

EPLORATORY DATA ANALYSIS ON GERMAN

CREDIT DATASET

Importing Relevant Labraries

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

IMPORTING AND INSPECTING OUR DATA

SET

```
In [4]: df=pd.read_csv("C:/Users/Sanayak/Desktop/german_credit_data.csv")
df.head()
```

```
Out[4]:
```

| | Unnamed: 0 | Age | Sex | Job | Housing | Saving accounts | Checking account | Credit amount | Duration | |
|---|------------|-----|--------|-----|---------|-----------------|------------------|---------------|----------|----------|
| 0 | 0 | 67 | male | 2 | own | NaN | little | 1169 | 6 | |
| 1 | 1 | 22 | female | 2 | own | little | moderate | 5951 | 48 | |
| 2 | 2 | 49 | male | 1 | own | little | NaN | 2096 | 12 | |
| 3 | 3 | 45 | male | 2 | free | little | little | 7882 | 42 | furnitur |
| 4 | 4 | 53 | male | 2 | free | little | little | 4870 | 24 | |

```
In [5]: df.tail()
```

```
Out[5]:
```

| | Unnamed: 0 | Age | Sex | Job | Housing | Saving accounts | Checking account | Credit amount | Duration | |
|-----|------------|-----|--------|-----|---------|-----------------|------------------|---------------|----------|-------|
| 995 | 995 | 31 | female | 1 | own | little | NaN | 1736 | 12 | furni |
| 996 | 996 | 40 | male | 3 | own | little | little | 3857 | 30 | |
| 997 | 997 | 38 | male | 2 | own | little | NaN | 804 | 12 | |
| 998 | 998 | 23 | male | 2 | free | little | little | 1845 | 45 | |
| 999 | 999 | 27 | male | 2 | own | moderate | moderate | 4576 | 45 | |

```
In [7]: df.count()
```

```
Out[7]: Unnamed: 0      1000
        Age           1000
        Sex           1000
        Job           1000
        Housing       1000
        Saving accounts 817
        Checking account 606
        Credit amount  1000
        Duration       1000
        Purpose       1000
        dtype: int64
```

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

```
Out[8]: Unnamed: 0      1000
        Age           53
        Sex           2
        Job           4
        Housing       3
        Saving accounts 4
        Checking account 3
        Credit amount  921
        Duration       33
        Purpose        8
        dtype: int64
```

```
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Unnamed: 0            1000 non-null  int64
 1   Age                   1000 non-null  int64
 2   Sex                   1000 non-null  object
 3   Job                   1000 non-null  int64
 4   Housing               1000 non-null  object
 5   Saving accounts       817 non-null   object
 6   Checking account      606 non-null   object
 7   Credit amount         1000 non-null  int64
 8   Duration              1000 non-null  int64
 9   Purpose               1000 non-null  object
dtypes: int64(5), object(5)
memory usage: 78.3+ KB
```

```
In [10]: df.describe()
```

Out[10]:

| | Unnamed: 0 | Age | Job | Credit amount | Duration |
|--------------|-------------|-------------|-------------|---------------|-------------|
| count | 1000.000000 | 1000.000000 | 1000.000000 | 1000.000000 | 1000.000000 |
| mean | 499.500000 | 35.546000 | 1.904000 | 3271.258000 | 20.903000 |
| std | 288.819436 | 11.375469 | 0.653614 | 2822.736876 | 12.058814 |
| min | 0.000000 | 19.000000 | 0.000000 | 250.000000 | 4.000000 |
| 25% | 249.750000 | 27.000000 | 2.000000 | 1365.500000 | 12.000000 |
| 50% | 499.500000 | 33.000000 | 2.000000 | 2319.500000 | 18.000000 |
| 75% | 749.250000 | 42.000000 | 2.000000 | 3972.250000 | 24.000000 |
| max | 999.000000 | 75.000000 | 3.000000 | 18424.000000 | 72.000000 |

In [12]: `df.shape`

Out[12]: (1000, 10)

DATA CLEANSING AND DATA

MANIPULATION

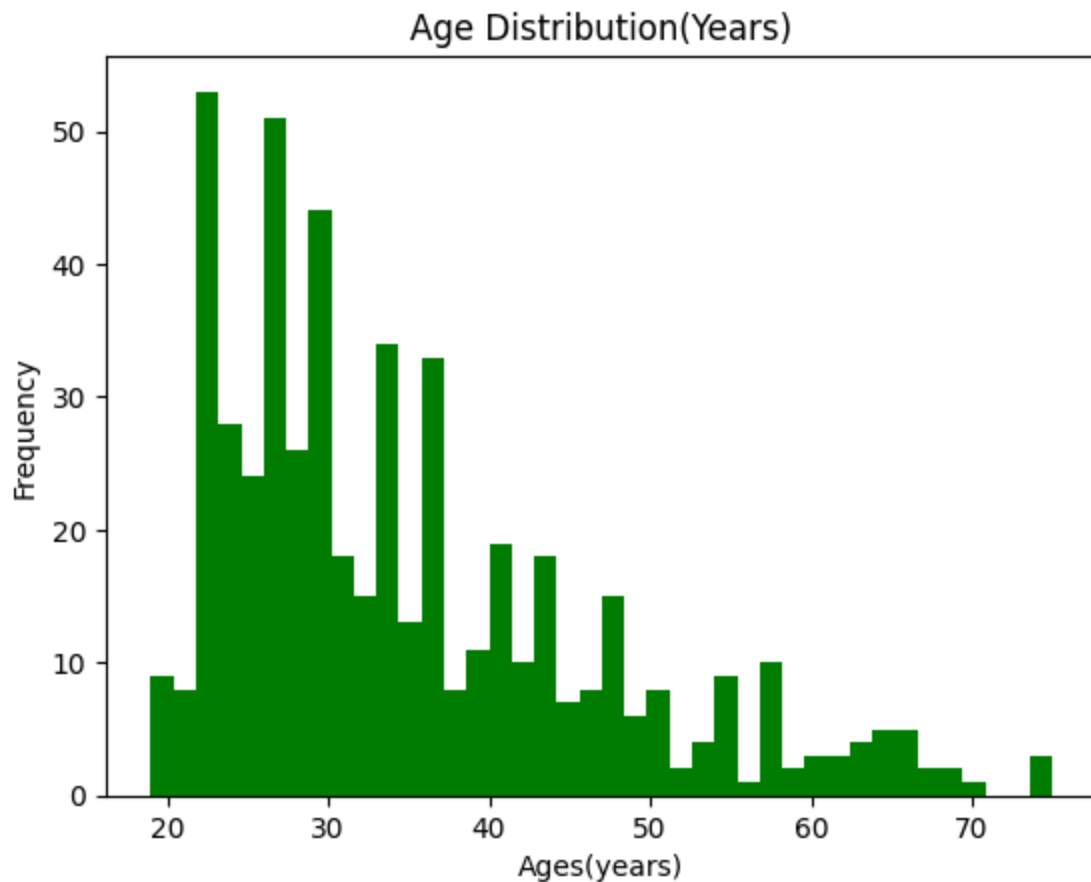
Our Data set has missing values(NaN), and it is best we drop the rows with missing string values

In [17]: `df.dropna(axis=0,how="any",inplace=True)`
`df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 522 entries, 1 to 999
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0            522 non-null   int64
1   Age                   522 non-null   int64
2   Sex                   522 non-null   object
3   Job                   522 non-null   int64
4   Housing               522 non-null   object
5   Saving accounts       522 non-null   object
6   Checking account      522 non-null   object
7   Credit amount         522 non-null   int64
8   Duration              522 non-null   int64
9   Purpose               522 non-null   object
dtypes: int64(5), object(5)
memory usage: 44.9+ KB
```

VISUALISATION OF OUR DATA SET USING RELEVANT GRAPHS IN
PYTHON LIBRARIES

```
In [18]: plt.hist(df.Age,color="green",bins=40)
plt.xlabel("Ages(years)")
plt.ylabel("Frequency")
plt.title("Age Distribution(Years)")
plt.style.use("fivethirtyeight")
plt.show()
```



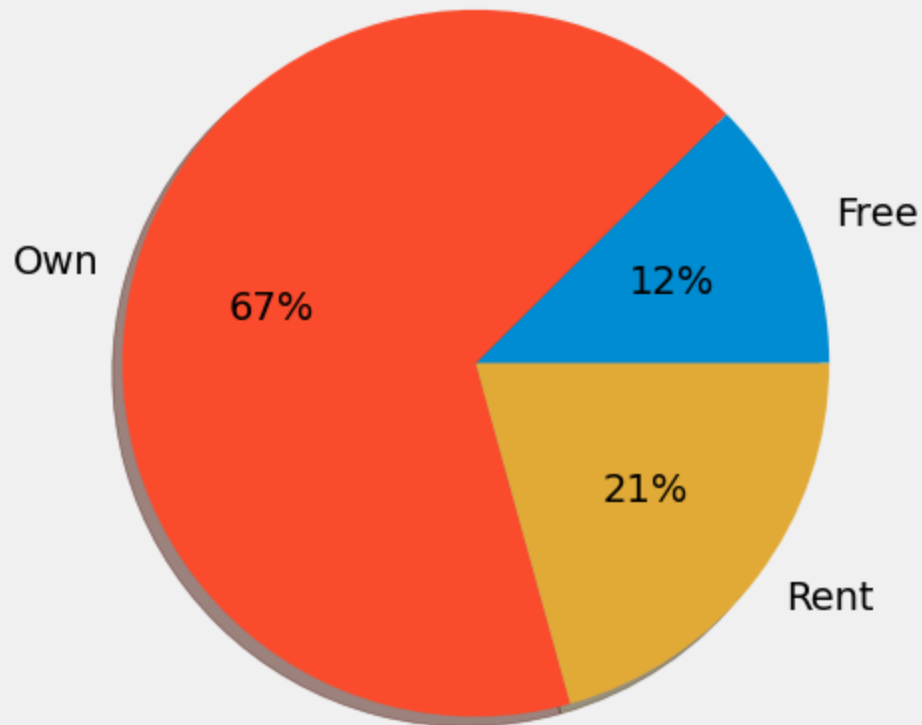
```
In [19]: house_type=pd.Categorical(df["Housing"])
house_type.describe()
```

```
Out[19]:
```

| | counts | fregs |
|-------------------|--------|----------|
| categories | | |
| free | 65 | 0.124521 |
| own | 349 | 0.668582 |
| rent | 108 | 0.206897 |

```
In [20]: house_category=[65,349,108]
types=["Free","Own","Rent"]
plt.pie(house_category,labels=types,shadow=True,autopct="%1.f%%")
plt.title("Sectorial Representation of House Type")
plt.axis("equal")
plt.show()
```

Sectorial Representation of House Type

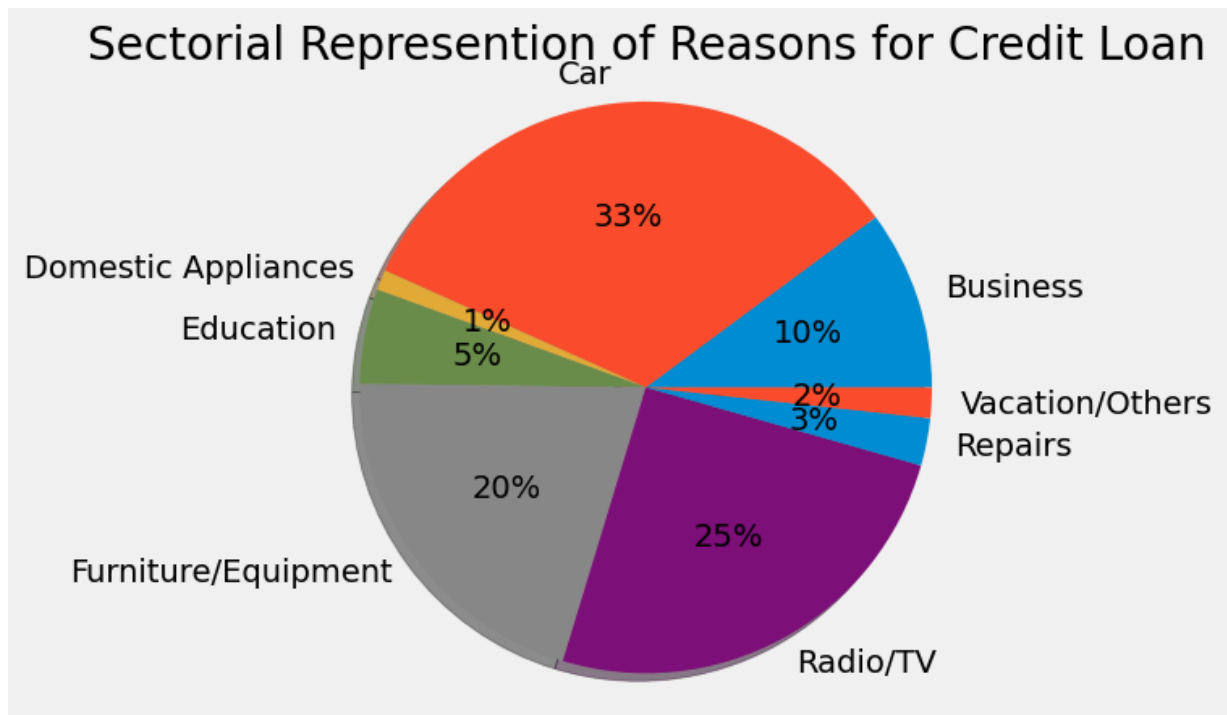


```
In [21]: purpose=pd.Categorical(df["Purpose"])
         purpose.describe()
```

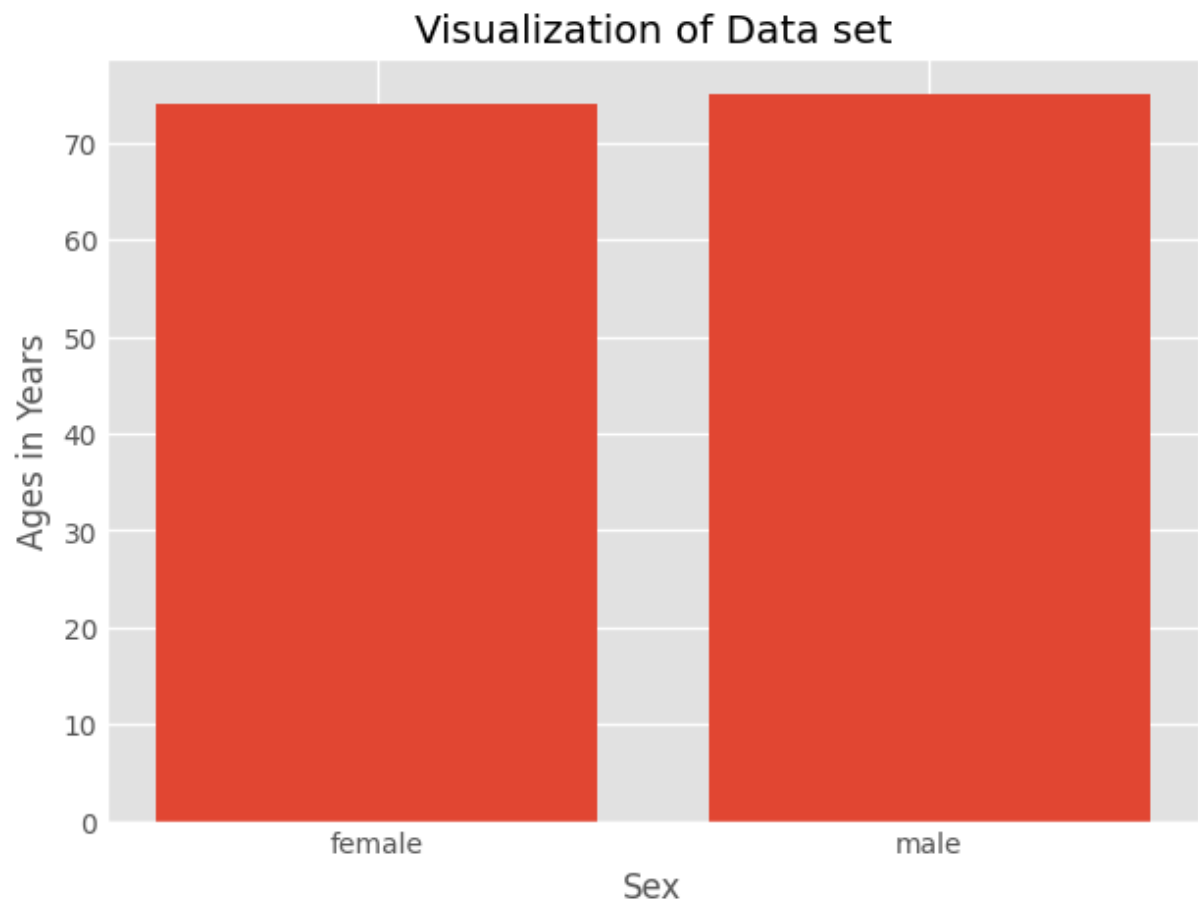
```
Out[21]:
```

| | counts | freqs |
|---------------------|--------|----------|
| business | 53 | 0.101533 |
| car | 173 | 0.331418 |
| domestic appliances | 6 | 0.011494 |
| education | 28 | 0.053640 |
| furniture/equipment | 107 | 0.204981 |
| radio/TV | 132 | 0.252874 |
| repairs | 14 | 0.026820 |
| vacation/others | 9 | 0.017241 |

```
In [24]: purpose=[53,173,6,28,107,132,14,9]
         types=["Business","Car","Domestic Appliances","Education","Furniture/Equipment","R
         plt.pie(purpose,labels=types,shadow=True,autopct="%1.f%%")
         plt.title("Sectorial Representation of Reasons for Credit Loan")
         plt.axis("equal")
         plt.show()
```



```
In [40]: plt.bar(df.Sex, df.Age)
plt.xlabel("Sex")
plt.ylabel("Ages in Years")
plt.title("Visualization of Data set")
plt.style.use("ggplot")
plt.show()
```



PREPARED BY VICTOR INIOBONG ECONOMIST | DATA SCIENTIST | MATHEMATICS TUTOR