



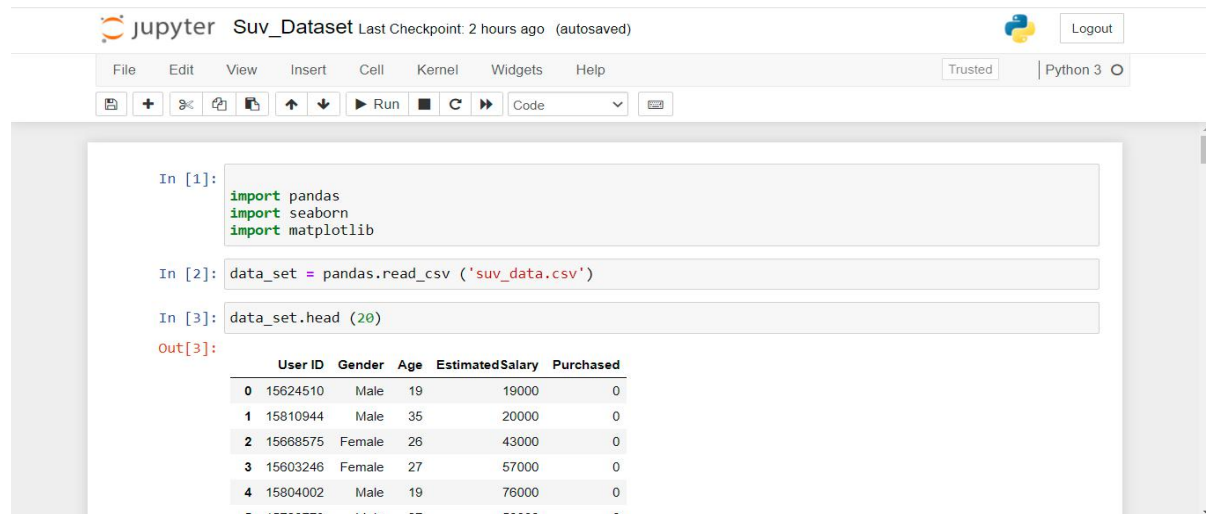
Data Science Lab - 5

Suv dataset

Name: Manikandan P
RegNo: 2019202030

WORKING WITH SUV - DATA SET

Importing Libraries



The screenshot shows a Jupyter Notebook titled 'Suv_Dataset' with a last checkpoint 2 hours ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code cells show the following:

```
In [1]: import pandas
import seaborn
import matplotlib

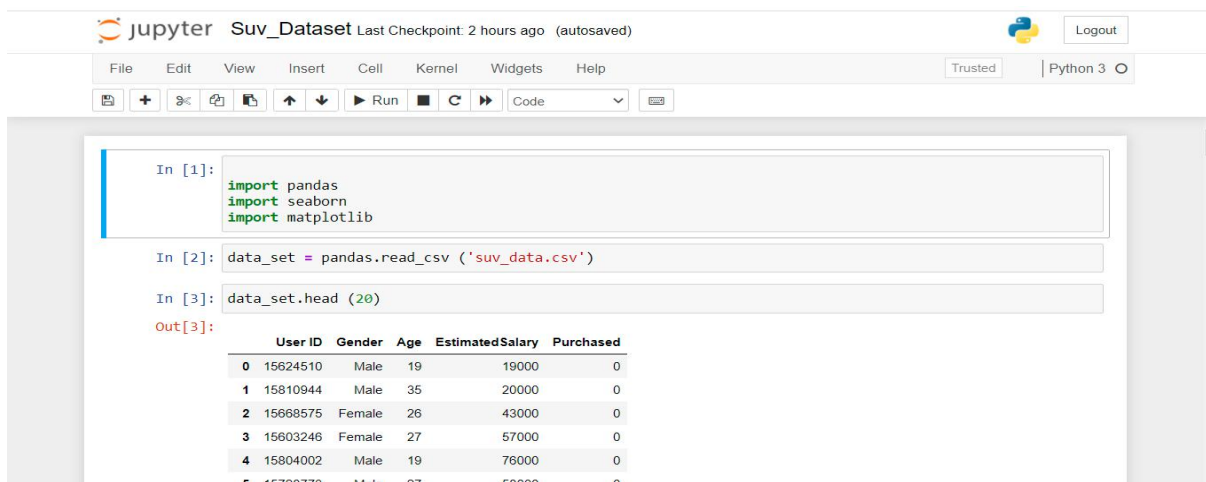
In [2]: data_set = pandas.read_csv ('suv_data.csv')

In [3]: data_set.head (20)
```

The output of the third cell is a table showing the first 20 rows of the dataset:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15726773	Male	27	50000	0

Loading suv_data.csv data set



The screenshot shows a Jupyter Notebook titled 'Suv_Dataset' with a last checkpoint 2 hours ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code cells show the following:

```
In [1]: import pandas
import seaborn
import matplotlib

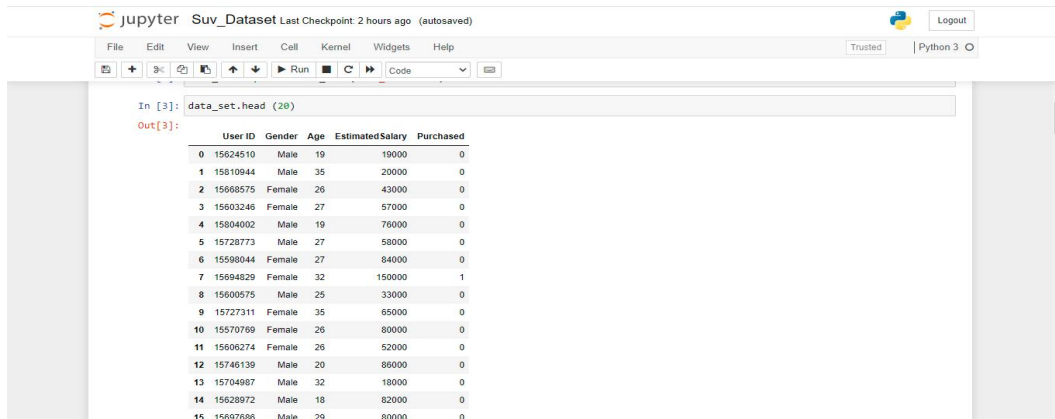
In [2]: data_set = pandas.read_csv ('suv_data.csv')

In [3]: data_set.head (20)
```

The output of the third cell is a table showing the first 20 rows of the dataset:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15726773	Male	27	50000	0

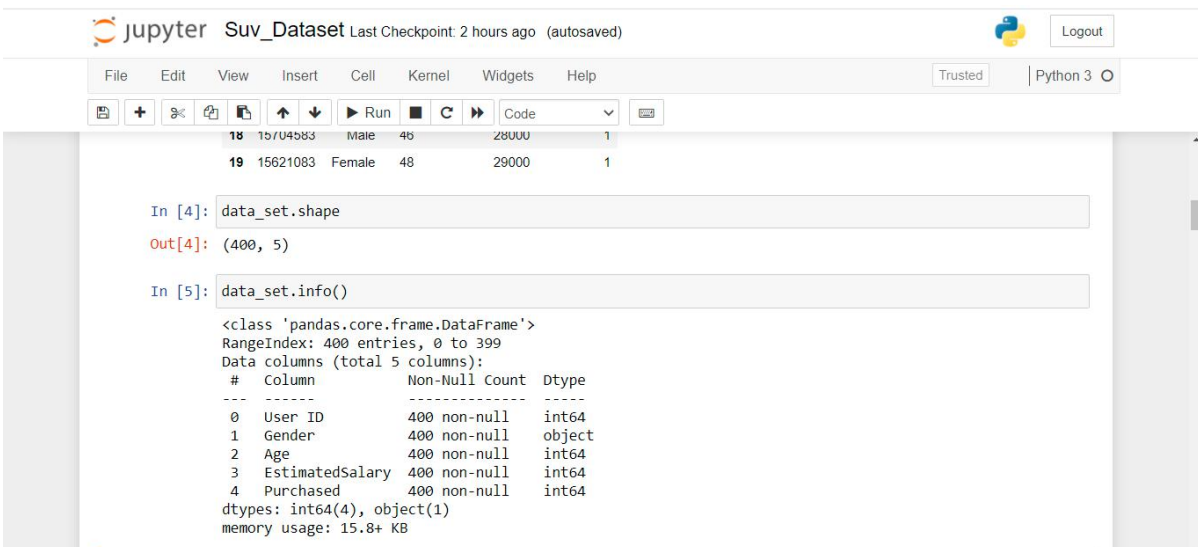
Displaying first 20 records of data



The screenshot shows a Jupyter Notebook interface with the title "Suv_Dataset". The code cell contains the command `data_set.head(20)`. The output displays a table with 20 rows and 5 columns: User ID, Gender, Age, EstimatedSalary, and Purchased. The data is as follows:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15610944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0
10	15570769	Female	26	80000	0
11	15696274	Female	26	52000	0
12	15746139	Male	20	86000	0
13	15704987	Male	32	18000	0
14	15628972	Male	18	82000	0
15	15607686	Male	20	80000	0

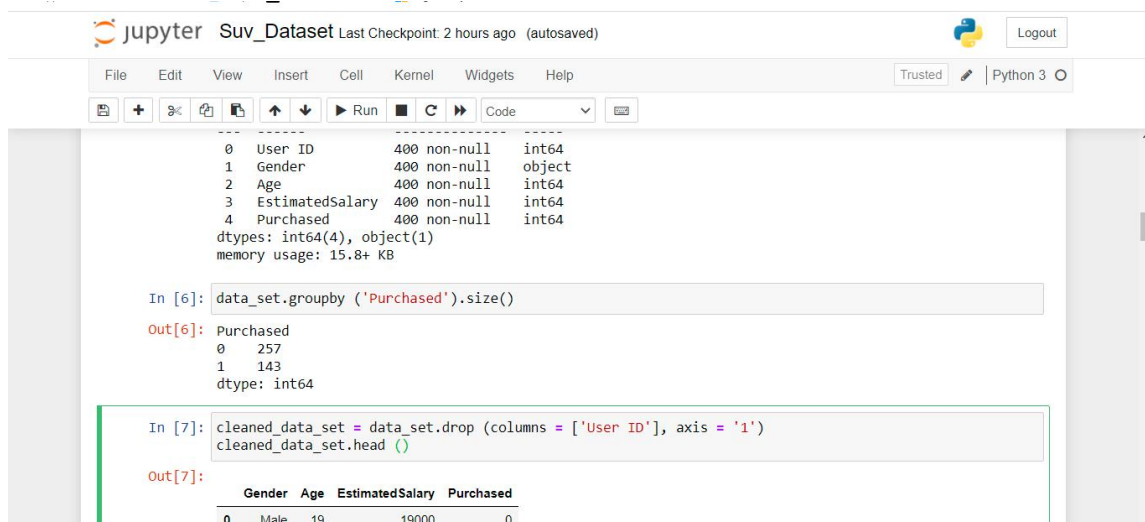
Displaying datatypes of columns



The screenshot shows a Jupyter Notebook interface with the title "Suv_Dataset". The code cell contains the command `data_set.info()`. The output displays the shape of the dataset and the data types of the columns. The shape is (400, 5). The data types are: User ID (int64), Gender (object), Age (int64), EstimatedSalary (int64), and Purchased (int64). The output also shows the non-null count for each column, which is 400 for all columns.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   User ID               400 non-null   int64
1   Gender                400 non-null   object
2   Age                   400 non-null   int64
3   EstimatedSalary       400 non-null   int64
4   Purchased             400 non-null   int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

Grouping data set by “purchased”



The Jupyter Notebook interface shows the 'Suv_Dataset' with a last checkpoint 2 hours ago. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for file operations, zooming, and running code. The code cell contains the following:

```
0 User ID      400 non-null  int64
1 Gender      400 non-null  object
2 Age         400 non-null  int64
3 EstimatedSalary 400 non-null  int64
4 Purchased   400 non-null  int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

In [6]: `data_set.groupby('Purchased').size()`

Out[6]:

Purchased	size
0	257
1	143

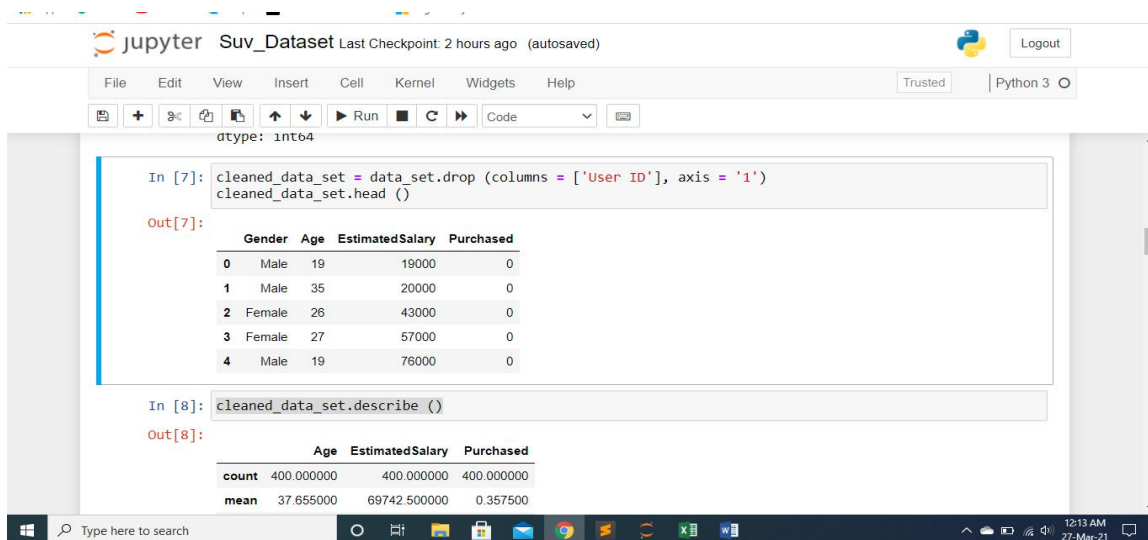
dtype: int64

In [7]: `cleaned_data_set = data_set.drop(columns=['User ID'], axis='1')`
`cleaned_data_set.head()`

Out[7]:

	Gender	Age	EstimatedSalary	Purchased
0	Male	19	19000	0

Dropping column “User ID” and displaying



The Jupyter Notebook interface shows the 'Suv_Dataset' with a last checkpoint 2 hours ago. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for file operations, zooming, and running code. The code cell contains the following:

```
dtype: int64
```

In [7]: `cleaned_data_set = data_set.drop(columns=['User ID'], axis='1')`
`cleaned_data_set.head()`

Out[7]:

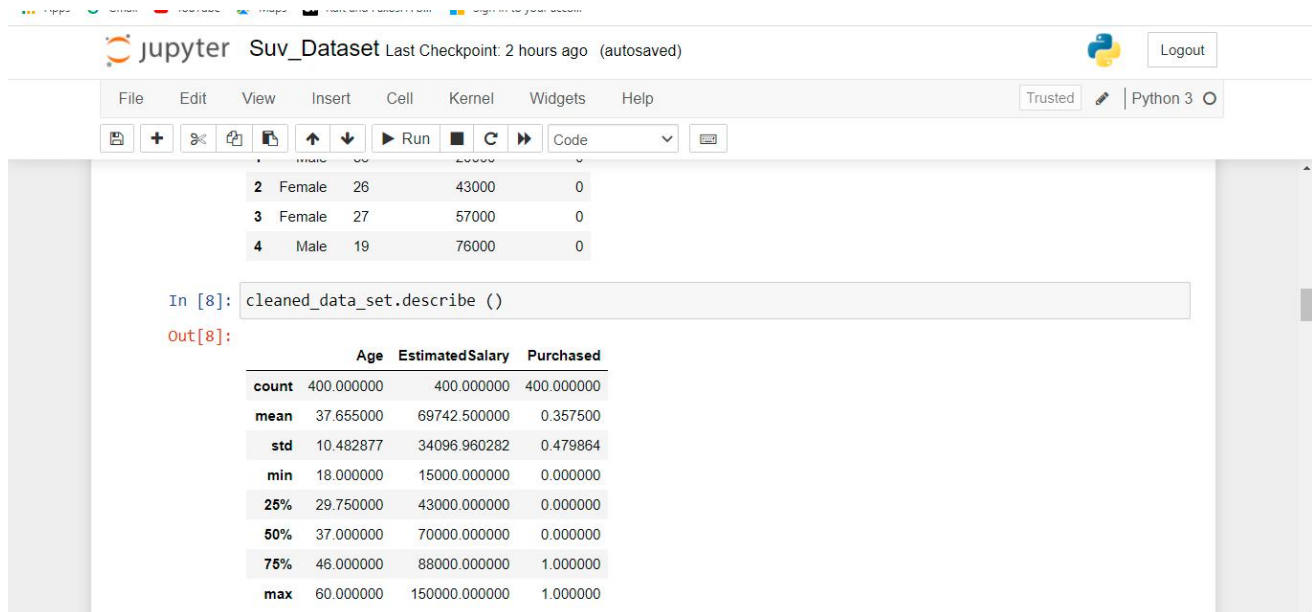
	Gender	Age	EstimatedSalary	Purchased
0	Male	19	19000	0
1	Male	35	20000	0
2	Female	26	43000	0
3	Female	27	57000	0
4	Male	19	76000	0

In [8]: `cleaned_data_set.describe()`

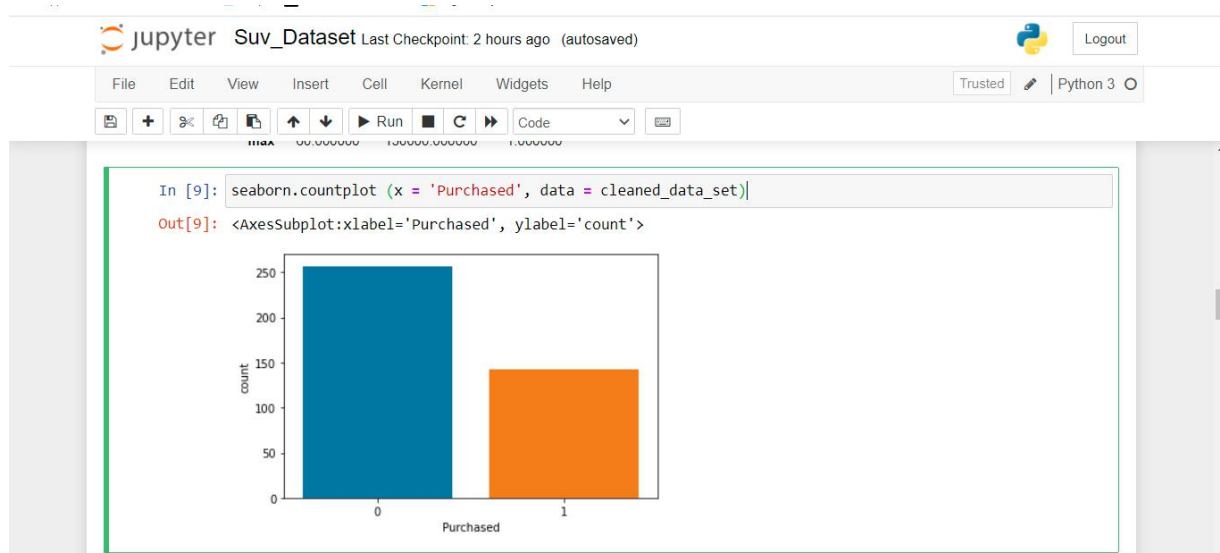
Out[8]:

	Age	EstimatedSalary	Purchased
count	400.000000	400.000000	400.000000
mean	37.655000	69742.500000	0.357500

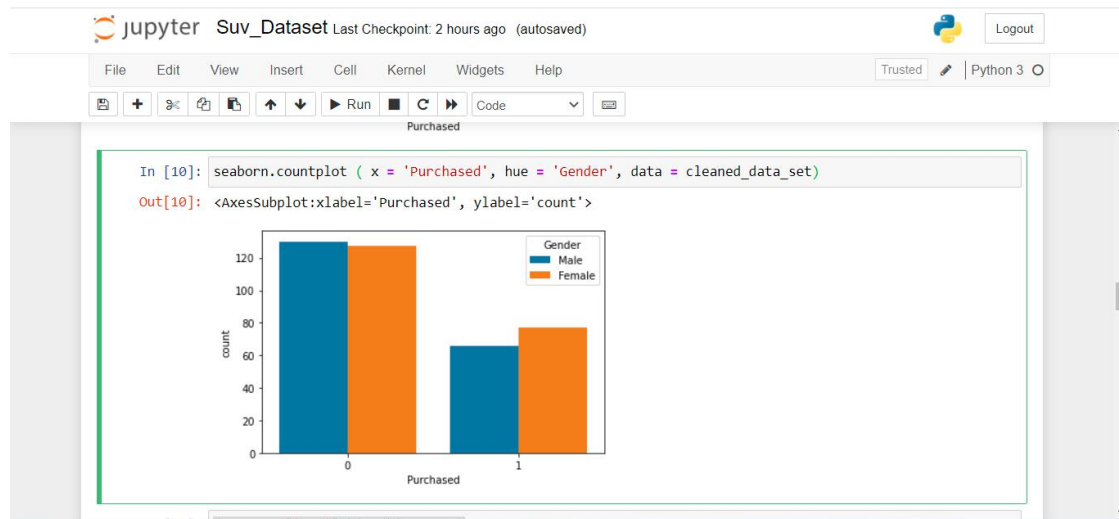
Describing Data set



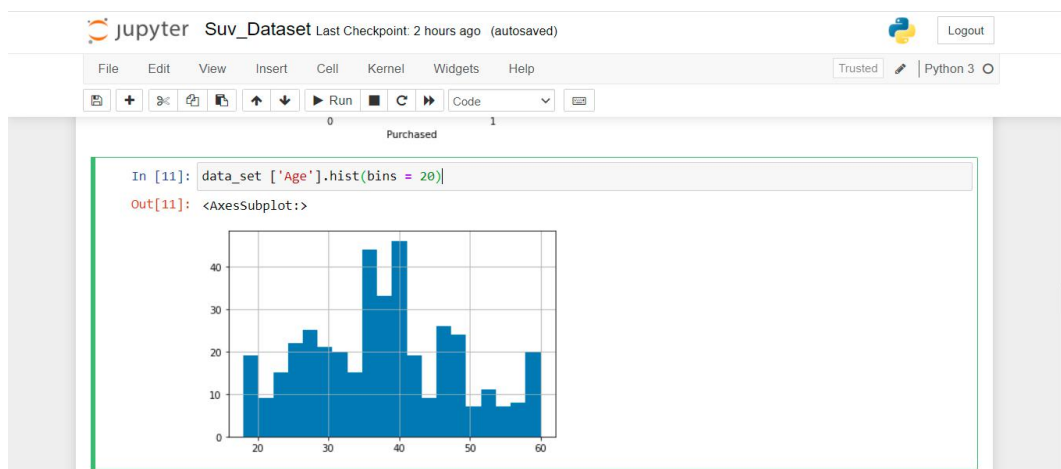
Displaying purchased and not purchased count in Bar Diagram



Displaying purchased SUV by categorizing in gender



Data set of column “Age” in bins



Categorizing by age of SUV-Purchasers

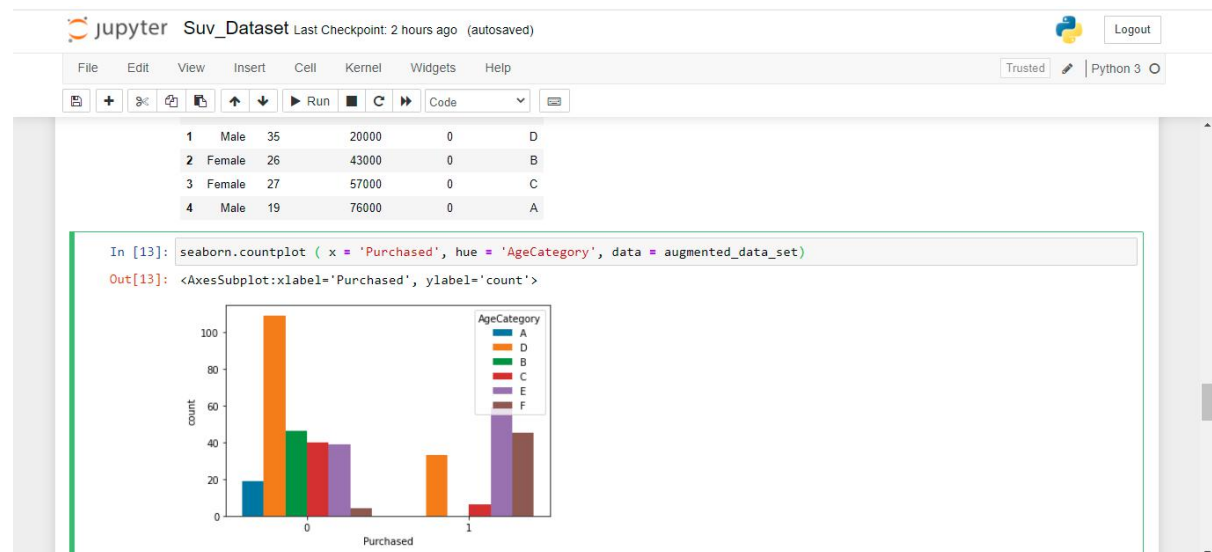
```
Jupyter Suv_Dataset Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [12]: age_category = []
for i in range(0, len(data_set['Age'])):
    if cleaned_data_set['Age'][i] <= 20:
        age_category.append('A');
    elif 20 < cleaned_data_set['Age'][i] <= 26:
        age_category.append('B');
    elif 26 < cleaned_data_set['Age'][i] <= 30:
        age_category.append('C');
    elif 30 < cleaned_data_set['Age'][i] <= 40:
        age_category.append('D');
    elif 40 < cleaned_data_set['Age'][i] <= 50:
        age_category.append('E');
    else:
        age_category.append('F');

age_data_frame = pandas.DataFrame(data = age_category, columns = ['AgeCategory'])
augmented_data_set = pandas.concat([cleaned_data_set, age_data_frame], axis = 1)
augmented_data_set.head()


Out[12]:
```

	Gender	Age	EstimatedSalary	Purchased	AgeCategory
0	Male	19	19000	0	A
1	Male	35	20000	0	D
2	Female	26	43000	0	B
3	Female	27	57000	0	C
4	Male	19	76000	0	A

Displaying that data in Bar Chart



Categorizing By Income of the SUV - Buyers

jupyter Suv_Dataset Last Checkpoint: 3 hours ago (autosaved)  Logout


File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

0 0 1 Purchased

```
In [14]: income_category = []
for i in range(0, len (data_set ['EstimatedSalary'])):
    if cleaned_data_set ['EstimatedSalary'][i] <= 19500:
        income_category.append ('Very Low');
    elif 19500 < cleaned_data_set ['EstimatedSalary'][i] <= 40000:
        income_category.append ('Low');
    elif 40000 < cleaned_data_set ['EstimatedSalary'][i] <= 60000:
        income_category.append ('Moderately Low');
    elif 60000 < cleaned_data_set ['EstimatedSalary'][i] <= 80000:
        income_category.append ('Medium');
    elif 80000 < cleaned_data_set ['EstimatedSalary'][i] <= 100000:
        income_category.append ('Moderately high');
    elif 100000 < cleaned_data_set ['EstimatedSalary'][i] <= 130000:
        income_category.append ('Very High');
    elif 130000 < cleaned_data_set ['EstimatedSalary'][i] <= 145000:
        income_category.append ('Very High');
    else:
        income_category.append ('Extremely High');

income_data_frame = pandas.DataFrame (data = income_category, columns = ['IncomeCategory'])
augmented_data_set_2 = pandas.concat([augmented_data_set, income_data_frame], axis = 1)
augmented_data_set_2.head()
```

Out[14]:

jupyter Suv_Dataset Last Checkpoint: 3 hours ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
elif 100000 < cleaned_data_set ['EstimatedSalary'][i] <= 130000:
    income_category.append ('Very High');
elif 130000 < cleaned_data_set ['EstimatedSalary'][i] <= 145000:
    income_category.append ('Very High');
else:
    income_category.append ('Extremely High');


income_data_frame = pandas.DataFrame (data = income_category, columns = ['IncomeCategory'])
augmented_data_set_2 = pandas.concat([augmented_data_set, income_data_frame], axis = 1)
augmented_data_set_2.head()
```

Out[14]:

	Gender	Age	EstimatedSalary	Purchased	AgeCategory	IncomeCategory
0	Male	19	19000	0	A	Very Low
1	Male	35	20000	0	D	Low
2	Female	26	43000	0	B	Moderately Low
3	Female	27	57000	0	C	Moderately Low
4	Male	19	76000	0	A	Medium

```
In [15]: seaborn.countplot ( x = 'Purchased', hue = 'IncomeCategory', data = augmented_data_set_2)
```

Out[15]: <AxesSubplot:xlabel='Purchased', ylabel='count'>



Displaying that data in Bar Chart

