***Anomaly detection using oneclass SVM***

# 1. Loading Data into Pandas DataFrame



**2. Getting Rid of NANs**

**import** **numpy** **as** **np**

**import** **seaborn** **as** **sns**

**import** **matplotlib**

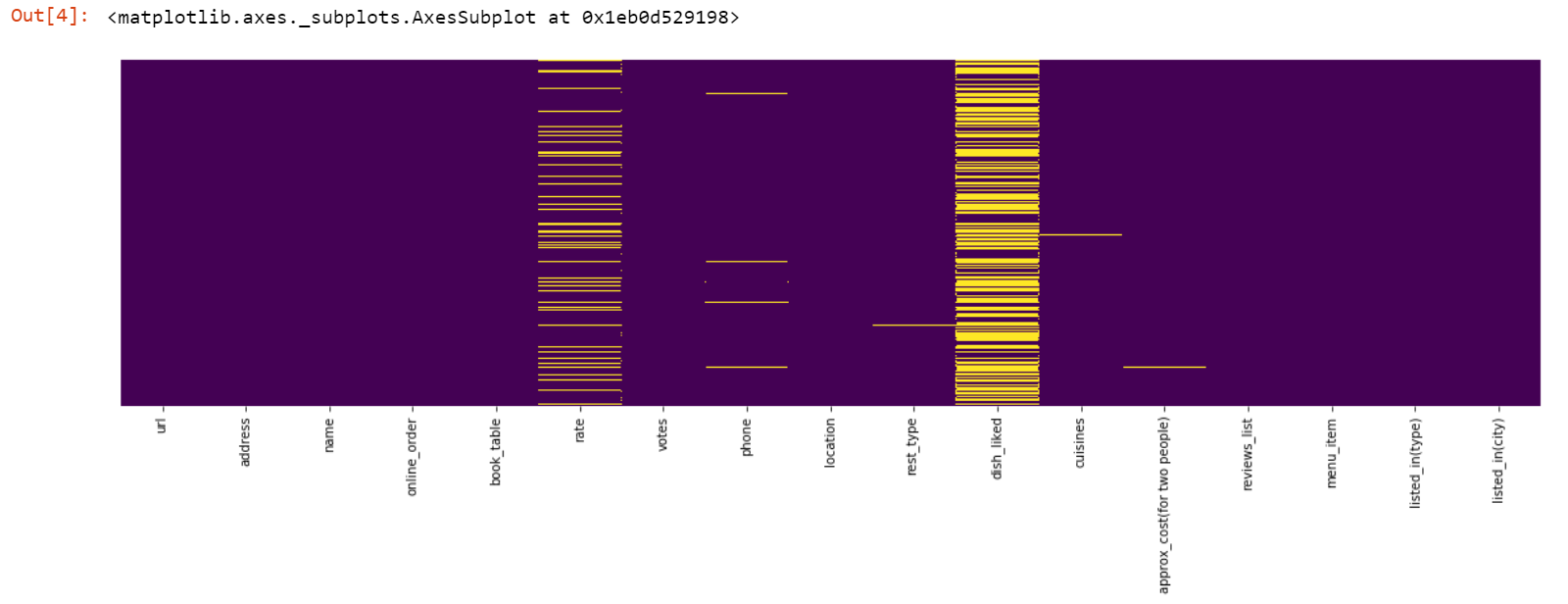
**import** **matplotlib.pyplot** **as** **plt**

%**matplotlib** inline

**import** **seaborn** **as** **sns**

fig, ax = plt.subplots(figsize=(20,5))

sns.heatmap(df.isnull(),yticklabels=**False**,cbar=**False**,cmap='viridis')

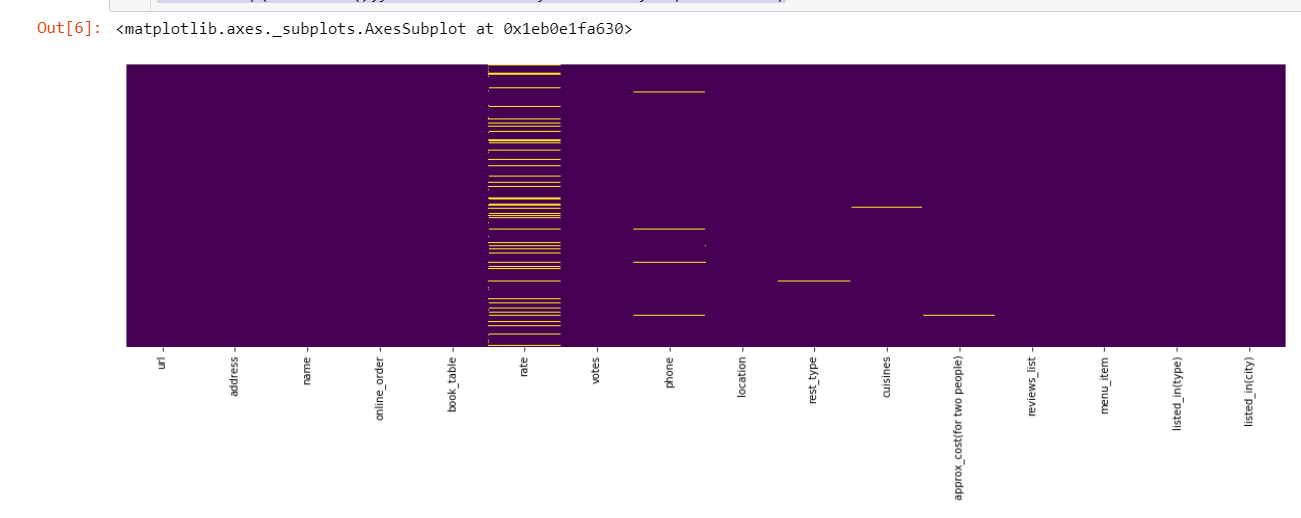


If we look at the distribution of NANs.Most of NANs are present in dish\_liked variables. WE are going to drop that column anyways as its not going to add any significant value to our algorithm.

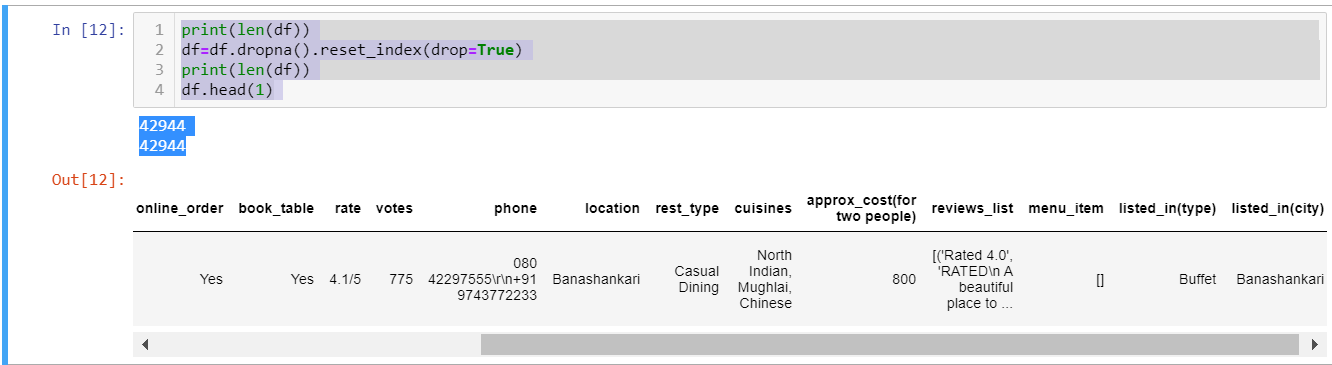
df=df.drop('dish\_liked',axis=1)

fig, ax = plt.subplots(figsize=(20,5))

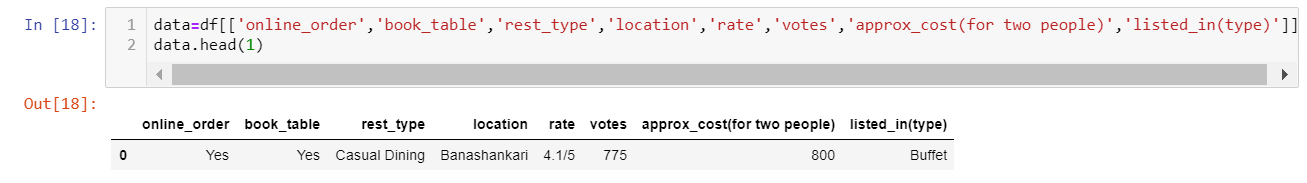
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')



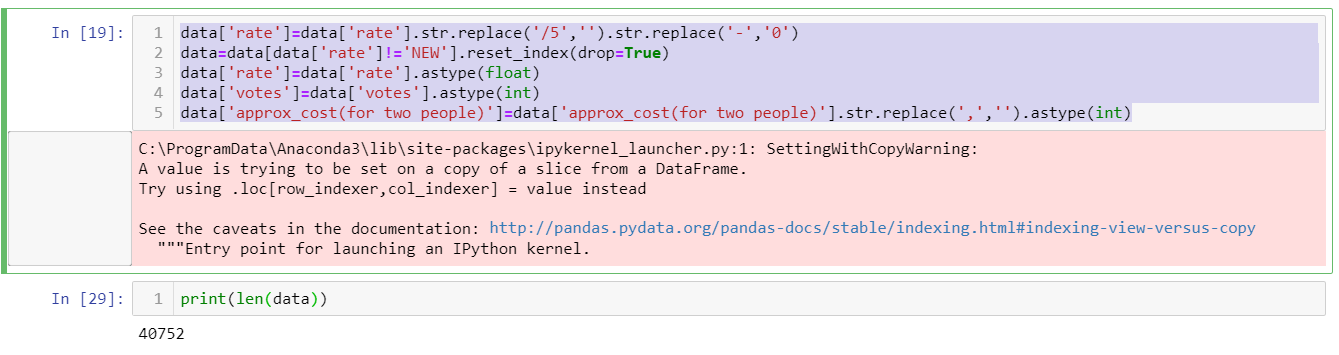
Since We are going to perform Anomaly detection. Its better to get rid of the NAN or null data as if we fill them somehow, those are oing to appear as anomalies as well in our algo. But we want the anomalies that we are real. Even if those do not appear as anomalies those are going to disrupt the algoritham.



# 3. Pre-processing of the Data



Now We are going to convert our data in the desired formats.



We are going to convert our categorical and ordinal variables into numerical data so that our algoritham is able to work on those.



# 4. Model Design, Fitting and prediction

One Class SVM is basically a type of Support Vector Machines Algoritham. This Algoritham is basically designed to find the outliers or anomalies in the data.

from sklearn.svm import OneClassSVM

ocsvm\_linear=OneClassSVM(nu=0.04,gamma=0.5,kernel='linear')

ocsvm\_linear.fit(data)

results\_linear=ocsvm\_linear.predict(data)

ocsvm\_rbf=OneClassSVM(nu=0.04,gamma=0.5,kernel='linear')

ocsvm\_rbf.fit(data)

results\_rbf=ocsvm\_rbf.predict(data)

nu decides the amount of percentage that we want to find out from the data as outliers. Gamma basically decides the smoothing of the contour lines.This Model basically is going to tell us that which observation in the datasets are going to be outliers and we are going to index it agains our parent datasets that we used

df=pd.read\_csv('zomato.csv')

df=df.drop('dish\_liked',axis=1)

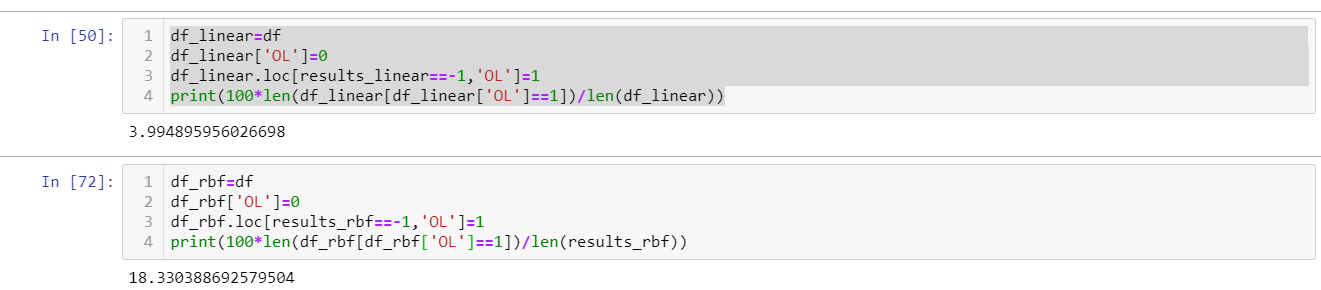
df=df.dropna().reset\_index(drop=True)

df['rate']=df['rate'].str.replace('/5','').str.replace('-','0')

df=df[df['rate']!='NEW'].reset\_index(drop=True)

print(len(df))

We are going to define an outlier column in our parent dataset and 0 wil mean that specific measurement is not an outier based on our analysis in the dataset while 1 means that point is an outlier. We will then map our results against each model to change the value to 1 based on the detection of outli-ers (anomalies)

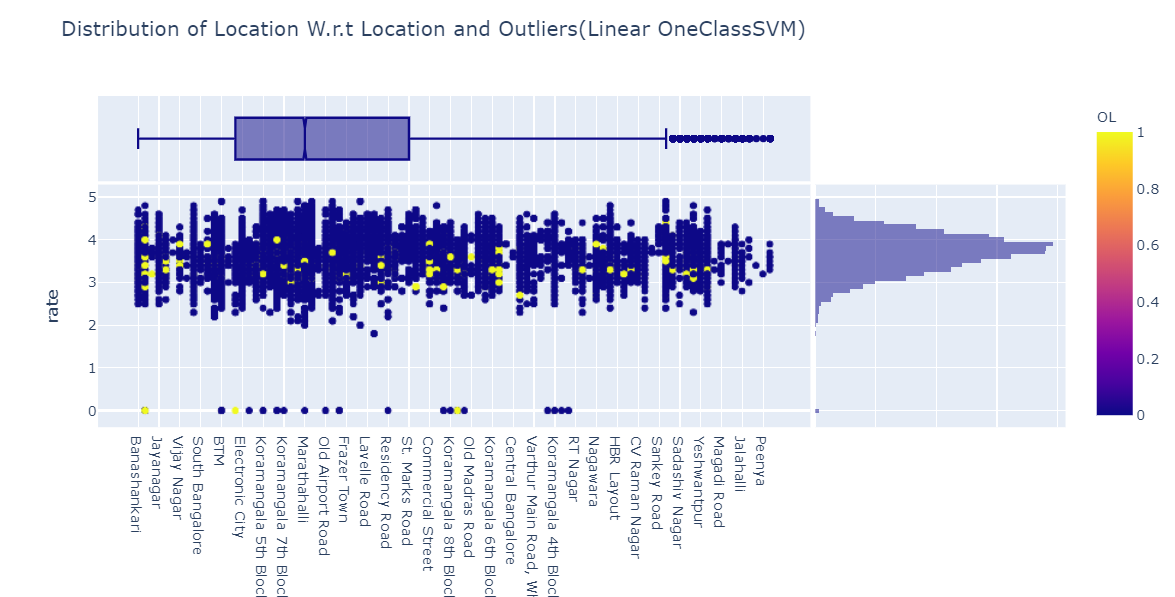


import plotly.express as px

fig2= px.scatter(df\_linear, x='location', y='rate',color="OL", title='Distribution of Location W.r.t Location and Outliers(RBF OneClassSVM)',

marginal\_y="histogram", marginal\_x="box")

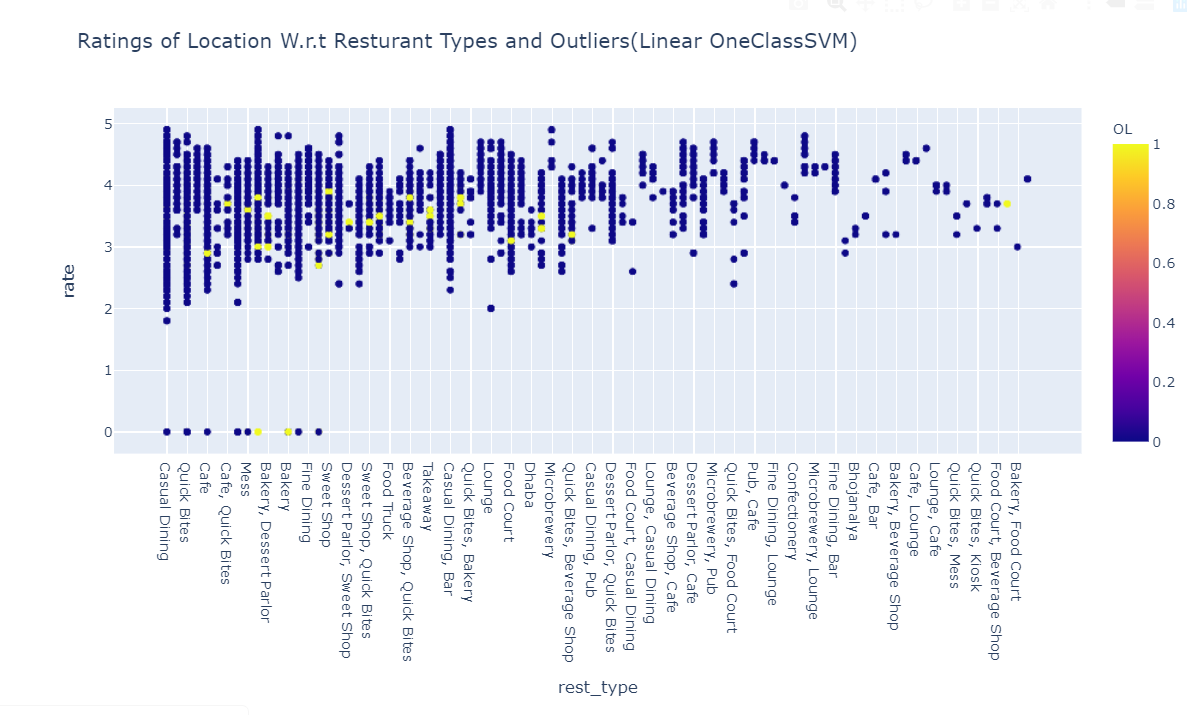
fig2.show()



import plotly.express as px

fig3 = px.scatter(df\_linear, x='rest\_type', y='rate',color="OL", title='Ratings of Location W.r.t Resturant Types and Outliers(Linear OneClassSVM)')

fig3.show()



import plotly.express as px

fig5 = px.scatter(df\_linear, x='rest\_type', y='approx\_cost(for two people)',color="OL", title='Cost of 2 people W.r.t Resturant Types and Outliers(linear)')

fig5.show()

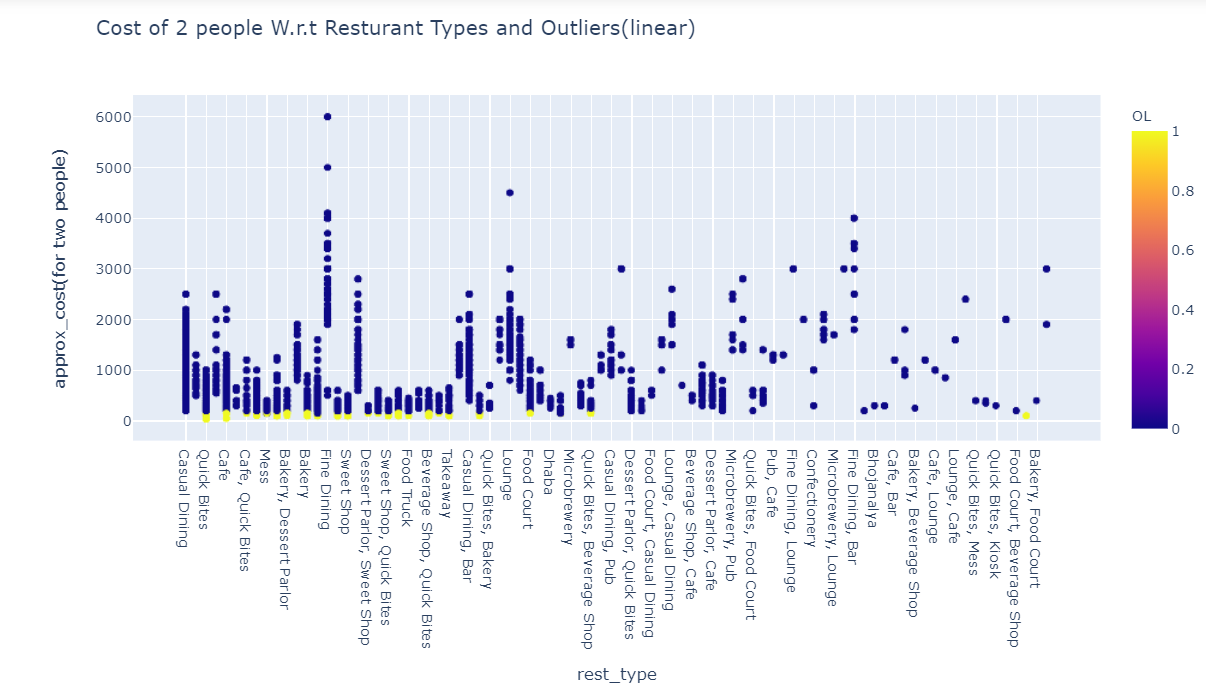


fig7 = px.scatter\_matrix(df\_linear, dimensions=["rate", "approx\_cost(for two people)", "votes",], color="OL", title='Distribution matrix(linear)')

fig7.show()

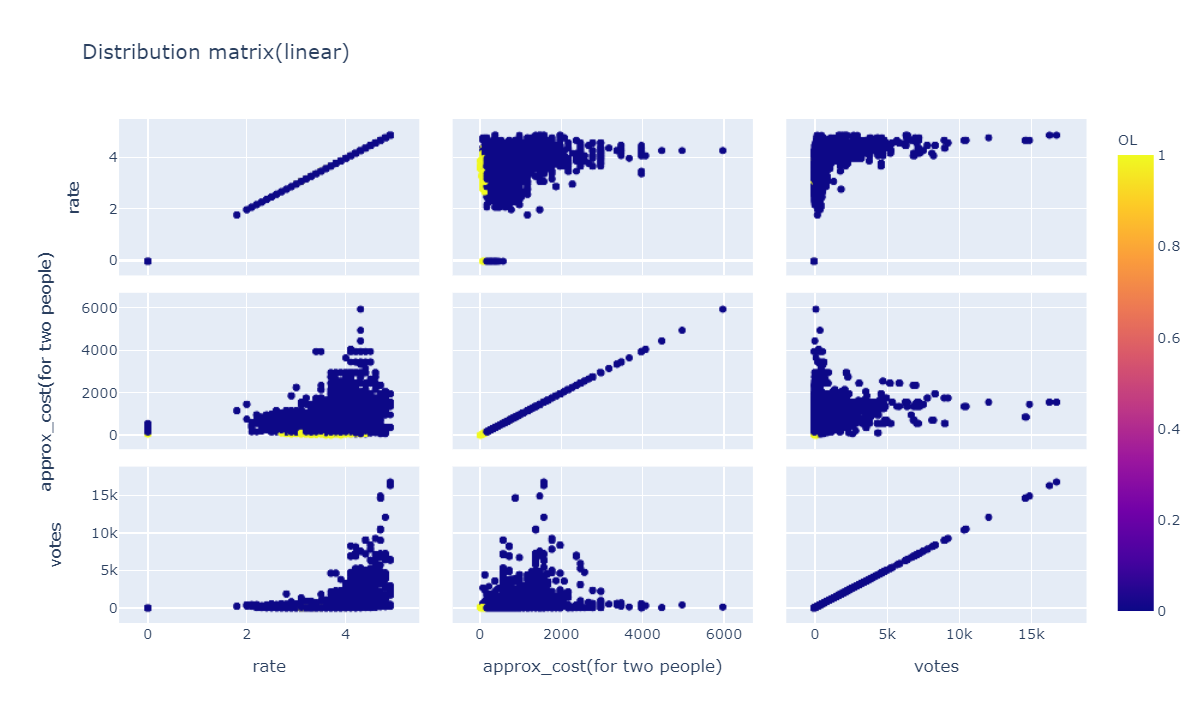


fig9 = px.scatter\_3d(df\_linear, x='votes', y='rate', z='approx\_cost(for two people)',color='OL',symbol='OL', opacity=0.7, title='3D view to show Outliers w.r.t Cost, Rating & Vote')

fig9.show()

