Madhur Jaripatke

Roll No. 48

TE A Computer

RMDSSOE, Warje, Pune

1. Data Wrangling, I

Perform the following operations using Python on any open source dataset (e.g., data.csv) the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the

1. Import all the required Python Libraries.
2. Locate an open source data from the web (e.g. https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).
3. Load the Dataset into pandas data frame.
4. Data Preprocessing: check for missing values in the data using pandas insult(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.
5. Data Formatting and Data Normalization: Summarize the types of variables by checking

data set. If variables are not in the correct data type, apply proper type conversions. 6. Turn categorical variables into quantitative variables in Python. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set.

In [1]:
```
import pandas as pd
from sklearn import preprocessing
```

Import the dataset

In [2]:
```
df = pd.read_csv('Datasets/Uber Request Data.csv')
```

In [3]:
```
df
```

Out[3]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	619	Airport	1.0	Trip Completed	11/7/2016 11:51	11/7/2016 13:00
1	867	Airport	1.0	Trip Completed	11/7/2016 17:57	11/7/2016 18:47
2	1807	City	1.0	Trip Completed	12/7/2016 9:17	12/7/2016 9:58
3	2532	Airport	1.0	Trip Completed	12/7/2016 21:08	12/7/2016 22:03
4	3112	City	1.0	Trip Completed	13-07-2016 08:33:16	13-07-2016 09:25:47
...	...	...	...	...	...	...
6740	6745	City	NaN	No Cars Available	15-07-2016 23:49:03	NaN
6741	6752	Airport	NaN	No Cars Available	15-07-2016 23:50:05	NaN
6742	6751	City	NaN	No Cars Available	15-07-2016 23:52:06	NaN
6743	6754	City	NaN	No Cars Available	15-07-2016 23:54:39	NaN
6744	6753	Airport	NaN	No Cars Available	15-07-2016 23:55:03	NaN

6745 rows × 6 columns

Exploratory Data Analysis

In [4]:
```
df.head()
```

Out[4]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	619	Airport	1.0	Trip Completed	11/7/2016 11:51	11/7/2016 13:00
1	867	Airport	1.0	Trip Completed	11/7/2016 17:57	11/7/2016 18:47
2	1807	City	1.0	Trip Completed	12/7/2016 9:17	12/7/2016 9:58
3	2532	Airport	1.0	Trip Completed	12/7/2016 21:08	12/7/2016 22:03
4	3112	City	1.0	Trip Completed	13-07-2016 08:33:16	13-07-2016 09:25:47

In [5]:
```
df.head(7)
```

Out[5]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	619	Airport	1.0	Trip Completed	11/7/2016 11:51	11/7/2016 13:00
1	867	Airport	1.0	Trip Completed	11/7/2016 17:57	11/7/2016 18:47
2	1807	City	1.0	Trip Completed	12/7/2016 9:17	12/7/2016 9:58
3	2532	Airport	1.0	Trip Completed	12/7/2016 21:08	12/7/2016 22:03
4	3112	City	1.0	Trip Completed	13-07-2016 08:33:16	13-07-2016 09:25:47
5	3879	Airport	1.0	Trip Completed	13-07-2016 21:57:28	13-07-2016 22:28:59
6	4270	Airport	1.0	Trip Completed	14-07-2016 06:15:32	14-07-2016 07:13:15

In [6]:
```
df.tail()
```

Out[6]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
6740	6745	City	NaN	No Cars Available	15-07-2016 23:49:03	NaN
6741	6752	Airport	NaN	No Cars Available	15-07-2016 23:50:05	NaN
6742	6751	City	NaN	No Cars Available	15-07-2016 23:52:06	NaN
6743	6754	City	NaN	No Cars Available	15-07-2016 23:54:39	NaN
6744	6753	Airport	NaN	No Cars Available	15-07-2016 23:55:03	NaN

In [7]:
```
df.tail(2)
```

Out[7]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
6743	6754	City	NaN	No Cars Available	15-07-2016 23:54:39	NaN
6744	6753	Airport	NaN	No Cars Available	15-07-2016 23:55:03	NaN

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6745 entries, 0 to 6744
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Request id         6745 non-null   int64  
 1   Pickup point       6745 non-null   object 
 2   Driver id          4095 non-null   float64
 3   Status             6745 non-null   object 
 4   Request timestamp  6745 non-null   object 
 5   Drop timestamp     2831 non-null   object 
dtypes: float64(1), int64(1), object(4)
memory usage: 316.3+ KB
```

In [9]:
```
df.columns.values
```

Out[9]:
```
array(['Request id', 'Pickup point', 'Driver id', 'Status',
       'Request timestamp', 'Drop timestamp'], dtype=object)
```

In [10]:
```
df.shape
```

Out[10]:
```
(6745, 6)
```

In [11]:
```
df.dtypes
```

Out[11]:
```
Request id             int64
Pickup point          object
Driver id            float64
Status                object
Request timestamp     object
Drop timestamp        object
dtype: object
```

In [12]:
```
df.describe()
```

Out[12]:

	Request id	Driver id
count	6745.000000	4095.000000
mean	3384.644922	149.501343
std	1955.099667	86.051994
min	1.000000	1.000000
25%	1691.000000	75.000000
50%	3387.000000	149.000000
75%	5080.000000	224.000000
max	6766.000000	300.000000

In [13]:
```
df.isnull()
```

Out[13]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
6740	False	False	True	False	False	True
6741	False	False	True	False	False	True
6742	False	False	True	False	False	True
6743	False	False	True	False	False	True
6744	False	False	True	False	False	True

6745 rows × 6 columns

In [14]:
```
df.notnull()
```

Out[14]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	True	True	True	True	True	True
1	True	True	True	True	True	True
2	True	True	True	True	True	True
3	True	True	True	True	True	True
4	True	True	True	True	True	True
...	...	...	...	...	...	...
6740	True	True	False	True	True	False
6741	True	True	False	True	True	False
6742	True	True	False	True	True	False
6743	True	True	False	True	True	False
6744	True	True	False	True	True	False

6745 rows × 6 columns

In [15]:
```
df.isna()
```

Out[15]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
6740	False	False	True	False	False	True
6741	False	False	True	False	False	True
6742	False	False	True	False	False	True
6743	False	False	True	False	False	True
6744	False	False	True	False	False	True

6745 rows × 6 columns

In [16]:
```
df.notna()
```

Out[16]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	True	True	True	True	True	True
1	True	True	True	True	True	True
2	True	True	True	True	True	True
3	True	True	True	True	True	True
4	True	True	True	True	True	True
...	...	...	...	...	...	...
6740	True	True	False	True	True	False
6741	True	True	False	True	True	False
6742	True	True	False	True	True	False
6743	True	True	False	True	True	False
6744	True	True	False	True	True	False

6745 rows × 6 columns

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
Request id              0
Pickup point            0
Driver id            2650
Status                  0
Request timestamp       0
Drop timestamp       3914
dtype: int64
```

In [18]:
```
df.isnull().any()
```

Out[18]:
```
Request id           False
Pickup point         False
Driver id             True
Status               False
Request timestamp    False
Drop timestamp        True
dtype: bool
```

In [19]:
```
df.iloc[69]
```

Out[19]:
```
Request id                    1769
Pickup point                  City
Driver id                      8.0
Status               Trip Completed
Request timestamp    12/7/2016 8:57
Drop timestamp       12/7/2016 9:24
Name: 69, dtype: object
```

In [20]:
```
df[0:70]
```

Out[20]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	619	Airport	1.0	Trip Completed	11/7/2016 11:51	11/7/2016 13:00
1	867	Airport	1.0	Trip Completed	11/7/2016 17:57	11/7/2016 18:47
2	1807	City	1.0	Trip Completed	12/7/2016 9:17	12/7/2016 9:58
3	2532	Airport	1.0	Trip Completed	12/7/2016 21:08	12/7/2016 22:03
4	3112	City	1.0	Trip Completed	13-07-2016 08:33:16	13-07-2016 09:25:47
...	...	...	...	...	...	...
65	5898	City	7.0	Trip Completed	15-07-2016 09:50:28	15-07-2016 10:40:39
66	6142	Airport	7.0	Trip Completed	15-07-2016 15:50:15	15-07-2016 16:36:56
67	380	Airport	8.0	Trip Completed	11/7/2016 8:18	11/7/2016 9:18
68	1050	Airport	8.0	Trip Completed	11/7/2016 19:39	11/7/2016 20:30
69	1769	City	8.0	Trip Completed	12/7/2016 8:57	12/7/2016 9:24

70 rows × 6 columns

In [21]:
```
df.describe(include = 'all')
```

Out[21]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
count	6745.000000	6745	4095.000000	6745	6745	2831
unique	NaN	2	NaN	3	5618	2598
top	NaN	City	NaN	Trip Completed	11/7/2016 19:02	11/7/2016 13:00
freq	NaN	3507	NaN	2831	6	4
mean	3384.644922	NaN	149.501343	NaN	NaN	NaN
std	1955.099667	NaN	86.051994	NaN	NaN	NaN
min	1.000000	NaN	1.000000	NaN	NaN	NaN
25%	1691.000000	NaN	75.000000	NaN	NaN	NaN
50%	3387.000000	NaN	149.000000	NaN	NaN	NaN
75%	5080.000000	NaN	224.000000	NaN	NaN	NaN
max	6766.000000	NaN	300.000000	NaN	NaN	NaN

In [22]:
```
df.isna().sum()
```

Out[22]:
```
Request id              0
Pickup point            0
Driver id            2650
Status                  0
Request timestamp       0
Drop timestamp       3914
dtype: int64
```

In [23]:
```
df.isnull().sum().sum()
```

Out[23]:
```
6564
```

In [24]:
```
df['Request id']
```

Out[24]:
```
0        619
1        867
2       1807
3       2532
4       3112
       ...
6740    6745
6741    6752
6742    6751
6743    6754
6744    6753
Name: Request id, Length: 6745, dtype: int64
```

In [25]:
```
df.sort_values(by='Request id')
```

Out[25]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
2700	1	Airport	285.0	Trip Completed	11/7/2016 0:20	11/7/2016 0:51
4098	2	Airport	NaN	No Cars Available	11/7/2016 0:23	NaN
776	3	Airport	80.0	Trip Completed	11/7/2016 0:24	11/7/2016 1:31
4101	4	City	NaN	No Cars Available	11/7/2016 0:37	NaN
2506	5	Airport	264.0	Trip Completed	11/7/2016 0:36	11/7/2016 1:35
...	...	...	...	...	...	...
2534	6762	Airport	267.0	Trip Completed	15-07-2016 00:07:29	15-07-2016 00:52:50
2137	6763	City	224.0	Trip Completed	15-07-2016 00:04:44	15-07-2016 01:06:42
2324	6764	City	243.0	Trip Completed	15-07-2016 00:06:12	15-07-2016 01:17:53
6165	6765	Airport	NaN	No Cars Available	15-07-2016 00:09:09	NaN
1042	6766	City	108.0	Trip Completed	15-07-2016 00:06:56	15-07-2016 01:10:34

6745 rows × 6 columns

In [26]:
```
df.sort_values(by='Pickup point')
```

Out[26]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	619	Airport	1.0	Trip Completed	11/7/2016 11:51	11/7/2016 13:00
4481	1126	Airport	NaN	No Cars Available	11/7/2016 20:28	NaN
4482	1120	Airport	NaN	No Cars Available	11/7/2016 20:29	NaN
4483	1122	Airport	NaN	No Cars Available	11/7/2016 20:29	NaN
4485	1127	Airport	NaN	No Cars Available	11/7/2016 20:30	NaN
...	...	...	...	...	...	...
1752	4693	City	184.0	Trip Completed	14-07-2016 13:01:23	14-07-2016 14:10:11
3799	1521	City	230.0	Cancelled	12/7/2016 5:50	NaN
3800	2771	City	230.0	Cancelled	13-07-2016 04:24:36	NaN
3767	3185	City	223.0	Cancelled	13-07-2016 09:24:46	NaN
3372	1738	City	132.0	Cancelled	12/7/2016 8:26	NaN

6745 rows × 6 columns

In [27]:
```
df['Request id'].isnull().sum()
```

Out[27]:
```
0
```

In [28]:
```
df['Request id'] = df['Request id'].astype('float64')
df.dtypes
```

Out[28]:
```
Request id           float64
Pickup point          object
Driver id            float64
Status                object
Request timestamp     object
Drop timestamp        object
dtype: object
```

Importing Iris Dataset

In [29]:
```
df = pd.read_csv('Datasets/Iris.csv')
```

In [30]:
```
df
```

Out[30]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [31]:
```
df.head(10)
```

Out[31]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

In [32]:
```
df.tail(10)
```

Out[32]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
140	141	6.7	3.1	5.6	2.4	Iris-virginica
141	142	6.9	3.1	5.1	2.3	Iris-virginica
142	143	5.8	2.7	5.1	1.9	Iris-virginica
143	144	6.8	3.2	5.9	2.3	Iris-virginica
144	145	6.7	3.3	5.7	2.5	Iris-virginica
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [33]:
```
df.index
```

Out[33]:
```
RangeIndex(start=0, stop=150, step=1)
```

In [34]:
```
df.columns
```

Out[34]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

In [35]:
```
df.columns.values
```

Out[35]:
```
array(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm',
       'PetalWidthCm', 'Species'], dtype=object)
```

In [36]:
```
df.shape
```

Out[36]:
```
(150, 6)
```

In [37]:
```
df.dtypes
```

Out[37]:
```
Id                int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [38]:
```
df.describe()
```

Out[38]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

Normalising the data

In [39]:
```
min_max_scaler = preprocessing.MinMaxScaler()
```

In [40]:
```
x = df.iloc[:,:4]
```

In [41]:
```
x_scaled = min_max_scaler.fit_transform(x)
```

In [42]:
```
df_normalised = pd.DataFrame(x_scaled)
```

In [43]:
```
df_normalised
```

Out[43]:

	0	1	2	3
0	0.000000	0.222222	0.625000	0.067797
1	0.006711	0.166667	0.416667	0.067797
2	0.013423	0.111111	0.500000	0.050847
3	0.020134	0.083333	0.458333	0.084746
4	0.026846	0.194444	0.666667	0.067797
...	...	...	...	...
145	0.973154	0.666667	0.416667	0.711864
146	0.979866	0.555556	0.208333	0.677966
147	0.986577	0.611111	0.416667	0.711864
148	0.993289	0.527778	0.583333	0.745763
149	1.000000	0.444444	0.416667	0.694915

150 rows × 4 columns

In [44]:
```
df['Species'].unique()
```

Out[44]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [45]:
```
features_df = df.drop(columns=['Species'])
features_df
```

Out[45]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2
...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5.0	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

150 rows × 5 columns

Encoding the Species column

In [46]:
```
enc = preprocessing.OneHotEncoder()
enc_df = (enc.fit_transform(df[['Species']]))
x = pd.DataFrame(enc_df)
x
```

Out[46]:

	0
0	(0, 0)\t1.0
1	(0, 0)\t1.0
2	(0, 0)\t1.0
3	(0, 0)\t1.0
4	(0, 0)\t1.0
...	...
145	(0, 2)\t1.0
146	(0, 2)\t1.0
147	(0, 2)\t1.0
148	(0, 2)\t1.0
149	(0, 2)\t1.0

150 rows × 1 columns

In [47]:
```
df_encode=features_df.join(x)
```

In [48]:
```
df_encode
```

Out[48]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	0
0	1	5.1	3.5	1.4	0.2	(0, 0)\t1.0
1	2	4.9	3.0	1.4	0.2	(0, 0)\t1.0
2	3	4.7	3.2	1.3	0.2	(0, 0)\t1.0
3	4	4.6	3.1	1.5	0.2	(0, 0)\t1.0
4	5	5.0	3.6	1.4	0.2	(0, 0)\t1.0
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	(0, 2)\t1.0
146	147	6.3	2.5	5.0	1.9	(0, 2)\t1.0
147	148	6.5	3.0	5.2	2.0	(0, 2)\t1.0
148	149	6.2	3.4	5.4	2.3	(0, 2)\t1.0
149	150	5.9	3.0	5.1	1.8	(0, 2)\t1.0

150 rows × 6 columns

In [49]:
```
df_encode.rename(columns={0:'Sentosa',1:'Versicolor',2:'Verginica'},inplace=True
```

In [50]:
```
df_encode
```

Out[50]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Sentosa
0	1	5.1	3.5	1.4	0.2	(0, 0)\t1.0
1	2	4.9	3.0	1.4	0.2	(0, 0)\t1.0
2	3	4.7	3.2	1.3	0.2	(0, 0)\t1.0
3	4	4.6	3.1	1.5	0.2	(0, 0)\t1.0
4	5	5.0	3.6	1.4	0.2	(0, 0)\t1.0
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	(0, 2)\t1.0
146	147	6.3	2.5	5.0	1.9	(0, 2)\t1.0
147	148	6.5	3.0	5.2	2.0	(0, 2)\t1.0
148	149	6.2	3.4	5.4	2.3	(0, 2)\t1.0
149	150	5.9	3.0	5.1	1.8	(0, 2)\t1.0

150 rows × 6 columns