Restaurant Data with Consumer Rating

Problem Statement

The problem is to predict the rating of the restaurant according to the consumer preferences

Dataset Description

This data is used for a study where the restaurants according to the consumer preferences and certain significant features.

There are total nine file

First 5 file based on restaurant they provided to customer payment accepted by restaurants,cuisines served by restaurant,parking area,location of restaurants,time spend in restaurant

Another 3 file based on customer profile ,their payment mode and cuisines like by customer

One file is rating file based on placeID, user food rating,service rating

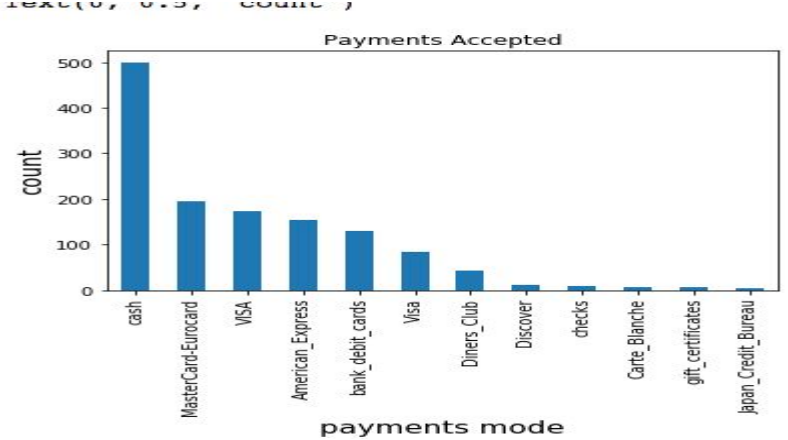
Data Analysis

We analyze first 5 file that has data of restaurant

1. chefmozaccepts.csv : This file defines mode of payment accepted by the restaurants. There are total 12 cards used such as cash, bank Debit cards , MasterCard , American-Express etc in which cash payment is more accepted

```
#Describe the dataset
accepts['Rpayment'].describe()
```

```
count      1314
unique       12
top        cash
freq        500
Name: Rpayment, dtype: object
```



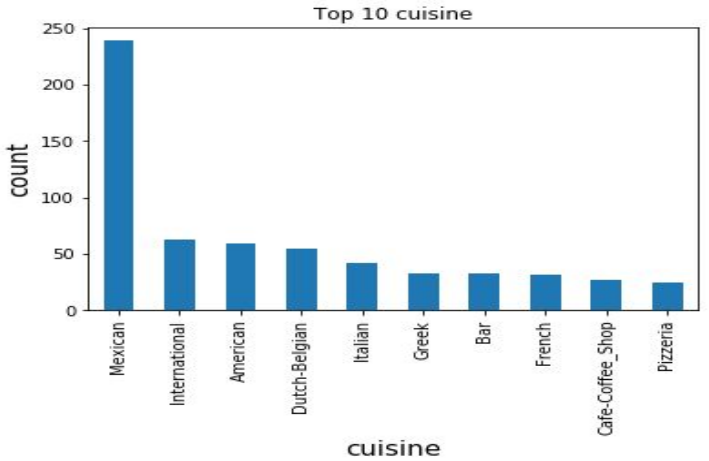
Data Analysis

2.chefmozcuisine.csv : This file has contain 59 different cuisines such as Spanish , Italian , mexican,Fast-food etc in which more cuisine served is Mexican.

```
#Describe the dataset
cuisine['Rcuisine'].describe()
```

```
count          916
unique          59
top        Mexican
freq           239
Name: Rcuisine, dtype: object
```

```
Text(0, 0.5, 'count')
```



Data Analysis

3. chefmozhours4.csv : This file has information about opening and closing time of restaurant and number of days for which it is open. We just split the given number of days.

```
hours['days'].describe()
```

```
count                     2339
unique                       3
top       Mon;Tue;Wed;Thu;Fri;
freq                       793
Name: days, dtype: object
```

Data Analysis

4. chefmozparking.csv : This file has data related to parking areas owned by restaurants such as public parking , no parking etc. in which no parking was the most most frequent.

```
parking['parking_lot'].describe()
```

```
count       702
unique        7
top        none
freq        348
Name: parking_lot, dtype: object
```

Data Analysis

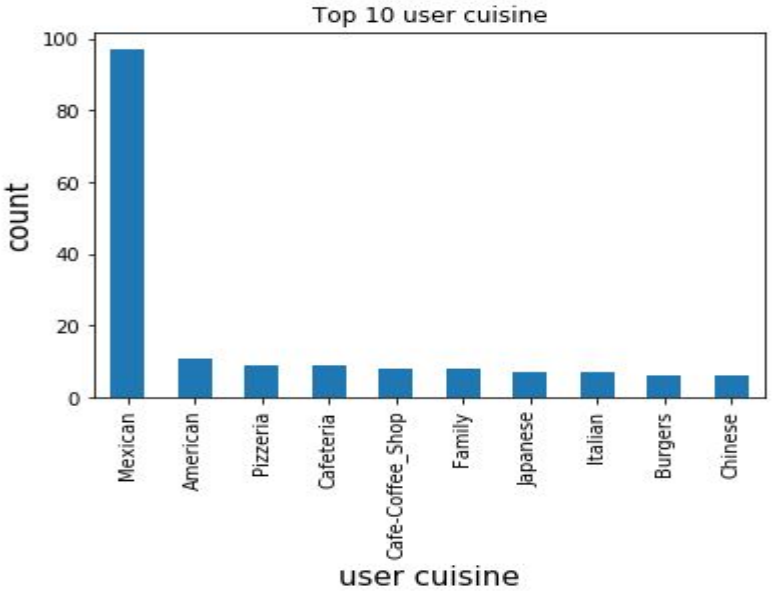
5.geoplaces.csv : This file contain data like name , address, fax , url of ambience , smoking area, alcohol , price etc of restaurant.

	missing_value	percent_missing
placeID	0	0.000000
latitude	0	0.000000
longitude	0	0.000000
the_geom_meter	0	0.000000
name	0	0.000000
address	27	20.769231
city	18	13.846154
state	18	13.846154
country	28	21.538462
fax	130	100.000000
zip	74	56.923077
alcohol	0	0.000000
smoking_area	0	0.000000
dress_code	0	0.000000
accessibility	0	0.000000
price	0	0.000000
url	116	89.230769
Rambience	0	0.000000
franchise	0	0.000000
area	0	0.000000
other_services	0	0.000000

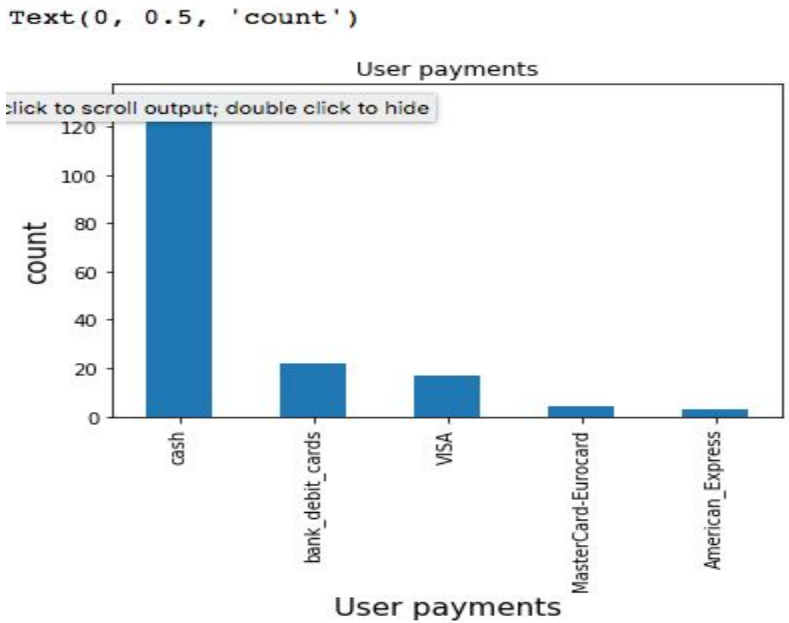
Data Analysis

6. usercuisine.csv : This file has contain 103 cuisine of users choice.

```
Text(0, 0.5, 'count')
```

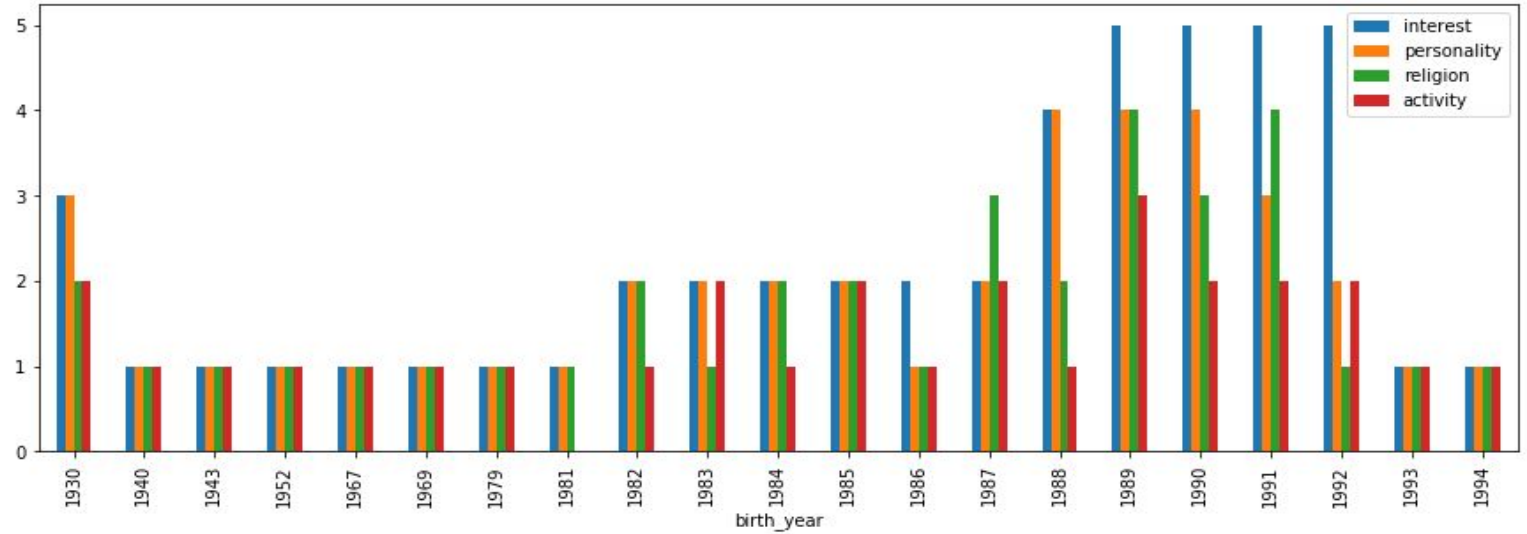


7. userpayment.csv : This file has 5 mode of payment of user like cash,bank-debit-card,american express,visa,mastercard



8. userprofile.csv : This file has contain all the user related information like his location, drinking_level , interest, religion,birth_year,smoker or not ,marital status, address etc.

```
#plot to visualize user's personal info based on birthyear.
profileplt=profilerep.groupby('birth_year')['interest','personality','religion','activity'].nunique().plot.bar(figsize=
```



9. Rating.csv : This file has given rating according user and restaurant

	placeID	rating	food_rating	service_rating
count	1161.000000	1161.000000	1161.000000	1161.000000
mean	134192.041344	1.199828	1.215332	1.090439
std	1100.916275	0.773282	0.792294	0.790844
min	132560.000000	0.000000	0.000000	0.000000
25%	132856.000000	1.000000	1.000000	0.000000
50%	135030.000000	1.000000	1.000000	1.000000
75%	135059.000000	2.000000	2.000000	2.000000
max	135109.000000	2.000000	2.000000	2.000000

Correlation Matrix

1.We merged all eight file with rating one by one on the basis of their userID and placeID
2.We plot correlation matrix to see correlation between all the variables.

3.By plotting , we got that some of the variables like marital_status, parking_lot ,address etc are not linearly correlated.

4.Hence we dropped these column and tried to fit a line between all the variables and the overall ratings.

Model building

1.Training and testing:

Before using algorithm , we split our data into training set (75%) and test set (25%).

2.Algorithm:

In this classification algorithm is used for calculating score which is done using

- Logistic regression
- Decision Tree
- Random Forest

Logistic Regression

1. By using Decision Tree Algorithm we got accuracy of 78.07%.

```
print(logmodel.score(X_train, y_train))
```

```
0.7772320866389624
```

```
print(logmodel.score(X_test, y_test))
```

```
0.7805891238670695
```

```
print("classification_report")
print(classification_report(y_test,predict1))
```

```
classification_report
              precision    recall  f1-score   support

           0       0.91      0.88      0.90      1129
           1       0.66      0.56      0.61       624
           2       0.70      0.81      0.75       895

    accuracy                           0.78      2648
   macro avg       0.76      0.75      0.75      2648
weighted avg       0.78      0.78      0.78      2648
```

Decision Tree

1. By using Decision Tree Algorithm we got accuracy of 97.5%.

```
print(decimodel.score(X_train,y_train))
```

```
1.0
```

```
print(decimodel.score(X_test, y_test))
```

```
0.974320241691843
```

```
print("classification_report")
print(classification_report(y_test,predict2))
```

```
classification_report
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      1129
           1       0.95      0.96      0.96       624
           2       0.98      0.97      0.97       895

    accuracy                           0.97      2648
   macro avg       0.97      0.97      0.97      2648
weighted avg       0.97      0.97      0.97      2648
```

Random Forest

1. We also compared decision tree score with the ensemble algorithm like Random Forest regressor but the score was again 75.94%.

```
print(Randmodel.score(X_train, y_train))
```

```
0.7604835663014734
```

```
print(Randmodel.score(X_test, y_test))
```

```
0.7594410876132931
```

```
print("classification_report")
print(classification_report(y_test,predict3))
```

```
classification_report
              precision    recall  f1-score   support

           0       0.97      0.82      0.89      1129
           1       0.80      0.39      0.52       624
           2       0.61      0.95      0.74       895

    accuracy                           0.76      2648
   macro avg       0.79      0.72      0.72      2648
weighted avg       0.81      0.76      0.75      2648
```

Conclusion

- Using Decision tree algorithm ,we can predict the rating of a restaurant with a very high accuracy.
- The factors that were affecting the rating most are:
- Food_rating
- Service_rating
- Price
- Other Services