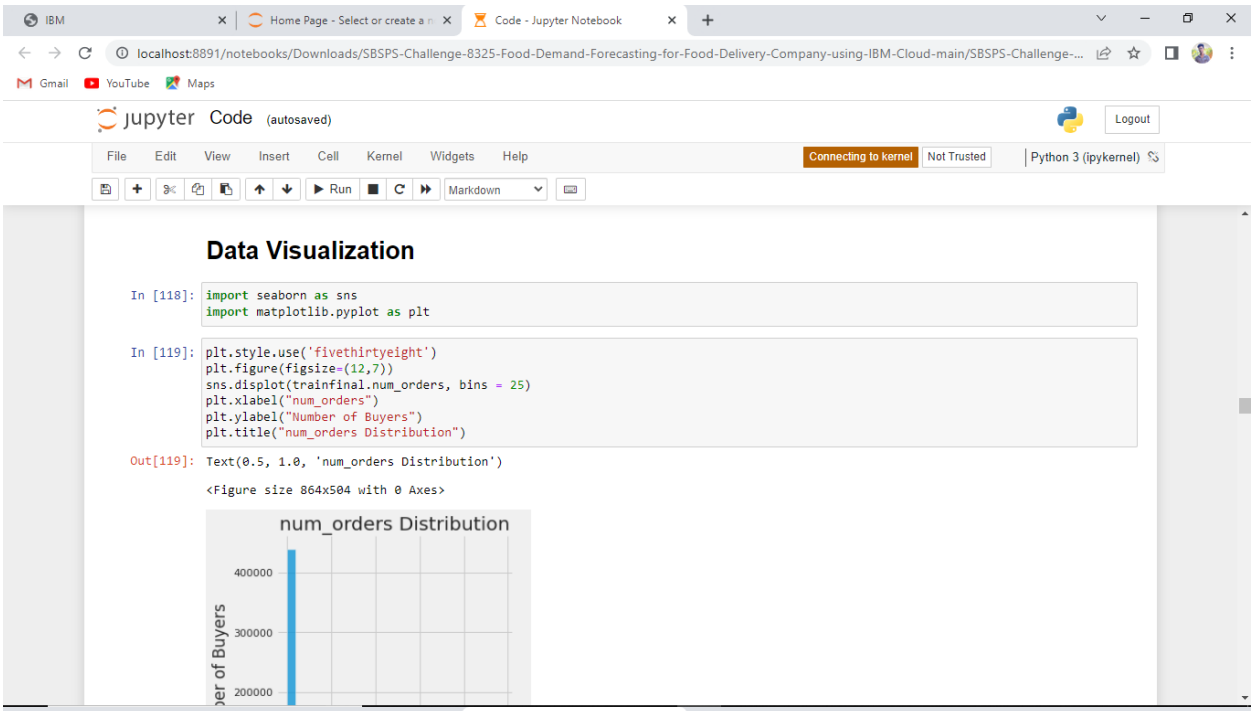
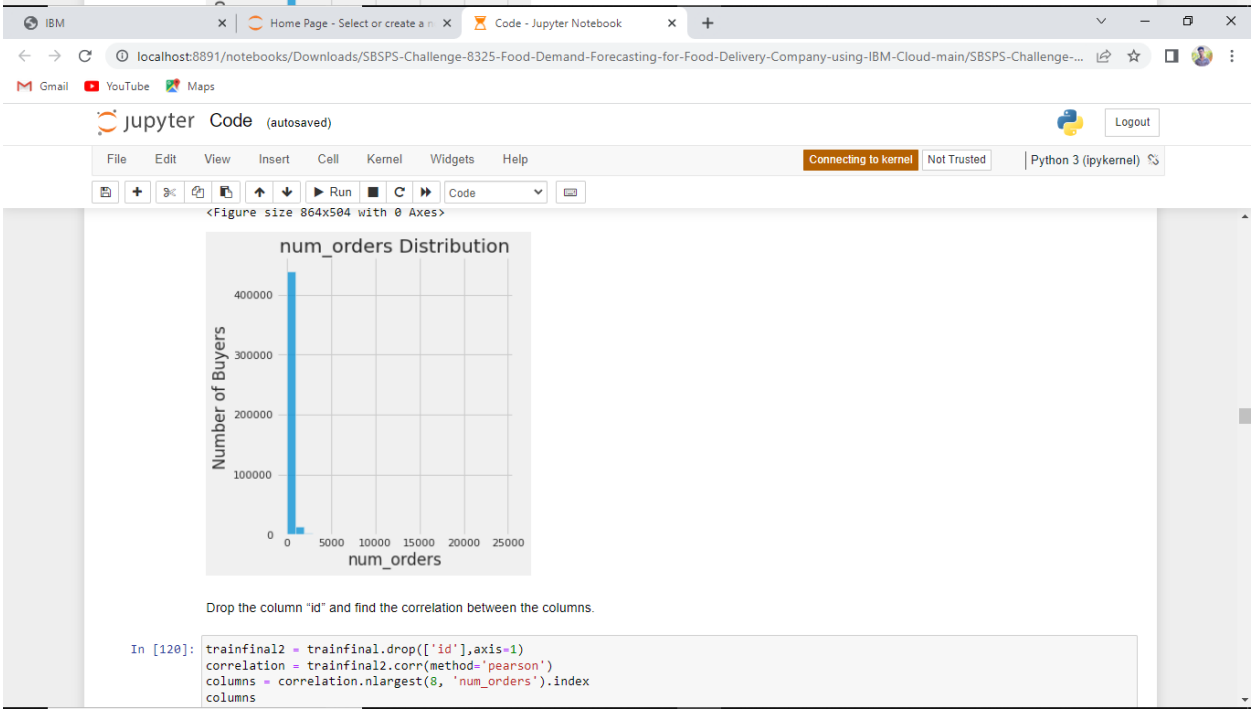
TEAM ID: PNT2022TMID32368

PROJECT NAME: DemandEst - AI powered Food Demand Forecaster

Team Leader





IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

Gmail YouTube Maps

jupyter Code (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

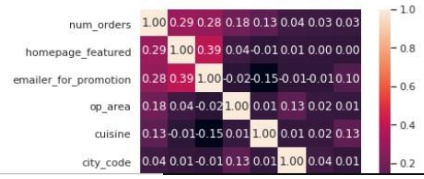
Run Code

In [120]:
```
trainfinal2 = trainfinal.drop(['id'],axis=1)
correlation = trainfinal2.corr(method='pearson')
columns = correlation.nlargest(8, 'num_orders').index
columns
```

Out[120]:
```
Index(['num_orders', 'homepage_featured', 'emailer_for_promotion', 'op_area',
       'cuisine', 'city_code', 'region_code', 'category'],
      dtype='object')
```

Correlation is a statistical relationship between two variables and it could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease. With the help of seaborn heatmap we will be plotting the heatmap and for finding the correlation between variable we have corr() available.

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```



IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

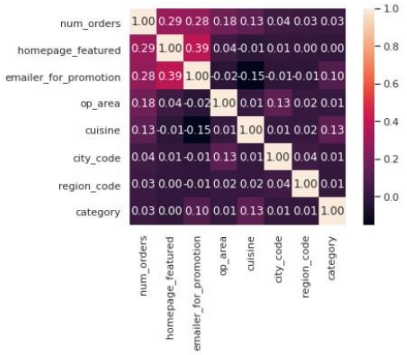
Gmail YouTube Maps

jupyter Code (autosaved) Logout

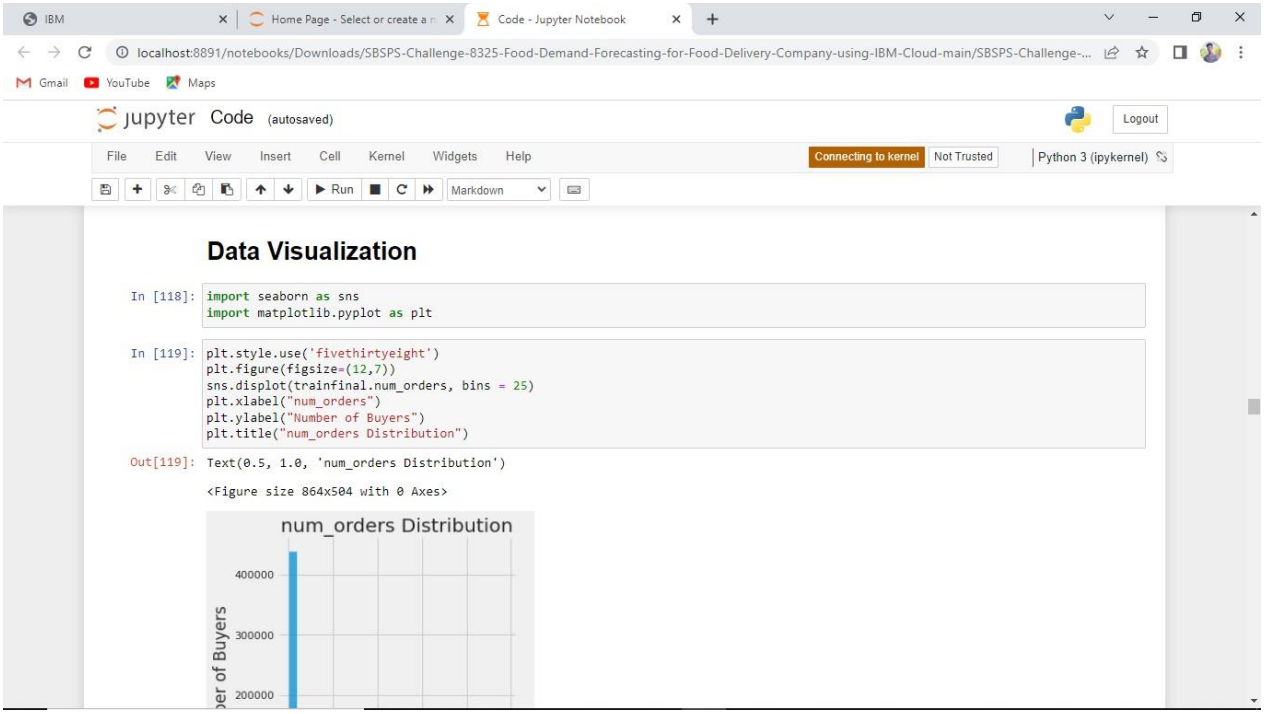
File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

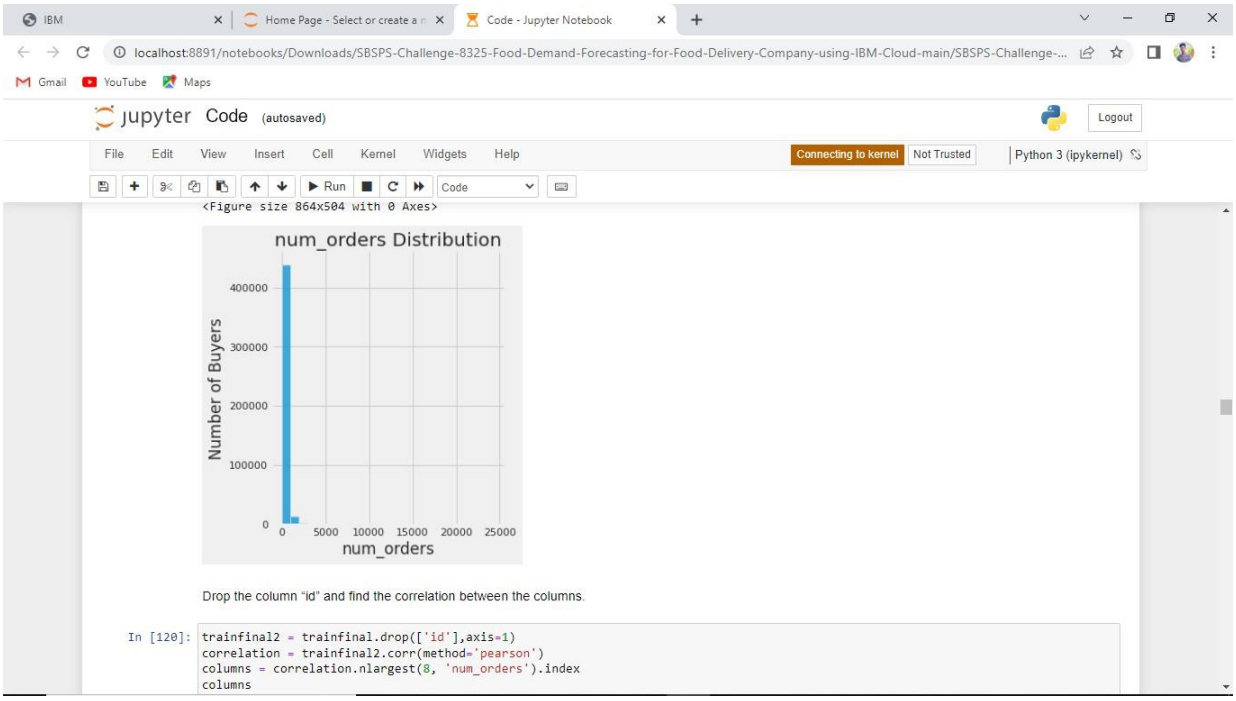
Run Code

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```



Team Member 1





IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

Gmail YouTube Maps

jupyter Code (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

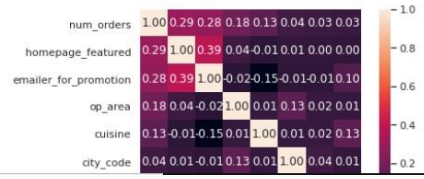
Run Code

In [120]:
```
trainfinal2 = trainfinal.drop(['id'],axis=1)
correlation = trainfinal2.corr(method='pearson')
columns = correlation.nlargest(8, 'num_orders').index
columns
```

Out[120]:
```
Index(['num_orders', 'homepage_featured', 'emailer_for_promotion', 'op_area',
       'cuisine', 'city_code', 'region_code', 'category'],
      dtype='object')
```

Correlation is a statistical relationship between two variables and it could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease. With the help of seaborn heatmap we will be plotting the heatmap and for finding the correlation between variable we have corr() available.

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```



IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

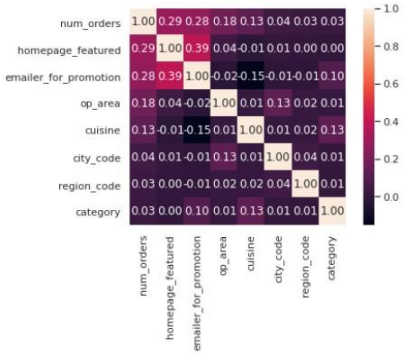
Gmail YouTube Maps

jupyter Code (autosaved) Logout

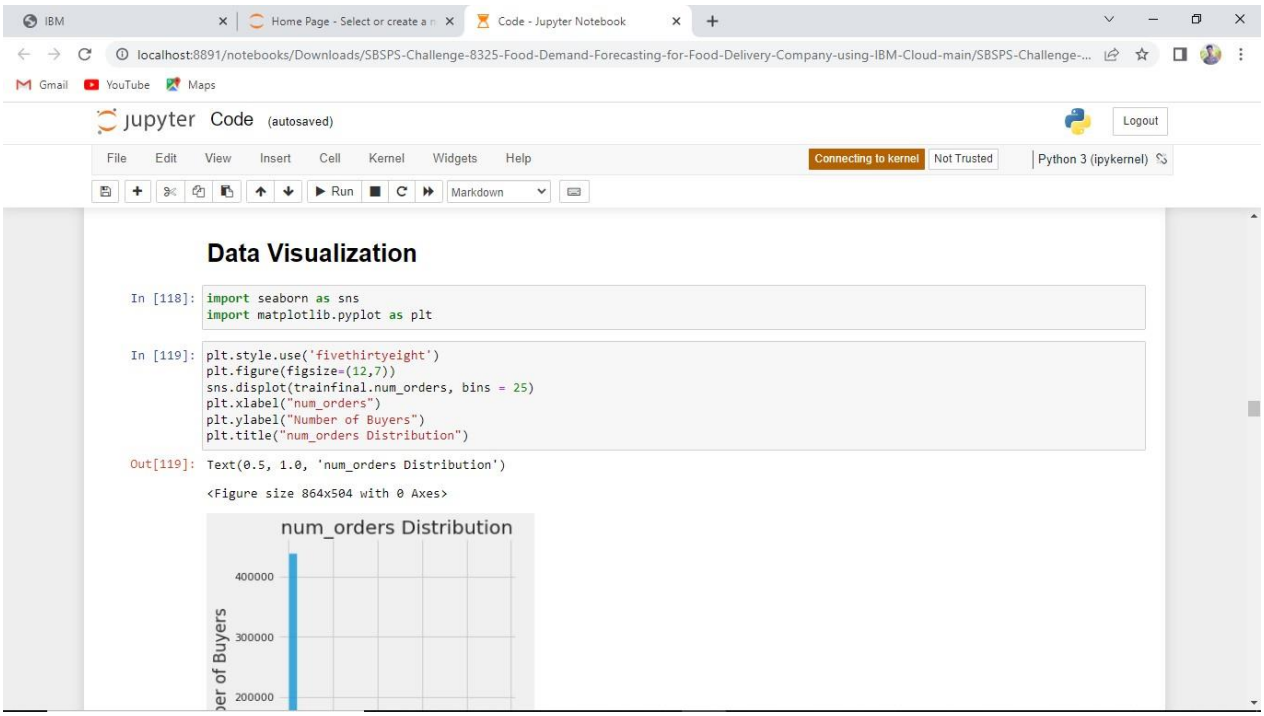
File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

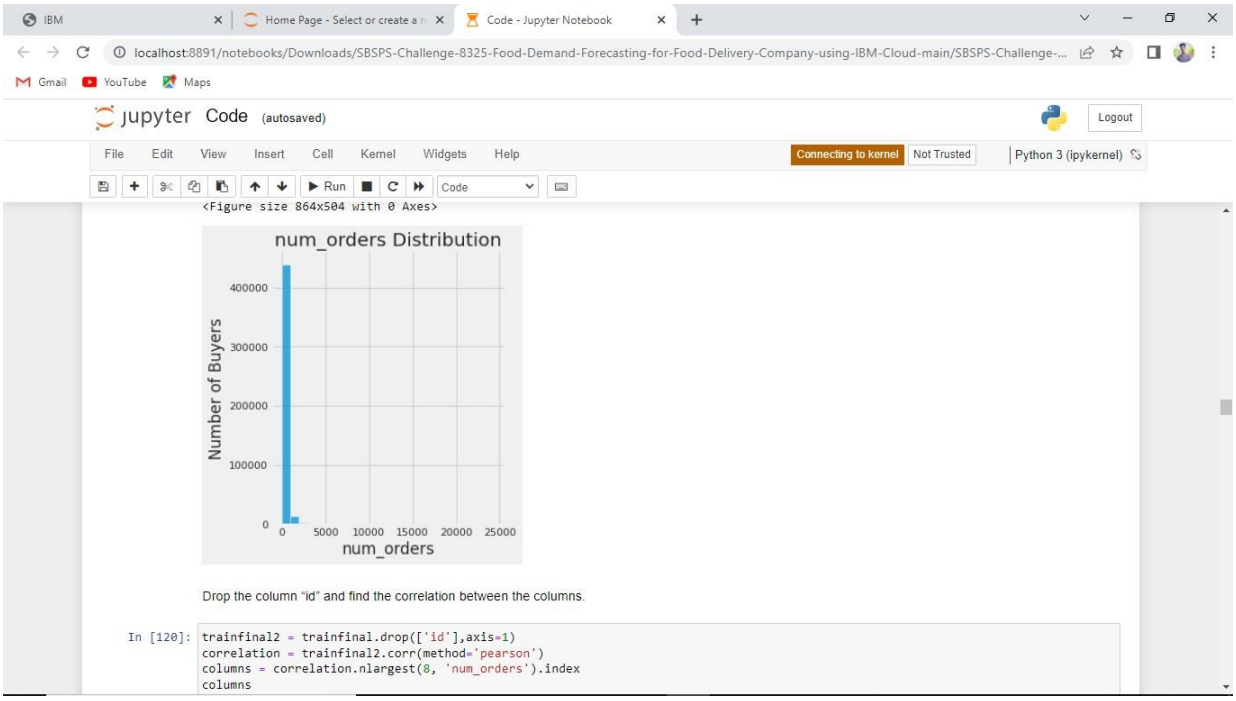
Run Code

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```



Team Member 2





IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

Gmail YouTube Maps

jupyter Code (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

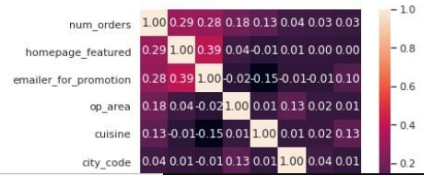
Run Code

In [120]:
```
trainfinal2 = trainfinal.drop(['id'],axis=1)
correlation = trainfinal2.corr(method='pearson')
columns = correlation.nlargest(8, 'num_orders').index
columns
```

Out[120]:
```
Index(['num_orders', 'homepage_featured', 'emailer_for_promotion', 'op_area',
       'cuisine', 'city_code', 'region_code', 'category'],
      dtype='object')
```

Correlation is a statistical relationship between two variables and it could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease. With the help of seaborn heatmap we will be plotting the heatmap and for finding the correlation between variable we have corr() available.

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```



IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

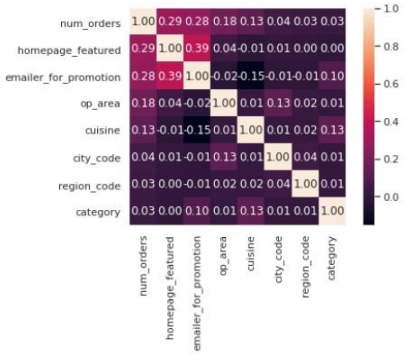
Gmail YouTube Maps

jupyter Code (autosaved) Logout

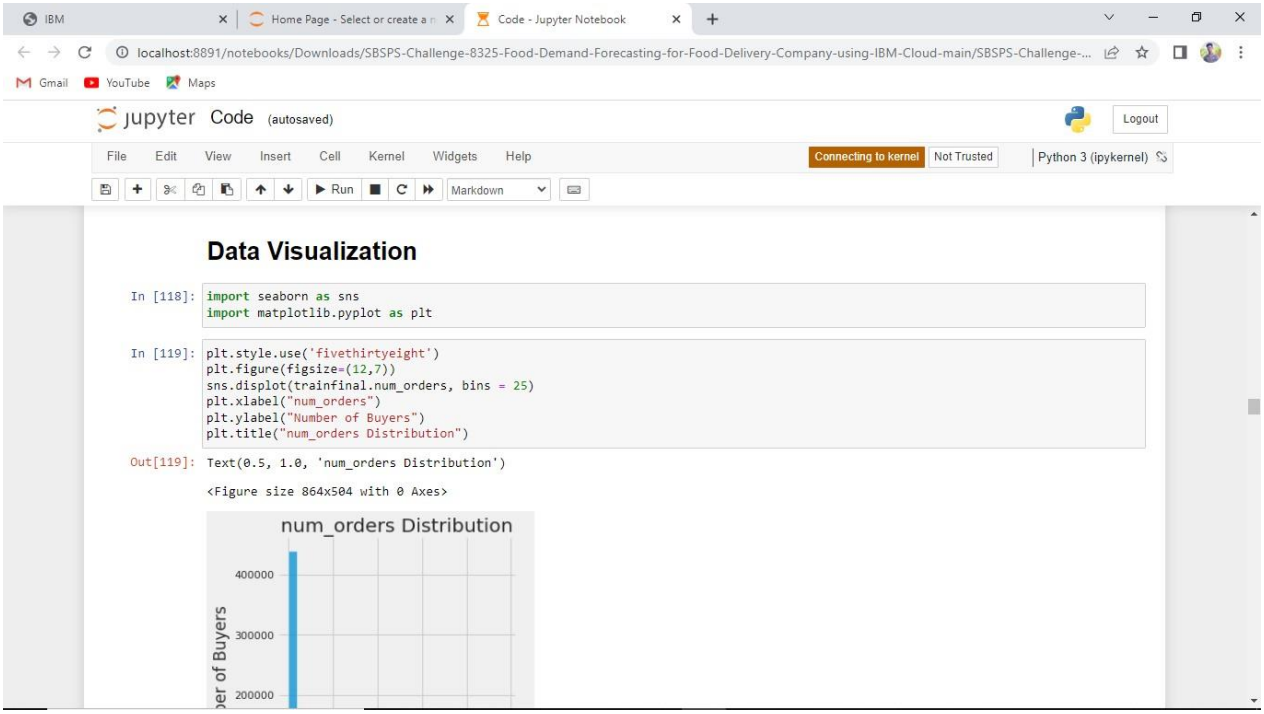
File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

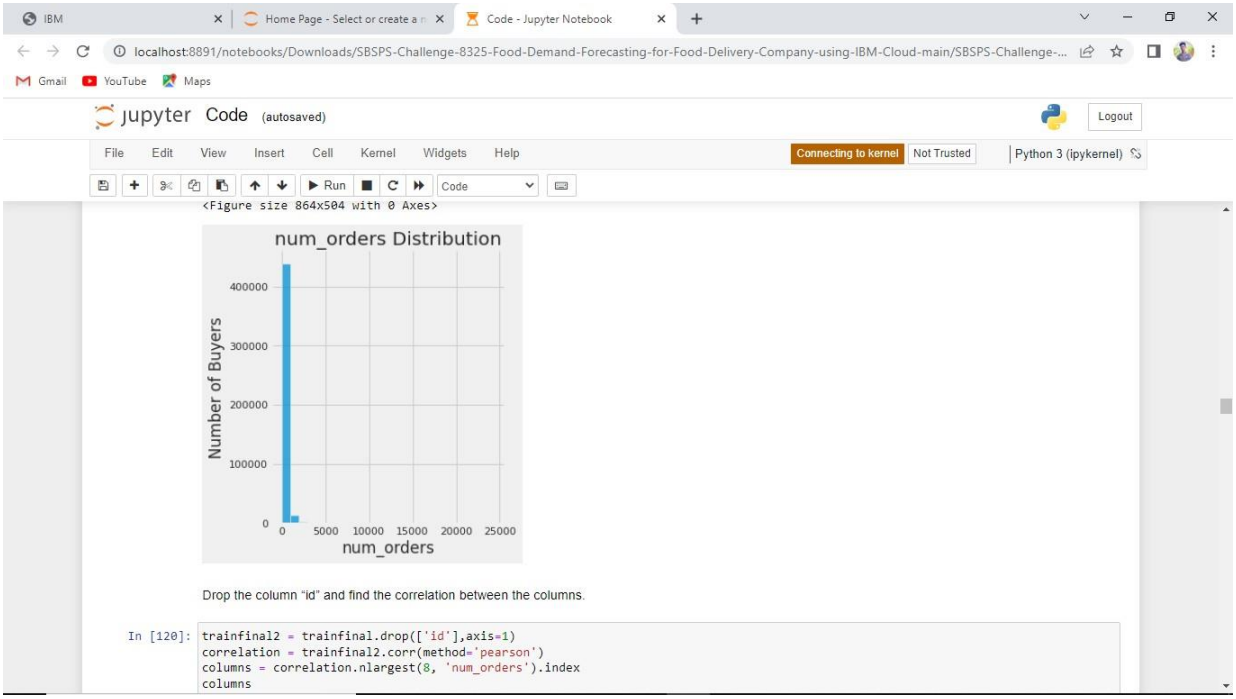
Run Code

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```



Team Member 3





IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

Gmail YouTube Maps

jupyter Code (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

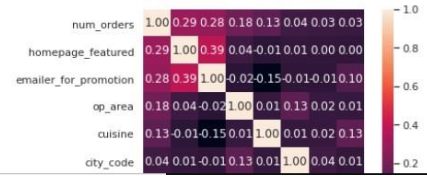
Run Code

In [120]:
```
trainfinal2 = trainfinal.drop(['id'],axis=1)
correlation = trainfinal2.corr(method='pearson')
columns = correlation.nlargest(8, 'num_orders').index
columns
```

Out[120]:
```
Index(['num_orders', 'homepage_featured', 'emailer_for_promotion', 'op_area',
       'cuisine', 'city_code', 'region_code', 'category'],
      dtype='object')
```

Correlation is a statistical relationship between two variables and it could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease. With the help of seaborn heatmap we will be plotting the heatmap and for finding the correlation between variable we have corr() available.

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```



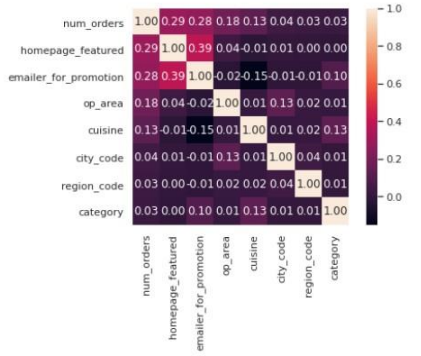
IBM | Home Page - Select or create a n | Code - Jupyter Notebook

localhost:8891/notebooks/Downloads/SBSPS-Challenge-8325-Food-Demand-Forecasting-for-Food-Delivery-Company-using-IBM-Cloud-main/SBSPS-Challenge-...

Gmail YouTube Maps

jupyter Code (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Connecting to kernel Not Trusted Python 3 (ipykernel)

Run Code

In [121]:
```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f',
                      yticklabels=columns.values, xticklabels=columns.values)
```