Table Of Content :-

1- import libraries :-

In [2]:
```
# to make Data Manpulation
import pandas as pd
import numpy  as np

# to make Data Visuallization
import matplotlib.pyplot as plt
import seaborn as sns

# to ignore the Errors
import warnings
warnings.filterwarnings("ignore")
```

2- Read the dataset file:-

In [3]:
```
# the Dataframe we will call as data
data = pd.read_csv(r"C:\Users\dell\Desktop\powerbi projects\datasets\uber_eatscsv\uber_eat_rest.csv")
```

3- Show abrief of dataframe:-

In [4]:
```
# show top 5 row with all columns
data.head()
```

Out[4]:

	id	position	name	score	ratings	category	price_range	full_address	zip_code	lat	lng
0	1	19	PJ Fresh (224 Daniel Payne Drive)	NaN	NaN	Burgers, American, Sandwiches	$	224 Daniel Payne Drive, Birmingham, AL, 35207	35207	33.562365	-86.830703
1	2	9	J' ti``z Smoothie-N-Coffee Bar	NaN	NaN	Coffee and Tea, Breakfast and Brunch, Bubble Tea	NaN	1521 Pinson Valley Parkway, Birmingham, AL, 35217	35217	33.583640	-86.773330
2	3	6	Philly Fresh Cheesesteaks (541-B Graymont Ave)	NaN	NaN	American, Cheesesteak, Sandwiches, Alcohol	$	541-B Graymont Ave, Birmingham, AL, 35204	35204	33.509800	-86.854640
3	4	17	Papa Murphy's (1580 Montgomery Highway)	NaN	NaN	Pizza	$	1580 Montgomery Highway, Hoover, AL, 35226	35226	33.404439	-86.806614
4	5	162	Nelson Brothers Cafe (17th St N)	4.7	22.0	Breakfast and Brunch, Burgers, Sandwiches	NaN	314 17th St N, Birmingham, AL, 35203	35203	33.514730	-86.811700

In [86]:
```
# show last 5 row with all columns
data.head()
```

Out[86]:

	id	position	name	score	ratings	category	price_range	full_address	zip_code	lat	lng
0	1	19	PJ Fresh (224 Daniel Payne Drive)	NaN	NaN	Burgers, American, Sandwiches	$	224 Daniel Payne Drive, Birmingham, AL, 35207	35207	33.562365	-86.830703
1	2	9	J' ti``z Smoothie-N-Coffee Bar	NaN	NaN	Coffee and Tea, Breakfast and Brunch, Bubble Tea	NaN	1521 Pinson Valley Parkway, Birmingham, AL, 35217	35217	33.583640	-86.773330
2	3	6	Philly Fresh Cheesesteaks (541-B Graymont Ave)	NaN	NaN	American, Cheesesteak, Sandwiches, Alcohol	$	541-B Graymont Ave, Birmingham, AL, 35204	35204	33.509800	-86.854640
3	4	17	Papa Murphy's (1580 Montgomery Highway)	NaN	NaN	Pizza	$	1580 Montgomery Highway, Hoover, AL, 35226	35226	33.404439	-86.806614
4	5	162	Nelson Brothers Cafe (17th St N)	4.7	22.0	Breakfast and Brunch, Burgers, Sandwiches	NaN	314 17th St N, Birmingham, AL, 35203	35203	33.514730	-86.811700

4- Data Exploration:-

In [5]:
```
# Get the data shape
data.shape
```

Out[5]:
```
(40227, 11)
```

In [88]:
```
# Get the data size
data.size
```

Out[88]:
```
442497
```

In [89]:
```
# get more info about data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40227 entries, 0 to 40226
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   id            40227 non-null  int64  
 1   position      40227 non-null  int64  
 2   name          40227 non-null  object 
 3   score         22254 non-null  float64
 4   ratings       22254 non-null  float64
 5   category      40204 non-null  object 
 6   price_range   33581 non-null  object 
 7   full_address  39949 non-null  object 
 8   zip_code      39940 non-null  object 
 9   lat           40227 non-null  float64
 10  lng           40227 non-null  float64
dtypes: float64(4), int64(2), object(5)
memory usage: 3.4+ MB
```

That mean the dataset is consets of 40227 rows (instances) and 11 column (feature)

There are more than datatypes (int64, Float64, object) , so we have Numerical and Categorical Data

In [90]:
```
# Get more Statistical info about the Numerical data
data.describe()
```

Out[90]:

	id	position	score	ratings	lat	lng
count	40227.000000	40227.000000	22254.000000	22254.000000	40227.000000	40227.000000
mean	20114.000000	79.529843	4.560996	74.870989	39.927033	-96.549713
std	11612.678976	77.611449	0.298041	72.381529	5.765774	17.961361
min	1.000000	1.000000	1.300000	10.000000	0.000000	-123.841240
25%	10057.500000	14.000000	4.400000	25.000000	37.082007	-113.587301
50%	20114.000000	51.000000	4.600000	51.000000	39.000990	-96.587547
75%	30170.500000	129.000000	4.800000	100.000000	45.493640	-77.528825
max	40227.000000	300.000000	5.000000	500.000000	48.963950	0.000000

In [91]:
```
# Find how many unique values
data.nunique()
```

Out[91]:
```
id              40227
position          300
name            38863
score              33
ratings           416
category        10647
price_range         4
full_address    35302
zip_code         2281
lat             36780
lng             36745
dtype: int64
```

In [92]:
```
# print All the Data columns
data.columns
```

Out[92]:
```
Index(['id', 'position', 'name', 'score', 'ratings', 'category', 'price_range',
       'full_address', 'zip_code', 'lat', 'lng'],
      dtype='object')
```

In [6]:
```
# check null values
data.isna().sum()
```

Out[6]:
```
id                  0
position            0
name                0
score           17973
ratings         17973
category           23
price_range      6646
full_address      278
zip_code          287
lat                 0
lng                 0
dtype: int64
```

: There are Missing Values , We should make Data cleaning

5- Data cleaning:-

In [7]:
```
# Check Duplicates
print("\nDuplicates Rows:", data.duplicated().sum())
```

```
Duplicates Rows: 0
```

In [8]:
```
# check null values
data.isna().sum()
```

Out[8]:
```
id                  0
position            0
name                0
score           17973
ratings         17973
category           23
price_range      6646
full_address      278
zip_code          287
lat                 0
lng                 0
dtype: int64
```

- We have missing Values in : -

- 1- (score, ratings) which is numerical data
- 2- (category, price_range, full_address, zip_code) which is Categorical data

- Lets Handle This Missing values:-

- 1- numerical data ----> fill with median (beacuse of using mean affected by outliers)
- 2- Categorical data ----> fill (price_range) with Forward fill , drop (category, full_address, zip_code)

In [9]:
```
# create Data Dictionary to fill the missing values

values = {
            # Numerical data
            'score'  : data['score'].median(),
            'ratings': data['ratings'].median(),
            # Categorical data
            'price_range': data['price_range'].fillna(method='ffill') # forward fill
          }
```

In [10]:
```
# Use The Data Dictionary to fill missing Values And Save The Resualts
data.fillna(value= values, inplace=True)
```

We filled all Missing Values, Lets delete the others (category, full_address, zip_code)

In [11]:
```
# drop null missimg values and save the result
data.dropna(inplace=True)
```

In [12]:
```
# check null values
data.isna().sum()
```

Out[12]:
```
id              0
position        0
name            0
score           0
ratings         0
category        0
price_range     0
full_address    0
zip_code        0
lat             0
lng             0
dtype: int64
```

There Are not any Missing Values

✓ Now The Data is cleaned And Ready For Exploratory Data Analysis (EDA)

6- Exploratory Data Analysis:-

In [45]:
```
data.columns
```

Out[45]:
```
Index(['id', 'position', 'name', 'score', 'ratings', 'category', 'price_range',
       'full_address', 'zip_code', 'lat', 'lng'],
      dtype='object')
```

In [55]:
```
data.head()
```

Out[55]:

	id	position	name	score	ratings	category	price_range	full_address	zip_code	lat	lng
0	1	19	PJ Fresh (224 Daniel Payne Drive)	4.6	500	Burgers, American, Sandwiches	$	224 Daniel Payne Drive, Birmingham, AL, 35207	35207	33.562365	-86.830703
1	2	9	J' ti``z Smoothie-N-Coffee Bar	4.6	500	Coffee and Tea, Breakfast and Brunch, Bubble Tea	$	1521 Pinson Valley Parkway, Birmingham, AL, 35217	35217	33.583640	-86.773330
2	3	6	Philly Fresh Cheesesteaks (541-B Graymont Ave)	4.6	500	American, Cheesesteak, Sandwiches, Alcohol	$	541-B Graymont Ave, Birmingham, AL, 35204	35204	33.509800	-86.854640
3	4	17	Papa Murphy's (1580 Montgomery Highway)	4.6	500	Pizza	$	1580 Montgomery Highway, Hoover, AL, 35226	35226	33.404439	-86.806614
4	5	162	Nelson Brothers Cafe (17th St N)	4.7	500	Breakfast and Brunch, Burgers, Sandwiches	$	314 17th St N, Birmingham, AL, 35203	35203	33.514730	-86.811700

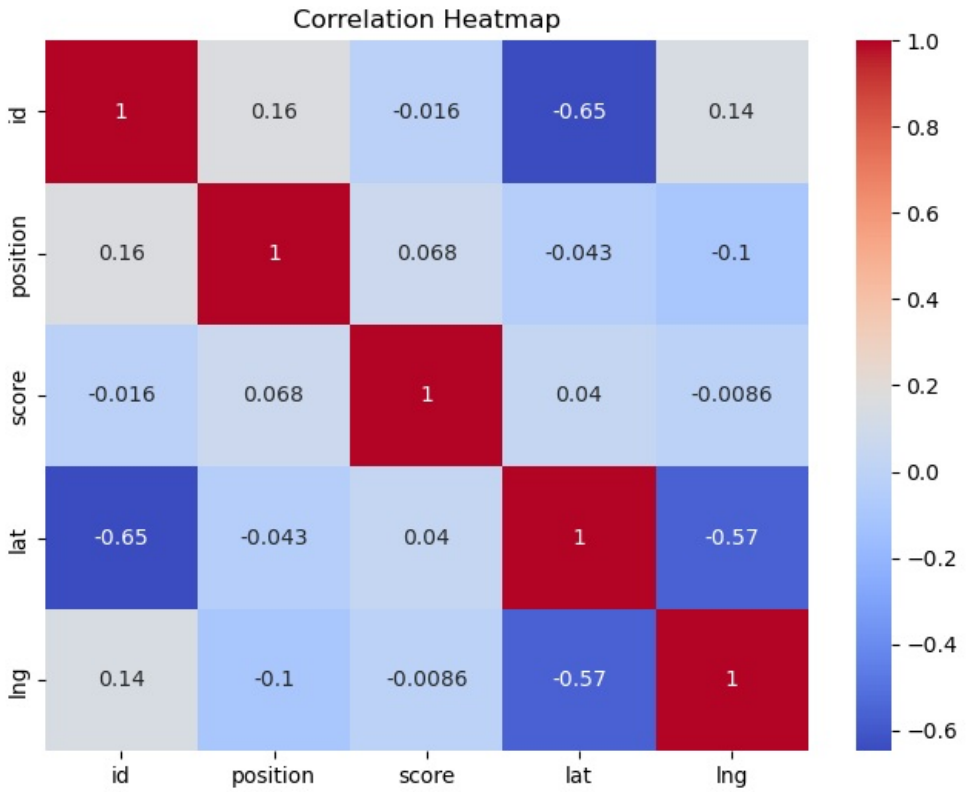
In [70]:
```
data[ (data['price_range']=='$$$') & (data['score']==5.0) ].sort_values(by='id')
```

Out[70]:

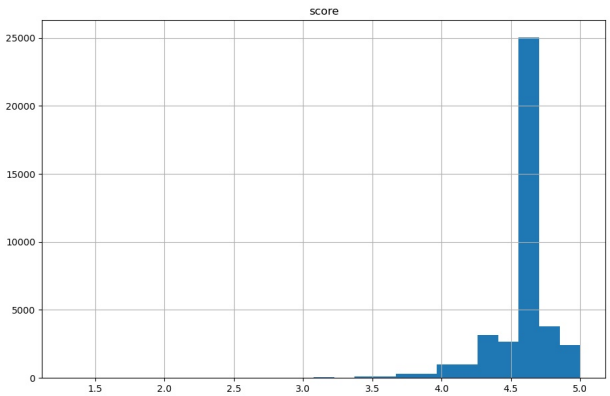
	id	position	name	score	ratings	category	price_range	full_address	zip_code	lat	lng
15	16	88	Jeni's Splendid Ice Cream (Pepper Place)	5.0	500	Ice Cream & Frozen Yogurt, Comfort Food, D...	$$$	219 29th St S, Birmingham, AL, 35233	35233	33.516600	-86.789950
13594	13595	209	Vego Eatz	5.0	500	Vegetarian, Healthy	$$$	203 W Pioneer Ave, Puyallup, WA, 98371	98371	47.190590	-122.295470
18947	18948	54	Jeni's Splendid Ice Creams (Old Town Alexandria)	5.0	500	Ice Cream & Frozen Yogurt, Comfort Food, D...	$$$	102 South Patrick Street, Alexandria, VA, 22314	22314	38.805220	-77.050317
20017	20018	219	Laporta's Restaurant	5.0	500	American, Burgers, Pasta	$$$	1600 Duke St, Alexandria, VA, 22314	22314	38.803849	-77.058314
20619	20620	59	Bar Charley	5.0	500	American	$$$	1825 18th St NW, Washington, DC, 20009	20009	38.915028	-77.041450
26444	26445	92	Jeni's Ice Cream Bethesda	5.0	500	Ice Cream & Frozen Yogurt, Comfort Food, D...	$$$	4918 Elm Street, Bethesda, MD, 20814	20814	38.982007	-77.097036
28976	28977	63	Jeni's Barracks Row	5.0	500	Ice Cream & Frozen Yogurt, Comfort Food, D...	$$$	526 8th Street Southeast, Washington, DC, 20003	20003	38.881810	-76.994580

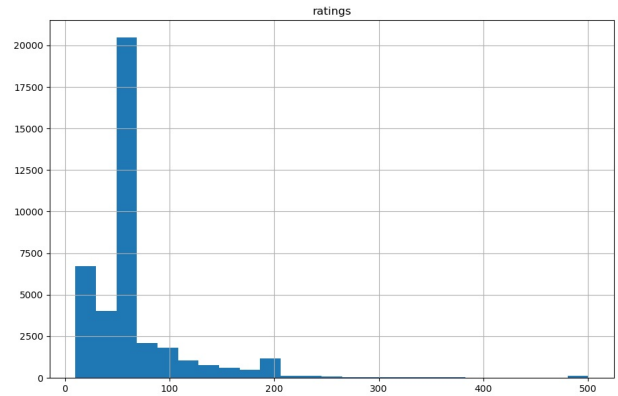
- Lets Show The Relationships Between Variables

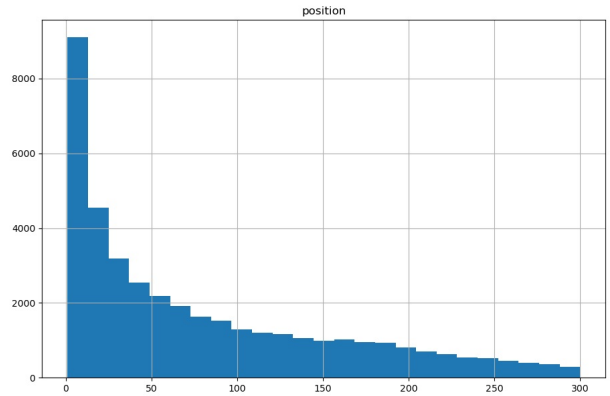
In [13]:
```
# Calculate Correlation
corr = data.drop(['ratings'], axis=1).corr()

# make the figure size
plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm')  # Add correlation values

plt.title("Correlation Heatmap")      # Add title
plt.show()
```

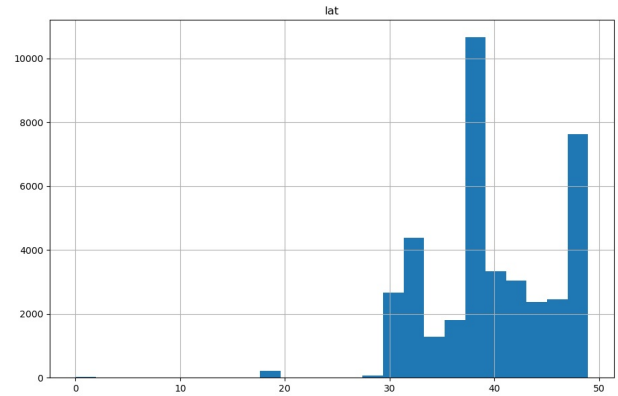


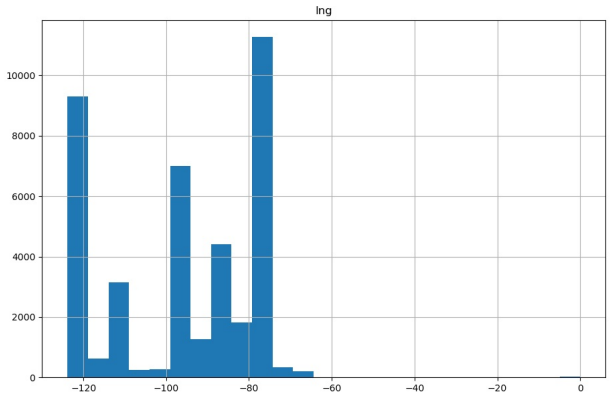
In [16]:
```
data.hist(column=['score','ratings','position','lat','lng'],figsize=(25,25),bins=25)
plt.show()
```





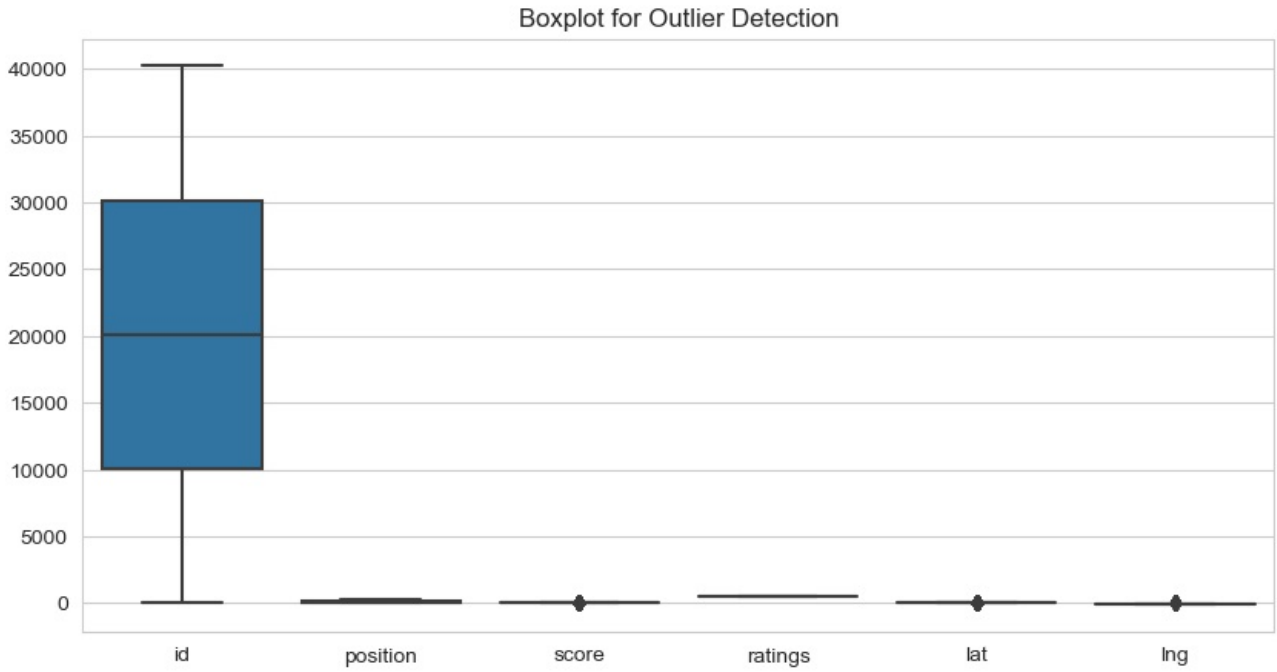






In [150…

```
#Check Outlires For numerical columns by boxpolt
num_cols = data.columns

plt.figure(figsize=(10, 5))
sns.boxplot(data=data[num_cols], orient='v')

plt.title('Boxplot for Outlier Detection')
plt.show()
```



In [153…
```
# what is cateogries of resturants
data['category'].values
```

Out[153]:
```
array(['Burgers, American, Sandwiches',
       'Coffee and Tea, Breakfast and Brunch, Bubble Tea',
       'American, Cheesesteak, Sandwiches, Alcohol', ...,
       'Sushi, Asian, Japanese, Exclusive to Eats, Group Friendly',
       'Mediterranean, Gluten Free Friendly, Allergy Friendly, Family Meals, Turkish, Greek, Middle Eastern, S
alads, Vegan Friendly, Vegetarian Friendly, Local Specialities',
       'Chinese, Asian, Asian Fusion, Family Friendly, Group Friendly'],
      dtype=object)
```

In [169…
```
var = data[data['category'].isin(['American','Sandwiches','Coffee and Tea','Japanese','Burgers','Asian'])]
var.head()
```

Out[169]:

	id	position	name	score	ratings	category	price_range	full_address	zip_code	lat	lng
551	552	23	Wild Burger (3400 Montgomery Hwy.)	4.1	500	Burgers	$$	3400 Montgomery Hwy, Dothan, AL, 36303	36303	31.25525	-85.43014
588	589	31	Songwriters Cafe (3320 Montgomery Highway)	4.6	500	American	$	3320 Montgomery Hwy, Dothan, AL, 36303	36303	31.25448	-85.42925
664	665	15	Steak 'n Shake (5901 University Drive, Suite I)	4.4	500	American	$$	5901 University Drive, Suite I, Huntsville, AL...	35806	34.73867	-86.66602
714	715	34	Wild Burger (7042 Highway 72 West)	4.6	500	Burgers	$$	7042 Highway 72 W, Huntsville, AL, 35806	35806	34.75441	-86.71043
1033	1034	58	Songwriters Cafe (3060 So. McKenzie Street)	4.6	500	American	$	3060 S McKenzie St, Foley, AL, 36535	36535	30.36941	-87.68412

In [55]:
```
## change id to restaurant_id
data.rename(columns={'id':'restaurant_id'}, inplace=True)
```

- What are the top 10 highest-score restaurants?

In [59]:
```
## list the top 10 restaurants in score
data[data['score']==np.max(data['score'])].head(10)
```

Out[59]:

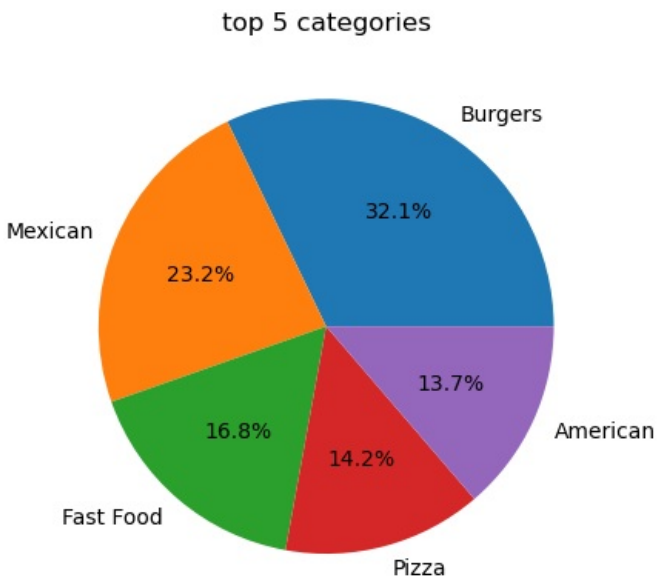
	restaurant_id	position	name	score	ratings	category	price_range	full_address	zip_code	lat	lng
15	16	88	Jeni's Splendid Ice Cream (Pepper Place)	5.0	500.0	Ice Cream & Frozen Yogurt, Comfort Food, D...	$$$	219 29th St S, Birmingham, AL, 35233	35233	33.516600	-86.789950
156	157	66	Wasabi Juan's (Downtown)	5.0	500.0	Sushi, Burritos, Tacos	$	2201 2nd Ave S Suite 105, Birmingham, AL, 35233	35233	33.512110	-86.799920
240	241	14	Honey Baked Ham (7001 Crestwood Blvd, Ste 114)	5.0	500.0	Sandwich, Family Meals	$	7001 Crestwood Blvd, Ste 114, Birmingham, AL, ...	35213	33.527938	-86.730360
262	263	8	Cookie Dough Magic	5.0	500.0	Dessert: Other, Ice Cream + Frozen Yogurt	$	400 41st St S, 102, Birmingham, AL, 35222	35222	33.522448	-86.773424
326	327	95	Papa Johns (736 Montgomery Hwy)	5.0	500.0	Pizza, Wings, Sandwiches, Desserts, American, ...	$	736 Montgomery Hwy, Vestavia Hills, AL, 35216	35216	33.444900	-86.791500
356	357	49	Mr. Lin Chinese Restaurant	5.0	500.0	Chinese, Asian, Asian Fusion	$$	475, Helena, AL, 35080	35080	33.279000	-86.851100
410	411	94	Great American Cookies (Riverchase Galleria)	5.0	500.0	Bakery, Desserts, Comfort Food	$	2000 Riverchase Galleria, Birmingham, AL, 35244	35244	33.379202	-86.808796
615	616	1	Tropical Smoothie Cafe - 3230 Ross Clark Circl...	5.0	500.0	Juice and Smoothies, Healthy, Fast Food	$	3230 Ross Clark Circle, Suite 3, Dothan, AL, 3...	36303	31.234995	-85.431238
632	633	10	Firehouse Subs (3255 South Oates Street. Suite 8)	5.0	500.0	Sandwich, Deli	$	3255 South Oates Street. Suite 8, Dothan, AL, ...	36301	31.179849	-85.401010
749	750	62	Hunt Brothers Pizza	5.0	500.0	American, Italian, Wings	$$	6090 Old Madison Pike NW, Huntsville, AL, 35806	35806	34.713470	-86.658061

- Which cuisine categories are most common in the dataset?

In [78]:
```
# list the top 5 categories
var = data['category'].value_counts()
print(var.head(5))
```

```
Burgers, American, Sandwiches           1606
Mexican, Latin American, New Mexican    1161
Fast Food, Sandwich, American            837
Pizza, American, Italian                 707
American, Burgers, Fast Food             685
Name: category, dtype: int64
```

In [84]:
```
category = ['Burgers','Mexican','Fast Food','Pizza','American']
nums     = [1606,1161,837,707,685]

# plot the pie chart
plt.pie(nums,labels=category,autopct='%1.1f%%')
plt.title('top 5 categories')

plt.show()
```



- Which zip codes have the most restaurants?

In [83]:
```
# zip codes have the most restaurants

var2 = data['zip_code'].value_counts()
print(var2.head(3))
```

```
98052    185
22314    185
22030    175
Name: zip_code, dtype: int64
```

- Is there a relationship between price range and rating?

In [93]:
```
# Group by price_range and get average rating
avg_rating_by_price = data.groupby('price_range')['score'].mean().sort_index()

print(avg_rating_by_price)
```

```
price_range
$       4.582227
$$      4.566983
$$$     4.625532
$$$$    4.603333
Name: score, dtype: float64
```

In [99]:
```
from scipy.stats import f_oneway

# Group scores by price range
grouped_scores = [group['score'].values for name, group in data.groupby('price_range') if len(group) > 1]

# Perform ANOVA test
anova_result = f_oneway(*grouped_scores)

print(f"F-statistic: {anova_result.statistic}")
print(f"P-value: {anova_result.pvalue}")
```

```
F-statistic: 15.042281486878323
P-value: 8.802519620599879e-10
```

- The Answer :- Yes, There is a statistically significant relationship between price_range and score

7- Conclusion:-

We explore the Ubereats dataset and learn more about data attributes then jump into how to visualize the data with Exploratory Data Analysis.

- We saw some basic and advanced level charts of seaborn and matplotlib like (Pie-chart, Bar chart, Countplot). ### - Questions we answered : -

What are the top 10 highest-rated restaurants?

Which cuisine categories are most common in the dataset?

Is there a relationship between price range and rating?

Which zip codes have the most restaurants?

Export cleaning dataset to csv for using in power bi

In [186…
```
## save the data and go to power bi report
data.to_csv('cleaned_uber_eats.csv', index=False)
```

- Done