['id', 'week', 'checkout\_price', 'base\_price', 'emailer\_for\_promotion', 'homepage\_featured', 'num\_orders', 'category', 'cuisine', 'city\_code', 'region\_code', 'center\_type', 'op\_area']

In [27]:

cols **=** cols[:2] **+** cols[9:] **+** cols[7:9] **+** cols[2:7]

print(cols)

['id', 'week', 'city\_code', 'region\_code', 'center\_type', 'op\_area', 'category', 'cuisine', 'checkout\_price', 'base\_price', 'emailer\_for\_promotion', 'homepage\_featured', 'num\_orders']

In [28]:

trainfinal **=** trainfinal[cols]

trainfinal**.**dtypes

Out[28]:

id int64

week int64

city\_code int64

region\_code int64

center\_type object

op\_area float64

category object

cuisine object

checkout\_price float64

base\_price float64

emailer\_for\_promotion int64

homepage\_featured int64

num\_orders int64

dtype: object

In [30]:

**from** sklearn.preprocessing **import** LabelEncoder

In [32]:

lb1 **=** LabelEncoder()

trainfinal['center\_type'] **=** lb1**.**fit\_transform(trainfinal['center\_type'])

lb2 **=** LabelEncoder()

trainfinal['category'] **=** lb1**.**fit\_transform(trainfinal['category'])

lb3 **=** LabelEncoder()

trainfinal['cuisine'] **=** lb1**.**fit\_transform(trainfinal['cuisine'])

In [33]:

trainfinal**.**head()

Out[33]:

|  | **id** | **week** | **city\_code** | **region\_code** | **center\_type** | **op\_area** | **category** | **cuisine** | **checkout\_price** | **base\_price** | **emailer\_for\_promotion** | **homepage\_featured** | **num\_orders** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1379560 | 1 | 647 | 56 | 2 | 2.0 | 0 | 3 | 136.83 | 152.29 | 0 | 0 | 177 |
| **1** | 1018704 | 2 | 647 | 56 | 2 | 2.0 | 0 | 3 | 135.83 | 152.29 | 0 | 0 | 323 |
| **2** | 1196273 | 3 | 647 | 56 | 2 | 2.0 | 0 | 3 | 132.92 | 133.92 | 0 | 0 | 96 |
| **3** | 1116527 | 4 | 647 | 56 | 2 | 2.0 | 0 | 3 | 135.86 | 134.86 | 0 | 0 | 163 |
| **4** | 1343872 | 5 | 647 | 56 | 2 | 2.0 | 0 | 3 | 146.50 | 147.50 | 0 | 0 | 215 |

trainfinal**.** shape

Out[34]:

(456548, 13)

In [36]:

plt**.**style**.**use('fivethirtyeight')

plt**.**figure(figsize**=**(12,7))

sns**.**distplot(trainfinal**.**num\_orders, bins **=** 25)

plt**.**xlabel("num\_orders")

plt**.**ylabel("Number of Buyers")

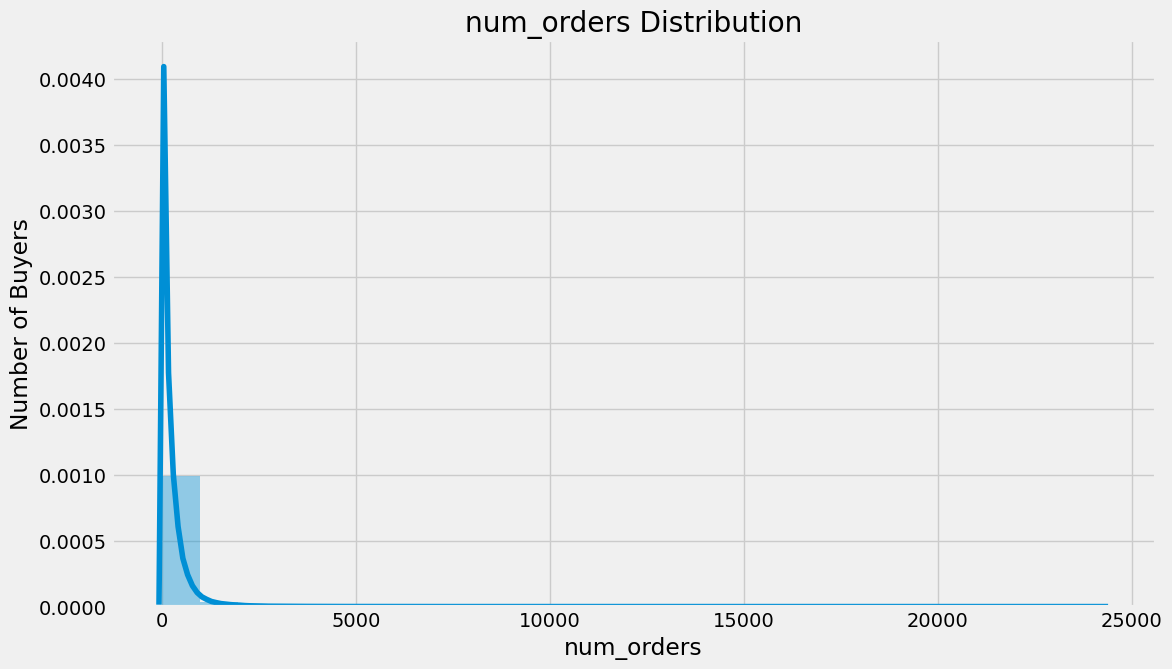
plt**.**title("num\_orders Distribution")

C:\Users\NANDHITHA\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

Out[36]:

Text(0.5, 1.0, 'num\_orders Distribution')



In [37]:

trainfinal2 **=** trainfinal**.**drop(['id'], axis**=**1)

correlation **=** trainfinal2**.**corr(method**=**'pearson')

columns **=** correlation**.**nlargest (8, 'num\_orders')**.**index

columns

Out[37]:

Index(['num\_orders', 'homepage\_featured', 'emailer\_for\_promotion', 'op\_area',

'cuisine', 'city\_code', 'region\_code', 'category'],

dtype='object')

In [39]:

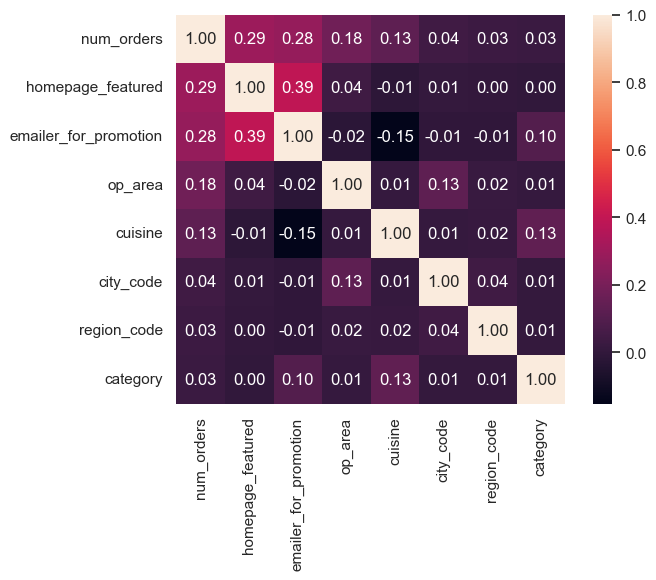
correlation\_map **=** np**.**corrcoef(trainfinal2[columns]**.**values**.**T)

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)

plt**.**show()



In [41]:

features **=** columns**.**drop(['num\_orders'])

trainfinal3 **=** trainfinal[features]

X **=** trainfinal3**.**values

y **=** trainfinal[ 'num\_orders']**.**values

In [42]:

trainfinal3**.**head()

Out[42]:

|  | **homepage\_featured** | **emailer\_for\_promotion** | **op\_area** | **cuisine** | **city\_code** | **region\_code** | **category** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | 0 | 2.0 | 3 | 647 | 56 | 0 |
| **1** | 0 | 0 | 2.0 | 3 | 647 | 56 | 0 |
| **2** | 0 | 0 | 2.0 | 3 | 647 | 56 | 0 |
| **3** | 0 | 0 | 2.0 | 3 | 647 | 56 | 0 |
| **4** | 0 | 0 | 2.0 | 3 | 647 | 56 | 0 |

In [45]:

**from** sklearn.model\_selection **import** train\_test\_split

In [47]:

X\_train, X\_val, y\_train, y\_val **=** train\_test\_split(X, y, test\_size**=**0.25)