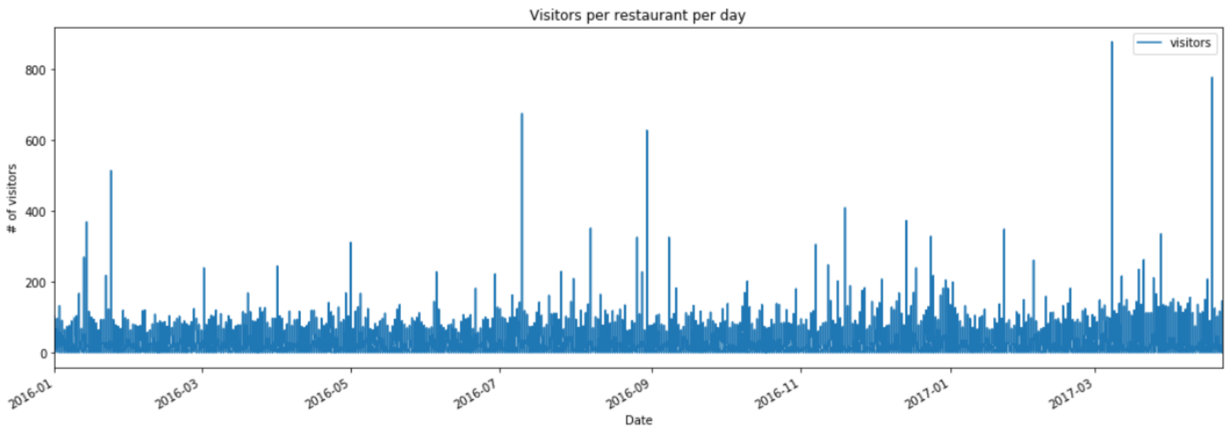
Dataset

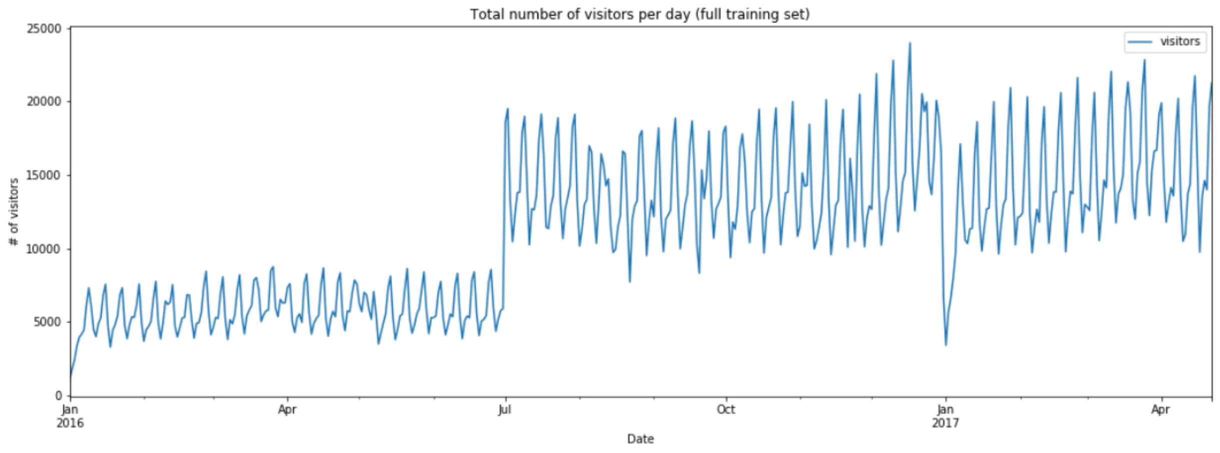
We selected two open large-scale real-world datasets from two restaurant booking websites: HOTPEPPER GOURMET and AirREGI. The two datasets contain the visitor and reservation history of 150 restaurants from year 2016 to the first season of 2017. We combined the two datasets to form a training dataset for our evaluation (2350992 instances in total). The test set, containing 2019 instances, records how many visitors go to these restaurants in the two following months in year 2017.

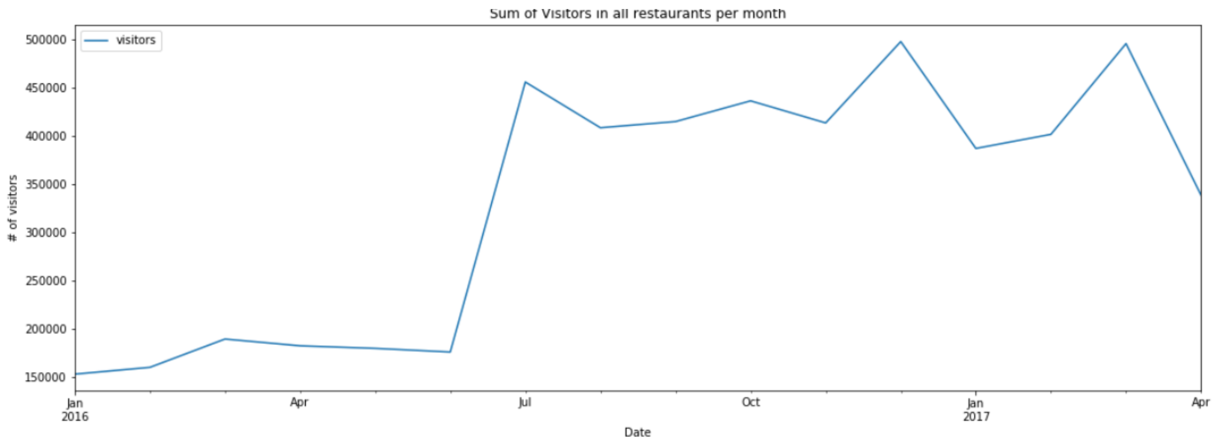
Model

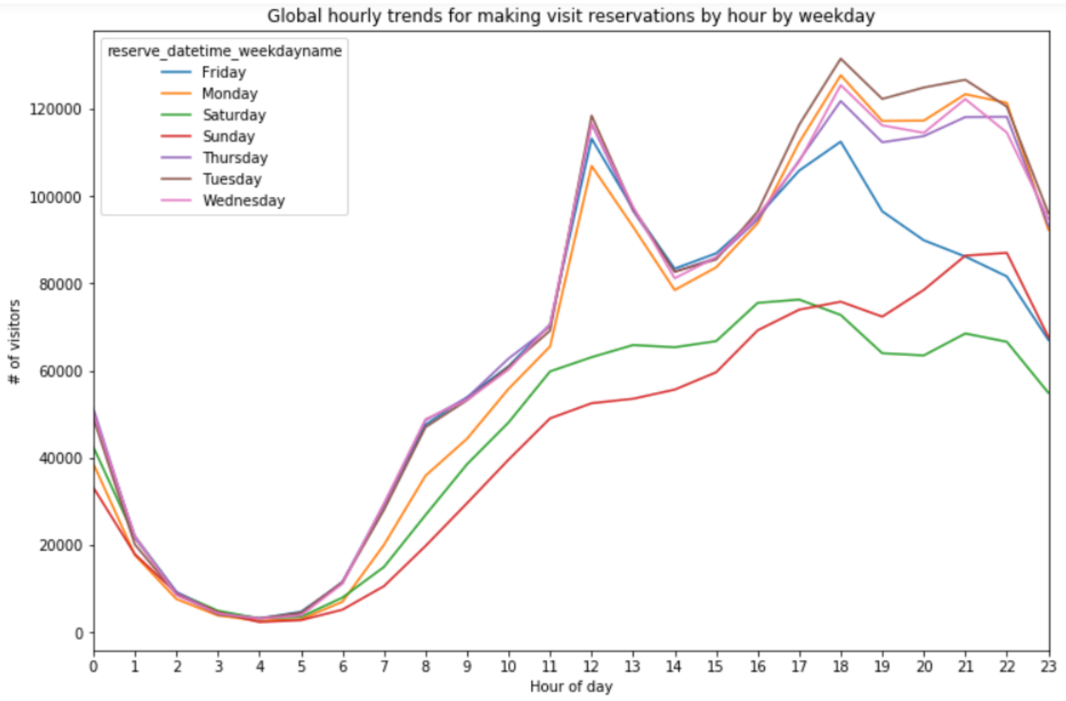

Global hourly trends for visits in restaurants by hour by weekday
visit_datetime_weekdayname
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
# of visitors
Hour of day

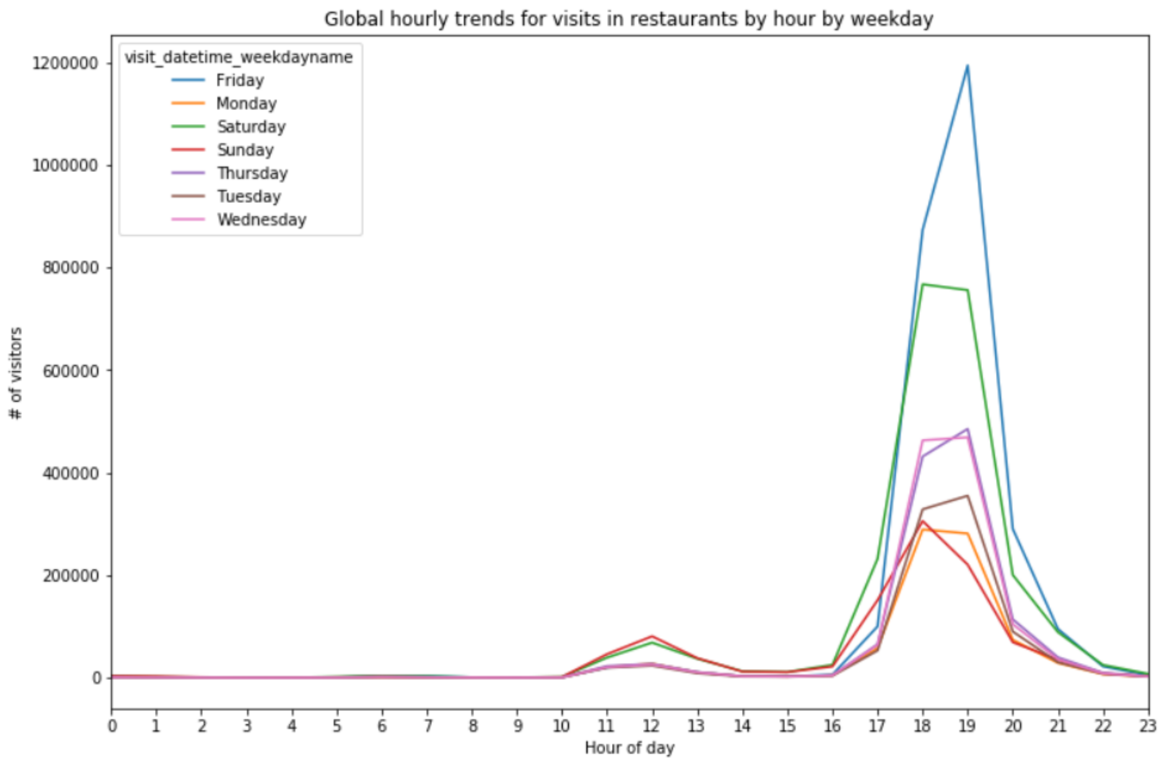

Visitors in restaurants over time
reserve_visitors
# of visitors
Date

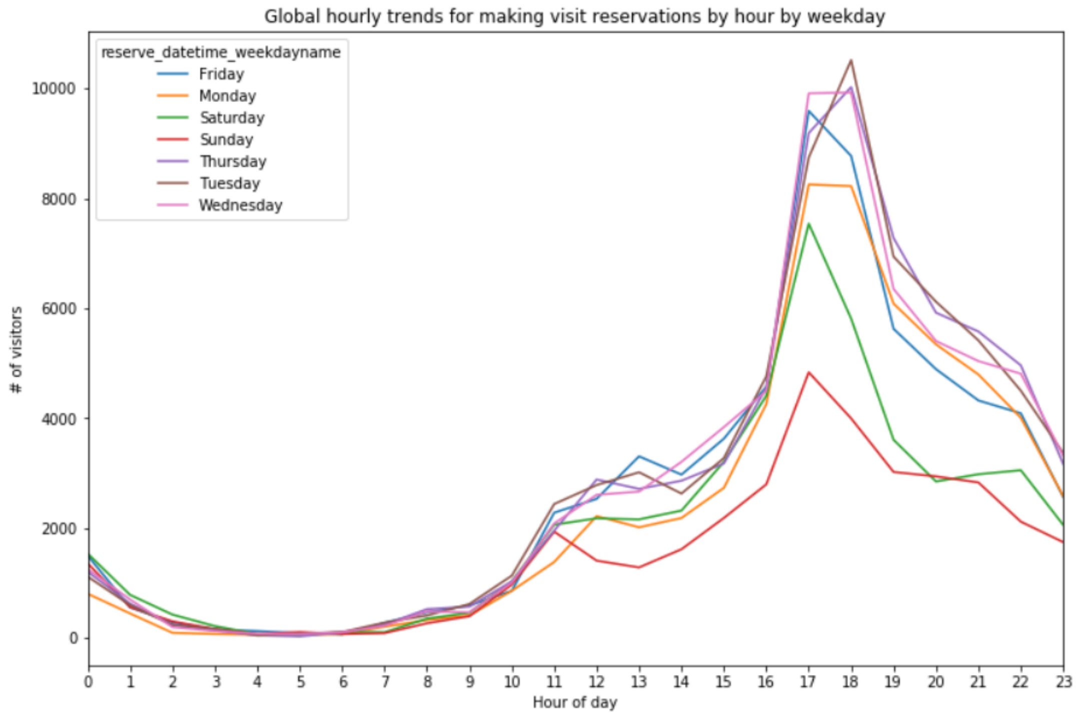

Visitors per restaurant per day
visitors
# of visitors
Date
800
600
400
200
0
2016-01
2016-03
2016-05
2016-07
2016-09
2016-11
2017-01
2017-03

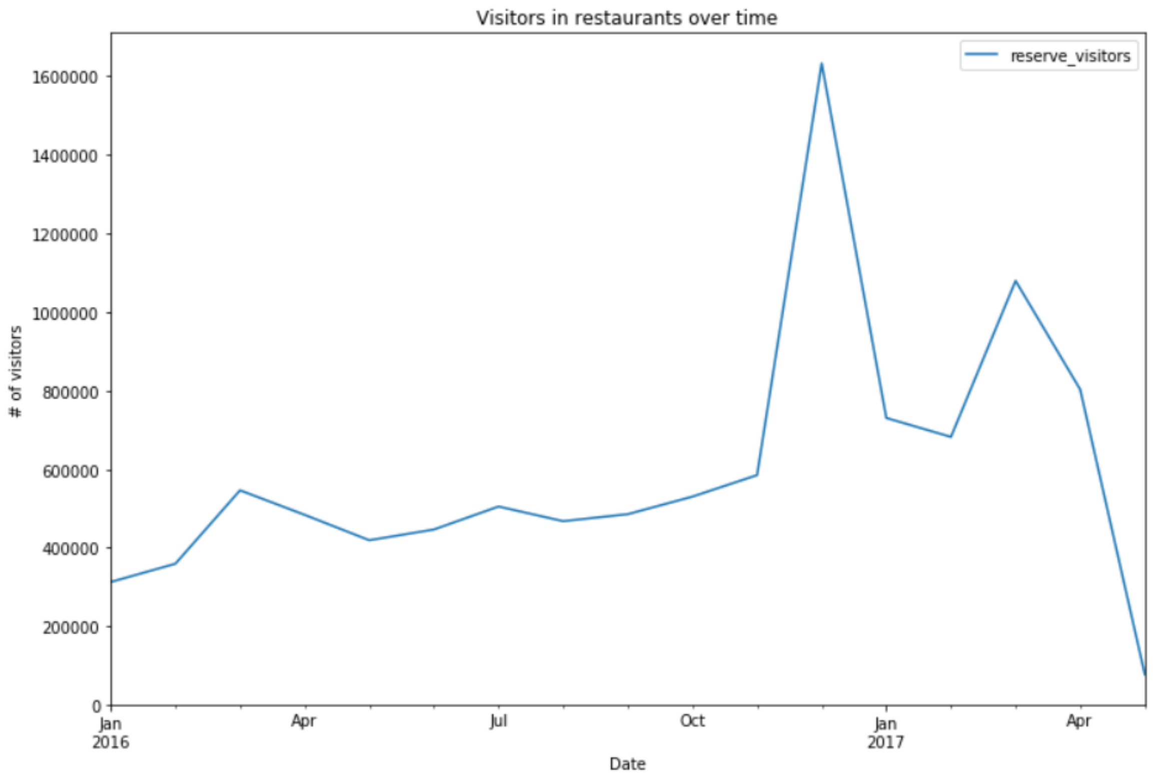

Total number of visitors per day (full training set)
visitors
# of visitors
Date
25000
20000
15000
10000
5000
0
Jan 2016
Apr
Jul
Oct
Jan 2017
Apr


Sum of Visitors in all restaurants per month
visitors
# of visitors
Date
500000
450000
400000
350000
300000
250000
200000
150000
Jan 2016
Apr
Jul
Oct
Jan 2017
Apr


Global hourly trends for making visit reservations by hour by weekday
reserve_datetime_weekdayname
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
# of visitors
Hour of day
0
20000
40000
60000
80000
100000
120000


Global hourly trends for visits in restaurants by hour by weekday
visit_datetime_weekdayname
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
# of visitors
Hour of day
0
200000
400000
600000
800000
1000000
1200000


Global hourly trends for making visit reservations by hour by weekday
reserve_datetime_weekdayname
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
# of visitors
Hour of day


Visitors in restaurants over time
reserve_visitors
# of visitors
Date

Limitation and Future Work

Beyond features considered by our approach, many more factors can facilitate accurate prediction of future visitors to a restaurant. For instance, bad weather of the restaurant location may reduce visitors. Also, if a new restaurant is opened next to an existing restaurant, the number of future visitors going to this existing restaurant may drop. Moreover, social events can bring more visitors to restaurants of the related venues. Hence, in future work, it is necessary to include more information to the predictive model, such as weather, competitors and social events. For some kinds of restaurants, it is challenging to accurately count how many visitors have come on a day. For example, one customer may buy food for all its friends at fast food restaurants, such as McDonald’s (especially when customers in a car buy food via ”Drive-Thru”). Future work can also explore new customer counting methods for restaurants based on new technology (such as combining video camera and the Internet of Things, or new software/hardware.

Implications

It is shown that it is feasible to predict how many future visitors will go to a restaurant using big data and supervised learning. To drive our approach, users only need to collect big data, including restaurant information, historical visits and historical reservations. Considering that restaurant owners may not have access to high-performance computers, we show that the prediction can be achieved by a mix of low-computation regressors, rather than relying on computation-intensive methods such as Deep Learning (requiring long time of training and mostly requiring high performance GPU) and SVM (requiring huge amount of memory). In evaluation, we found that a low training error cannot ensure a low test error for a regressor. Hence, users should not minimize training error by all means during training. A relatively high training error often reflects the robustness of a model. Regarding the strongest indicators of future visitors, we found that time-related features (such as week of year, month of year and day of week) and historical visitor records (such as mean, median, maximal and minimal visitors on a day) are most useful for the prediction. In addition, the unique ID and location of restaurant are also important features. This indicates that each individual restaurant follows a unique pattern, even if other factors are similar. By contrast, historical reservations can hardly help the prediction. We quantified the computation time on our computer in evaluation.

Code and results:

```
# In[1]:



import glob


files = glob.glob("*.csv")


print(files)



# In[2]:



import pandas as pd


df = {}


for filename in files:
    df[filename.replace('.csv', '')] = pd.read_csv(filename)


print(df.keys())
```

```
# In[3]:


print(len(pd.unique(df['store_id_relation']['air_store_id'])))
df['store_id_relation'].head()


# In[4]:


print(len(pd.unique(df['air_reserve']['air_store_id'])))
df['air_reserve'].head()


# In[5]:


print(len(df['air_store_info']))
df['air_store_info'].head()


# In[6]:
```

```
print(len(df['hpg_store_info']))

df['hpg_store_info'].head()



# In[7]:



df['sample_submission'].head()



# In[8]:



df['air_visit_data']['visit_datetime'] = pd.to_datetime(df['air_visit_data']['visit_date'])

df['air_visit_data']['visit_date'] = df['air_visit_data']['visit_datetime'].dt.date

df['air_visit_data'].head()



# In[9]:




df['sample_submission']['visit_datetime'] = df['sample_submission']['id'].map(lambda x:
str(x).split('_')[2])

df['sample_submission']['air_store_id'] = df['sample_submission']['id'].map(lambda x:
'_'.join(x.split('_')[:2]))
```

```
df['sample_submission']['visit_datetime'] =
pd.to_datetime(df['sample_submission']['visit_datetime'])

df['sample_submission']['visit_date'] = df['sample_submission']['visit_datetime'].dt.date

df['sample_submission'].drop(['id'], axis = 1, inplace = True)

df['sample_submission'].head()


# In[10]:


total_dataset = pd.concat([df['air_visit_data'], df['sample_submission']], axis = 0)

print(len(df['air_visit_data']))

print(len(df['sample_submission']))

print(len(total_dataset))

total_dataset.head()


# In[11]:


df['date_info']['calendar_date'] = pd.to_datetime(df['date_info']['calendar_date'])

df['date_info']['visit_date'] = df['date_info']['calendar_date'].dt.date

df['date_info'] = df['date_info'].drop(['calendar_date'], axis = 1)

df['date_info'].head()
```

```
# In[12]:


total_dataset = pd.merge(total_dataset, df['air_store_info'], how='left', on='air_store_id')
total_dataset = pd.merge(total_dataset, df['date_info'], how='left', on='visit_date')
total_dataset['visit_datetime'] = pd.to_datetime(total_dataset['visit_date'])
total_dataset['year']  = total_dataset['visit_datetime'].dt.year
total_dataset['month'] = total_dataset['visit_datetime'].dt.month
total_dataset['day']   = total_dataset['visit_datetime'].dt.day
total_dataset.drop('visit_datetime', axis=1, inplace=True)

# One Hot Encoding Conversion of the 'air_genre_name', 'air_area_name', 'day_of_week', 'year'
using the pandas.get_dummies

cat_features = [col for col in ['air_genre_name', 'air_area_name', 'day_of_week', 'year']]
for column in cat_features:
    temp = pd.get_dummies(pd.Series(total_dataset[column]))
    total_dataset = pd.concat([total_dataset,temp],axis=1)
    total_dataset = total_dataset.drop([column],axis=1)

total_dataset.head()
```

```
# In[13]:



total_dataset.drop(['latitude', 'longitude'], axis = 1, inplace = True)

temp = pd.get_dummies(total_dataset['air_store_id'])

total_dataset = pd.concat([total_dataset,temp],axis=1)

total_dataset.head()



# In[14]:



sep = len(df['air_visit_data'])

train = total_dataset[:sep]

to_predict = total_dataset[sep:]

print(len(train))

print(len(to_predict))



# In[15]:



import pandas as pd

import numpy as np

from sklearn.metrics import mean_squared_error
```

```
def RMSLE(y, pred):
    return mean_squared_error(y, pred)**0.5



# In[16]:



col = [c for c in train if c not in ['air_store_id', 'visit_date', 'visitors']]
X_train, y_train = train[col], train['visitors']
X_to_predict = to_predict[col]
value_X = X_train.values
value_y = y_train.values
value_X_to_predict = X_to_predict.values
print(value_X.shape)
print(value_y.shape)
print(value_X_to_predict.shape)



# In[17]:




from sklearn.preprocessing import MinMaxScaler


scaler_X = MinMaxScaler(feature_range=(0, 1))
```

```
scaled_X = scaler_X.fit_transform(value_X)

scaled_X_to_predict = scaler_X.transform(value_X_to_predict)



# In[18]:



from sklearn.cross_validation import train_test_split


X_train, X_test, y_train, y_test = train_test_split(scaled_X, value_y, test_size=0.3,
random_state=42)



# In[18]:



from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(scaled_X, value_y, test_size=0.3,
random_state=42)



# In[23]:



pip install keras
```

```
# In[19]:


from keras.models import Sequential
from keras import optimizers, regularizers
from keras.layers import Dense, Input, Activation, Dropout

# Build the Neural Network in Keras
model = Sequential()
model.add(Dense(512, activation = 'relu', input_shape=(958,)))
model.add(Dense(256, activation = 'relu', input_shape=(958,)))
model.add(Dense(128, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='relu'))

sgd = optimizers.SGD(lr=0.005, decay=1e-6, momentum=0.9, nesterov=False)

model.compile(loss='mean_squared_error', optimizer=sgd)

for l in model.layers:
    print(l.name, l.input_shape, l.output_shape)
```

```
# In[22]:


pip install tensorflow


# In[23]:


import tensorflow


# In[20]:




history = model.fit(X_train, np.log1p(y_train), batch_size = 128, epochs=20,
                    validation_data=(X_test, np.log1p(y_test)), verbose=1)


# In[21]:
```

```
# Evaluate the model with RMSLE

y_ = model.predict(X_test)

y_test = y_test.reshape(-1, 1)

rmsle = RMSLE(np.log1p(y_test), y_)

print('Test RMSLE: %.3f' % rmsle)



# In[22]:




y_predicted = np.expm1(model.predict(scaled_X_to_predict))


submit = pd.read_csv('sample_submission.csv')


submit['visitors'] = y_predicted


submit.to_csv('final.csv', encoding='utf-8', index=False)
```

Conclusion

In this project, we present an approach to estimate how many future visitors will go to a restaurant using big data and supervised learning. The included big data involves restaurant information, historical visits and historical reservations. With features constructed from the big data, our approach generates predictions. Approach is evaluated by using large-scale real-world datasets from two restaurant booking websites. The evaluation results show the effectiveness of our approach, as well as useful insights for future work.