Welcome to my JUPYTER

What is Outlier ?

An outlier is a data point that is noticeably different from the rest. They represent errors in measurement, bad data collection, or simply show variables not considered when collecting the data.

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

In [2]:
```
os.chdir('D:\machine learning\Raw data')
```

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

Out[4]:

	Unnamed: 0	Age	Sex	Job	Housing	Saving accounts	Checking_account	Credit_amount	Duration	Purpose	Risk
0	0	67	male	2	own	NaN	little	1169	6	radio/TV	good
1	1	22	female	2	own	little	moderate	5951	48	radio/TV	bad
2	2	49	male	1	own	little	NaN	2096	12	education	good
3	3	45	male	2	free	little	little	7882	42	furniture/equipment	good
4	4	53	male	2	free	little	little	4870	24	car	bad

Discussion Related With Outliers And Impact On Machine Learning!!

Which Machine LEarning Models Are Sensitive To Outliers?

Naivye Bayes Classifier--- Not Sensitive To Outliers

SVM-------- Not Sensitive To Outliers

Linear Regression---------- Sensitive To Outliers

Logistic Regression------- Sensitive To Outliers

Decision Tree Regressor or Classifier---- Not Sensitive

Ensemble(RF,XGboost,GB)------- Not Sensitive

KNN------------------------ Not Sensitive

Kmeans----------------------- Sensitive

Hierarichal------------------ Sensitive

PCA------------------------- Sensitive

Neural Networks-------------- Sensitive

In [5]:
```
df_credit.shape
```

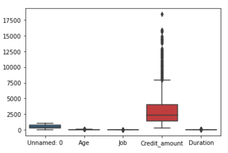
Out[5]:
```
(1000, 11)
```

In [6]:
```
df_credit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 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
 10  Risk              1000 non-null   object
dtypes: int64(5), object(6)
memory usage: 86.1+ KB
```

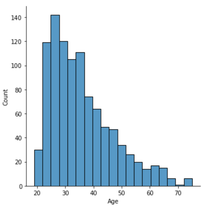
In [7]:
```
sns.boxplot(data=df_credit)
```

Out[7]:
```
<AxesSubplot:>
```



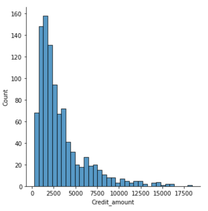
In [8]:
```
sns.displot(df_credit['Age'])
```

Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x1f53d8c31c0>
```



In [9]:
```
sns.displot(df_credit['Credit_amount'])
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x1f538ad7700>
```

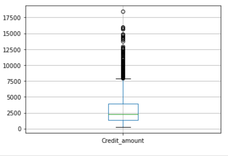


In [10]:
```
df_credit['Credit_amount'].describe()
```

Out[10]:
```
count     1000.000000
mean      3271.258000
std       2822.736876
min        250.000000
25%       1365.500000
50%       2319.500000
75%       3972.250000
max      18424.000000
Name: Credit_amount, dtype: float64
```

In [11]:
```
df_credit.boxplot(column="Credit_amount")
```

Out[11]:
```
<AxesSubplot:>
```



In [12]:
```
IQR=df_credit.Credit_amount.quantile(0.75)-df_credit.Credit_amount.quantile(0.25)
```

In [13]:
```
lower_bridge=df_credit['Credit_amount'].quantile(0.25)-(IQR*1.5)
upper_bridge=df_credit['Credit_amount'].quantile(0.75)+(IQR*1.5)
print(lower_bridge), print(upper_bridge)
```

```
-2544.625
7882.375
```

Out[13]:
```
(None, None)
```

In [14]:
```
lower_bridge=df_credit['Credit_amount'].quantile(0.25)-(IQR*3)
upper_bridge=df_credit['Credit_amount'].quantile(0.75)+(IQR*3)
print(lower_bridge), print(upper_bridge)
```

```
-6454.75
11792.5
```

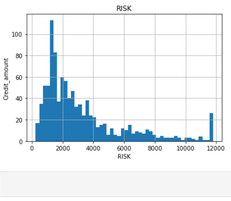
Out[14]:
```
(None, None)
```

In [15]:
```
data=df_credit.copy()
```

In [16]:
```
data.loc[data['Credit_amount']>=11792.5,'Credit_amount']=11792.5
```

In [17]:
```
figure=data.Credit_amount.hist(bins=50)
figure.set_title('RISK')
figure.set_xlabel('RISK')
figure.set_ylabel('Credit_amount')
```

Out[17]:
```
Text(0, 0.5, 'Credit_amount')
```



In [ ]:
```
```