```
# Importing Libraries
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")


# Importing Dataset
df_train=pd.read_csv("/content/train.csv")
df_test=pd.read_csv("/content/test.csv")
df_train.head()
```

	Patient Id	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Patient First Name	Family Name	Fathe r
0	PID0x6418	2.0	Yes	No	Yes	No	4.760603	Richard	NaN	L
1	PID0x25d5	4.0	Yes	Yes	No	No	4.910669	Mike	NaN	Bry
2	PID0x4a82	6.0	Yes	No	No	No	4.893297	Kimberly	NaN	Nas
3	PID0x4ac8	12.0	Yes	No	Yes	No	4.705280	Jeffery	Hoelscher	Aay
4	PID0x1bf7	11.0	Yes	No	NaN	Yes	4.720703	Johanna	Stutzman	Su

5 rows × 45 columns

```
df_train.tail()
```

	Patient Id	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Patient First Name	Family Name	Fa
22078	PID0x5598	4.0	Yes	Yes	Yes	No	5.258298	Lynn	NaN	Alh
22079	PID0x19cb	8.0	No	Yes	No	Yes	4.974220	Matthew	Farley	Da
22080	PID0x3c4f	8.0	Yes	No	Yes	No	5.186470	John	NaN	
22081	PID0x13a	7.0	Yes	No	Yes	Yes	4.858543	Sharon	NaN	

```
# Total Columns
df_train.columns
```

```
Index(['Patient Id', 'Patient Age', 'Genes in mother's side',
       'Inherited from father', 'Maternal gene', 'Paternal gene',
       'Blood cell count (mcL)', 'Patient First Name', 'Family Name',
       'Father's name', 'Mother's age', 'Father's age', 'Institute Name',
       'Location of Institute', 'Status', 'Respiratory Rate (breaths/min)',
       'Heart Rate (rates/min', 'Test 1', 'Test 2', 'Test 3', 'Test 4',
       'Test 5', 'Parental consent', 'Follow-up', 'Gender', 'Birth asphyxia',
       'Autopsy shows birth defect (if applicable)', 'Place of birth',
       'Folic acid details (peri-conceptional)',
       'H/O serious maternal illness', 'H/O radiation exposure (x-ray)',
       'H/O substance abuse', 'Assisted conception IVF/ART',
       'History of anomalies in previous pregnancies',
       'No. of previous abortion', 'Birth defects',
       'White Blood cell count (thousand per microliter)', 'Blood test result',
       'Symptom 1', 'Symptom 2', 'Symptom 3', 'Symptom 4', 'Symptom 5',
       'Genetic Disorder', 'Disorder Subclass'],
      dtype='object')
```

```
# Dropping unwanted columns
df_train.drop("Patient Id",axis=1,inplace=True)
df_train.drop("Family Name",axis=1,inplace=True)
df_train.drop("Patient First Name",axis=1,inplace=True)
df_train.drop("Father's name",axis=1,inplace=True)
df_train.drop("Institute Name",axis=1,inplace=True)
df_train.drop("Location of Institute",axis=1,inplace=True)
df_train.drop("Place of birth",axis=1,inplace=True)
```

```
# Checking total null values
df_train.isna().sum()
```

```
Patient Age                                1427
Genes in mother's side                        0
Inherited from father                       306
```

```
Maternal gene                                        2810
Paternal gene                                           0
Blood cell count (mcL)                                  0
Mother's age                                         6036
Father's age                                         5986
Status                                                  0
Respiratory Rate (breaths/min)                       2149
Heart Rate (rates/min                                2113
Test 1                                               2127
Test 2                                               2152
Test 3                                               2147
Test 4                                               2140
Test 5                                               2170
Parental consent                                     2125
Follow-up                                            2166
Gender                                               2173
Birth asphyxia                                       2139
Autopsy shows birth defect (if applicable)           1026
Folic acid details (peri-conceptional)               2117
H/O serious maternal illness                         2152
H/O radiation exposure (x-ray)                       2153
H/O substance abuse                                  2195
Assisted conception IVF/ART                          2122
History of anomalies in previous pregnancies         2172
No. of previous abortion                             2162
Birth defects                                        2154
White Blood cell count (thousand per microliter)     2148
Blood test result                                    2145
Symptom 1                                            2155
Symptom 2                                            2222
Symptom 3                                            2101
Symptom 4                                            2113
Symptom 5                                            2153
Genetic Disorder                                     2146
Disorder Subclass                                    2168
dtype: int64
```

```
df_train["Patient Age"]
```

```
0         2.0
1         4.0
2         6.0
3        12.0
4        11.0
         ...
22078     4.0
22079     8.0
22080     8.0
22081     7.0
22082    11.0
Name: Patient Age, Length: 22083, dtype: float64
```

```
# Filling Null values with mode
df_train["Patient Age"].fillna(str(df_train["Patient Age"].mode().values[0]),inplace=True)
df_train["Inherited from father"].fillna(str(df_train["Inherited from father"].mode().values[0]),inpl
df_train["Maternal gene"].fillna(str(df_train["Maternal gene"].mode().values[0]),inplace=True)
df_train["Mother's age"].fillna(str(df_train["Mother's age"].mode().values[0]),inplace=True)
df_train["Father's age"].fillna(str(df_train["Father's age"].mode().values[0]),inplace=True)
df_train["Respiratory Rate (breaths/min)"].fillna(str(df_train["Respiratory Rate (breaths/min)"].mode
df_train["Heart Rate (rates/min"].fillna(str(df_train["Heart Rate (rates/min"].mode().values[0]),inpl
df_train["Test 1"].fillna(str(df_train["Test 1"].mode().values[0]),inplace=True)
df_train["Test 2"].fillna(str(df_train["Test 2"].mode().values[0]),inplace=True)
df_train["Test 3"].fillna(str(df_train["Test 3"].mode().values[0]),inplace=True)
df_train["Test 4"].fillna(str(df_train["Test 4"].mode().values[0]),inplace=True)
df_train["Test 5"].fillna(str(df_train["Test 5"].mode().values[0]),inplace=True)
df_train["Parental consent"].fillna(str(df_train["Parental consent"].mode().values[0]),inplace=True)
df_train["Follow-up"].fillna(str(df_train["Follow-up"].mode().values[0]),inplace=True)
df_train["Gender"].fillna(str(df_train["Gender"].mode().values[0]),inplace=True)
df_train["Birth asphyxia"].fillna(str(df_train["Birth asphyxia"].mode().values[0]),inplace=True)
df_train["Autopsy shows birth defect (if applicable)"].fillna(str(df_train["Autopsy shows birth defec
df_train["Folic acid details (peri-conceptional)"].fillna(str(df_train["Folic acid details (peri-conc
df_train["H/O serious maternal illness"].fillna(str(df_train["H/O serious maternal illness"].mode().v
df_train["H/O radiation exposure (x-ray)"].fillna(str(df_train["H/O radiation exposure (x-ray)"].mode
df_train["H/O substance abuse"].fillna(str(df_train["H/O substance abuse"].mode().values[0]),inplace=
df_train["Assisted conception IVF/ART"].fillna(str(df_train["Assisted conception IVF/ART"].mode().val
df_train["History of anomalies in previous pregnancies"].fillna(str(df_train["History of anomalies ir
df_train["No. of previous abortion"].fillna(str(df_train["No. of previous abortion"].mode().values[0]
df_train["Birth defects"].fillna(str(df_train["Birth defects"].mode().values[0]),inplace=True)
df_train["White Blood cell count (thousand per microliter)"].fillna(str(df_train["White Blood cell co
df_train["Blood test result"].fillna(str(df_train["Blood test result"].mode().values[0]),inplace=True
df_train["Symptom 1"].fillna(str(df_train["Symptom 1"].mode().values[0]),inplace=True)
df_train["Symptom 2"].fillna(str(df_train["Symptom 2"].mode().values[0]),inplace=True)
df_train["Symptom 3"].fillna(str(df_train["Symptom 3"].mode().values[0]),inplace=True)
df_train["Symptom 4"].fillna(str(df_train["Symptom 4"].mode().values[0]),inplace=True)
df_train["Symptom 5"].fillna(str(df_train["Symptom 5"].mode().values[0]),inplace=True)
df_train["Genetic Disorder"].fillna(str(df_train["Genetic Disorder"].mode().values[0]),inplace=True)
df_train["Disorder Subclass"].fillna(str(df_train["Disorder Subclass"].mode().values[0]),inplace=True


# Checking if any null value is present
df_train.isna().sum()
```

```
Patient Age                                0
Genes in mother's side                     0
Inherited from father                      0
Maternal gene                              0
Paternal gene                              0
Blood cell count (mcL)                     0
Mother's age                               0
Father's age                               0
Status                                     0
Respiratory Rate (breaths/min)             0
Heart Rate (rates/min                      0
Test 1                                     0
Test 2                                     0
Test 3                                     0
Test 4                                     0
Test 5                                     0
Parental consent                           0
Follow-up                                  0
```

```
Gender                                                0
Birth asphyxia                                        0
Autopsy shows birth defect (if applicable)            0
Folic acid details (peri-conceptional)                0
H/O serious maternal illness                          0
H/O radiation exposure (x-ray)                        0
H/O substance abuse                                   0
Assisted conception IVF/ART                           0
History of anomalies in previous pregnancies          0
No. of previous abortion                              0
Birth defects                                         0
White Blood cell count (thousand per microliter)      0
Blood test result                                     0
Symptom 1                                             0
Symptom 2                                             0
Symptom 3                                             0
Symptom 4                                             0
Symptom 5                                             0
Genetic Disorder                                      0
Disorder Subclass                                     0
dtype: int64
```

```
df_train.head()
```

	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Mother's age	Father's age	Sta
0	2.0	Yes	No	Yes	No	4.760603	23.0	20.0	A
1	4.0	Yes	Yes	No	No	4.910669	23.0	23.0	Decea
2	6.0	Yes	No	No	No	4.893297	41.0	22.0	A
3	12.0	Yes	No	Yes	No	4.705280	21.0	20.0	Decea
4	11.0	Yes	No	Yes	Yes	4.720703	32.0	20.0	A

5 rows × 38 columns

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22083 entries, 0 to 22082
Data columns (total 38 columns):
 #   Column                                          Non-Null Count  Dtype  
---  ------                                          --------------  -----  
 0   Patient Age                                     22083 non-null  object 
 1   Genes in mother's side                          22083 non-null  object 
 2   Inherited from father                           22083 non-null  object 
 3   Maternal gene                                   22083 non-null  object 
 4   Paternal gene                                   22083 non-null  object 
 5   Blood cell count (mcL)                          22083 non-null  float64
 6   Mother's age                                    22083 non-null  object 
 7   Father's age                                    22083 non-null  object 
 8   Status                                          22083 non-null  object 
 9   Respiratory Rate (breaths/min)                  22083 non-null  object 
 10  Heart Rate (rates/min                           22083 non-null  object 
 11  Test 1                                          22083 non-null  object 
 12  Test 2                                          22083 non-null  object 
 13  Test 3                                          22083 non-null  object 
 14  Test 4                                          22083 non-null  object 
 15  Test 5                                          22083 non-null  object 
 16  Parental consent                                22083 non-null  object 
 17  Follow-up                                       22083 non-null  object 
 18  Gender                                          22083 non-null  object 
 19  Birth asphyxia                                  22083 non-null  object 
 20  Autopsy shows birth defect (if applicable)      22083 non-null  object 
 21  Folic acid details (peri-conceptional)          22083 non-null  object 
 22  H/O serious maternal illness                    22083 non-null  object 
 23  H/O radiation exposure (x-ray)                  22083 non-null  object 
 24  H/O substance abuse                             22083 non-null  object 
 25  Assisted conception IVF/ART                     22083 non-null  object 
 26  History of anomalies in previous pregnancies    22083 non-null  object 
 27  No. of previous abortion                        22083 non-null  object 
 28  Birth defects                                   22083 non-null  object 
 29  White Blood cell count (thousand per microliter)  22083 non-null  object 
 30  Blood test result                               22083 non-null  object 
 31  Symptom 1                                       22083 non-null  object 
 32  Symptom 2                                       22083 non-null  object 
 33  Symptom 3                                       22083 non-null  object 
 34  Symptom 4                                       22083 non-null  object 
 35  Symptom 5                                       22083 non-null  object 
 36  Genetic Disorder                                22083 non-null  object 
 37  Disorder Subclass                               22083 non-null  object 
dtypes: float64(1), object(37)
memory usage: 6.4+ MB
```

```
# Optional Column name change
# for column in df_train:
#     columnSeriesObj = df_train[column]
#     print('Column Name : ', column)
#     print('Column Contents : ', columnSeriesObj.values)
#     print("----------------------------------------------")
```

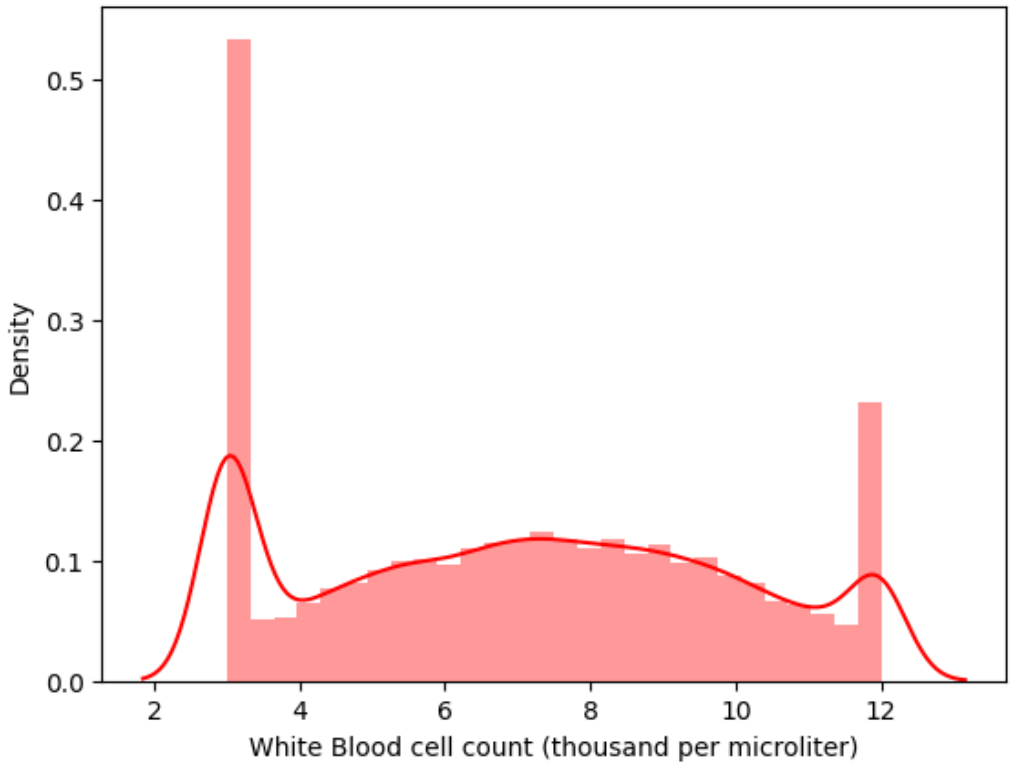
```
df_train.columns
```

```
Index(['Patient Age', 'Genes in mother's side', 'Inherited from father',
       'Maternal gene', 'Paternal gene', 'Blood cell count (mcL)',
       'Mother's age', 'Father's age', 'Status',
       'Respiratory Rate (breaths/min)', 'Heart Rate (rates/min', 'Test 1',
```

```
       'Test 2', 'Test 3', 'Test 4', 'Test 5', 'Parental consent', 'Follow-up',
       'Gender', 'Birth asphyxia',
       'Autopsy shows birth defect (if applicable)',
       'Folic acid details (peri-conceptional)',
       'H/O serious maternal illness', 'H/O radiation exposure (x-ray)',
       'H/O substance abuse', 'Assisted conception IVF/ART',
       'History of anomalies in previous pregnancies',
       'No. of previous abortion', 'Birth defects',
       'White Blood cell count (thousand per microliter)', 'Blood test result',
       'Symptom 1', 'Symptom 2', 'Symptom 3', 'Symptom 4', 'Symptom 5',
       'Genetic Disorder', 'Disorder Subclass'],
      dtype='object')
```

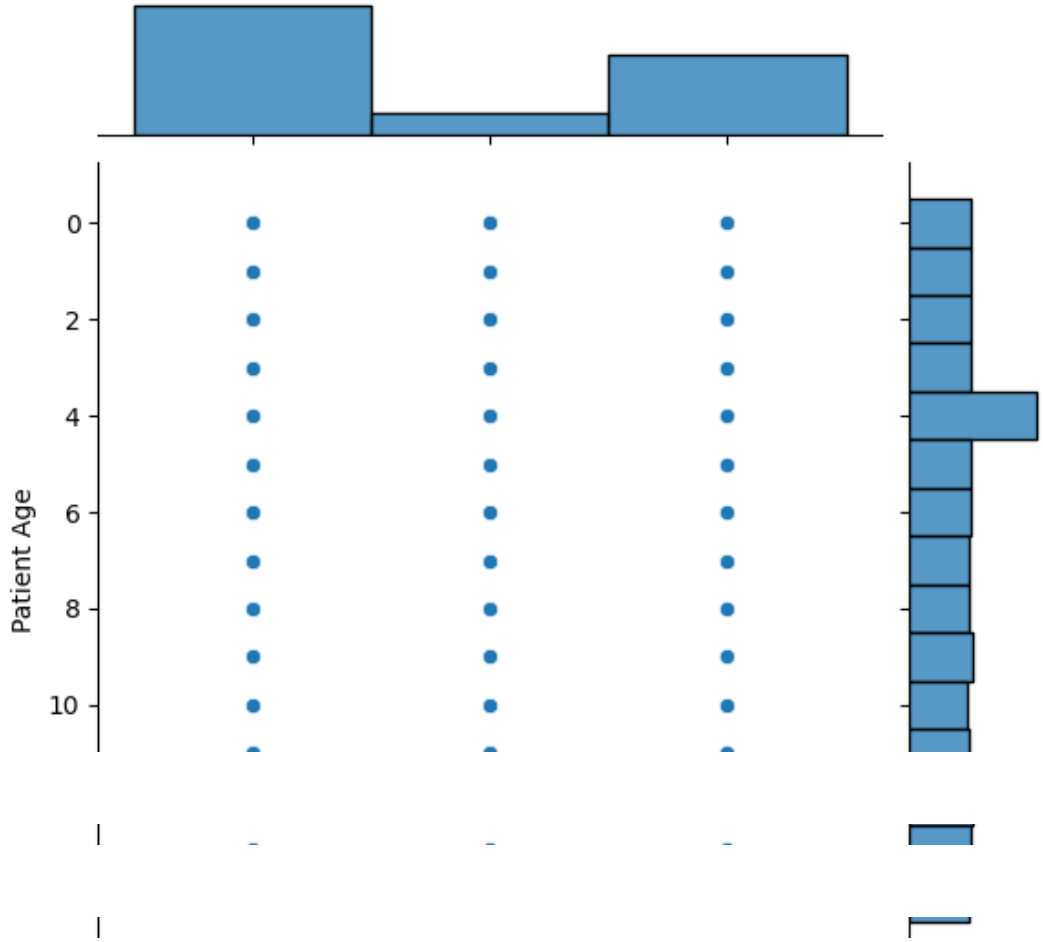
```
sns.distplot(df_train["White Blood cell count (thousand per microliter)"],color = "red")
```

```
<Axes: xlabel='White Blood cell count (thousand per microliter)', ylabel='Density'>
```



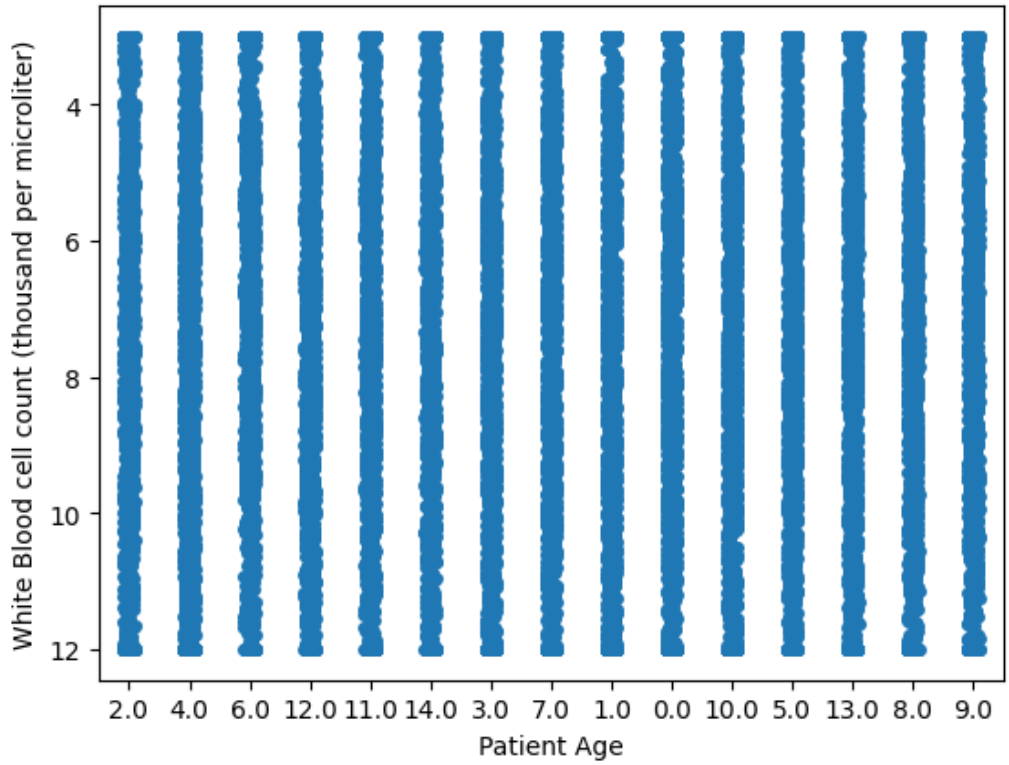

```
sns.jointplot(x="Genetic Disorder",y="Patient Age",data=df_train)
```

```
<seaborn.axisgrid.JointGrid at 0x7e15b4cf4b20>
```



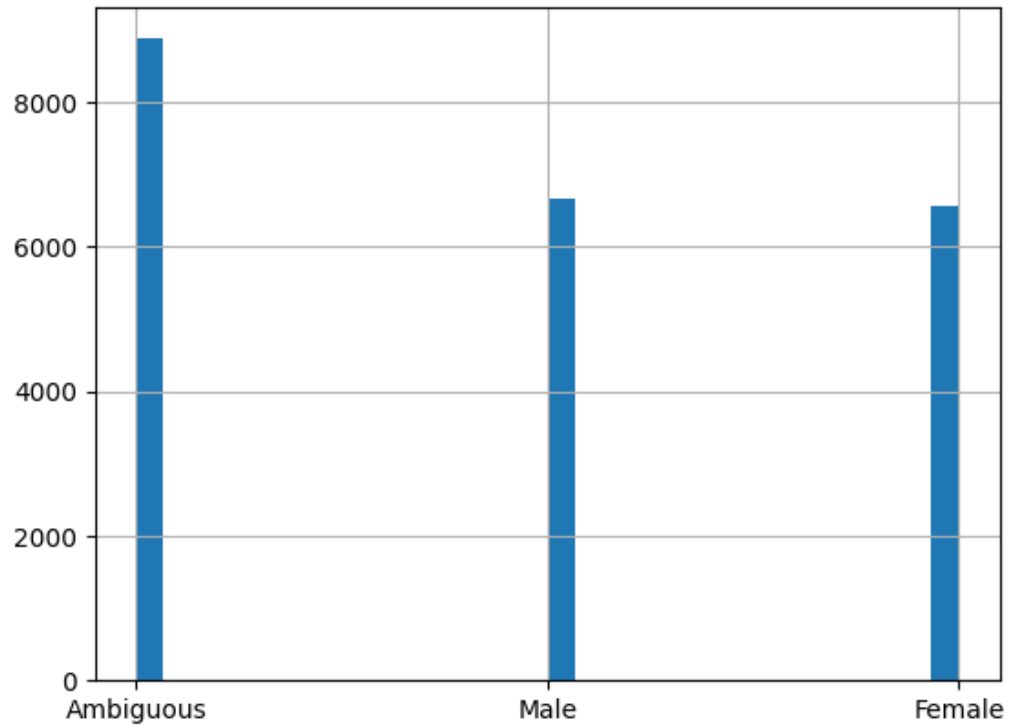

```
sns.stripplot(x="Patient Age",y="White Blood cell count (thousand per microliter)",data=df_train,jitt
```

```
<Axes: xlabel='Patient Age', ylabel='White Blood cell count (thousand per
microliter)'>
```



```
df_train["Gender"].hist(bins=30)
```

```
<Axes: >
```




```
# Changing from yes or no to numerical(1 or 0)
df_train["Genes in mother's side"]=[1 if i.strip()== "Yes" else 0 for i in df_train["Genes in mother'
df_train["Inherited from father"]=[1 if i.strip()== "Yes" else 0 for i in df_train["Inherited from fa
df_train["Maternal gene"]=[1 if i.strip()== "Yes" else 0 for i in df_train["Maternal gene"]]
df_train["Paternal gene"]=[1 if i.strip()== "Yes" else 0 for i in df_train["Paternal gene"]]
df_train["Parental consent"]=[1 if i.strip()== "Yes" else 0 for i in df_train["Parental consent"]]
df_train["Birth asphyxia"]=[1 if i.strip()== "Yes" else 0 for i in df_train["Birth asphyxia"]]
df_train["Folic acid details (peri-conceptional)"]=[1 if i.strip()== "Yes" else 0 for i in df_train["
df_train["H/O radiation exposure (x-ray)"]=[1 if i.strip()== "Yes" else 0 for i in df_train["H/O radi
df_train["H/O substance abuse"]=[1 if i.strip()== "Yes" else 0 for i in df_train["H/O substance abuse
df_train["Assisted conception IVF/ART"]=[1 if i.strip()== "Yes" else 0 for i in df_train["Assisted co
df_train["History of anomalies in previous pregnancies"]=[1 if i.strip()== "Yes" else 0 for i in df_t
df_train["H/O serious maternal illness"]=[1 if i.strip()=="Yes" else 0 for i in df_train["H/O serious


df_train.head()
```

	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Mother's age	Father's age	Sta
0	2.0	1	0	1	0	4.760603	23.0	20.0	A
1	4.0	1	1	0	0	4.910669	23.0	23.0	Decea
2	6.0	1	0	0	0	4.893297	41.0	22.0	A

```
# Check if you changed the column name
# for column in df_train:
#     columnSeriesObj = df_train[column]
#     print('Column Name : ', column)
#     print('Column Contents : ', columnSeriesObj.values)
#     print("----------------------------------------------")


# Checking the unique elements in Categorical Columns
print("Status: ",df_train["Status"].unique())
print("Respiratory Rate (breaths/min): ",df_train["Respiratory Rate (breaths/min)"].unique())
print("Heart Rate (rates/min: ",df_train["Heart Rate (rates/min"].unique())
print("Follow-up: ",df_train["Follow-up"].unique())
print("Gender: ",df_train["Gender"].unique())
print("Autopsy shows birth defect (if applicable): ",df_train["Autopsy shows birth defect (if applica
print("Birth defects: ",df_train["Birth defects"].unique())
print("Blood test result: ",df_train["Blood test result"].unique())
print("Genetic Disorder: ",df_train["Genetic Disorder"].unique())
print("Disorder Subclass: ",df_train["Disorder Subclass"].unique())
```

```
Status:  ['Alive' 'Deceased']
Respiratory Rate (breaths/min):  ['Normal (30-60)' 'Tachypnea']
Heart Rate (rates/min:  ['Normal' 'Tachycardia']
Follow-up:  ['High' 'Low']
Gender:  ['Ambiguous' 'Male' 'Female']
Autopsy shows birth defect (if applicable):  ['Not applicable' 'None' 'No' 'Yes']
Birth defects:  ['Singular' 'Multiple']
Blood test result:  ['slightly abnormal' 'normal' 'inconclusive' 'abnormal']
Genetic Disorder:  ['Mitochondrial genetic inheritance disorders'
 'Multifactorial genetic inheritance disorders'
 'Single-gene inheritance diseases']
Disorder Subclass:  ["Leber's hereditary optic neuropathy" 'Cystic fibrosis' 'Diabetes'
 'Leigh syndrome' 'Cancer' 'Tay-Sachs' 'Hemochromatosis'
 'Mitochondrial myopathy' "Alzheimer's"]
```

```
# plots
df_train.head()
```

	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Mother's age	Father's age	Status	Res (brea
0	2.0	1	0	1	0	4.760603	23.0	20.0	Alive	Norm
1	4.0	1	1	0	0	4.910669	23.0	23.0	Deceased	T
2	6.0	1	0	0	0	4.893297	41.0	22.0	Alive	Norm
3	12.0	1	0	1	0	4.705280	21.0	20.0	Deceased	T
4	11.0	1	0	1	1	4.720703	32.0	20.0	Alive	T

5 rows × 38 columns

```
# Changing Categorical Values to Numerical Values
#Alive':1 'Deceased:0'
df_train["Status"]=[1 if i.strip()== "Alive" else 0 for i in df_train["Status"]]
#Normal (30-60):1' 'Tachypnea:0
df_train["Respiratory Rate (breaths/min)"]=[1 if i.strip()== "Normal (30-60)" else 0 for i in df_trai
#Normal:1' 'Tachycardia:0
df_train["Heart Rate (rates/min"]=[1 if i.strip()== "Normal" else 0 for i in df_train["Heart Rate (ra
#High:1, Low:0
df_train["Follow-up"]=[1 if i.strip()== "High" else 0 for i in df_train["Follow-up"]]
#['Singular' 'Multiple']
df_train["Birth defects"]=[1 if i.strip()== "Singular" else 0 for i in df_train["Birth defects"]]
#1: male 0: female 2: ambiguous
df_train["Gender"]=[1 if i.strip()== "Male" else 0 if i.strip() == "Female" else 2 for i in df_train[
#Not applicable:3' 'None:2' 'No:0' 'Yes:1'
df_train["Autopsy shows birth defect (if applicable)"]=[1 if i.strip()== "Yes" else 0 if i.strip() ==
#'slightly abnormal':1, 'normal':0, 'inconclusive':2 'abnormal:3']
df_train["Blood test result"]=[1 if i.strip()== "slightly abnormal" else 0 if i.strip() == "normal" e
#'Mitochondrial genetic inheritance disorders':1,'Multifactorial genetic inheritance disorders':0'Sin
df_train["Genetic Disorder"]=[1 if i.strip()== "Mitochondrial genetic inheritance disorders" else 0 i
#Leber's hereditary optic neuropathy:1
#Cystic fibrosis:0
#Diabetes:2
#Leigh syndrome:3
#Cancer:4
#Tay-Sachs:5
#Hemochromatosis:6
#Mitochondrial myopathy:7
#Alzheimer's:8
df_train["Disorder Subclass"]=[1 if i.strip()== "Leber's hereditary optic neuropathy"
                               else 0 if i.strip() == "Cystic fibrosis"
                              else 2 if i.strip()=="Diabetes"
                              else 3 if i.strip()=="Leigh syndrome"
                              else 4 if i.strip()=="Cancer"
                              else 5 if i.strip()=="Tay-Sachs"
                              else 6 if i.strip()=="Hemochromatosis"
                              else 7 if i.strip()=="Mitochondrial myopathy"
                              else 8 for i in df_train["Disorder Subclass"]]
```

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22083 entries, 0 to 22082
Data columns (total 38 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   Patient Age                                     22083 non-null  object
 1   Genes in mother's side                          22083 non-null  int64
 2   Inherited from father                           22083 non-null  int64
 3   Maternal gene                                   22083 non-null  int64
 4   Paternal gene                                   22083 non-null  int64
 5   Blood cell count (mcL)                          22083 non-null  float64
 6   Mother's age                                    22083 non-null  object
 7   Father's age                                    22083 non-null  object
 8   Status                                          22083 non-null  int64
 9   Respiratory Rate (breaths/min)                  22083 non-null  int64
```

```
 10  Heart Rate (rates/min                                22083 non-null  int64
 11  Test 1                                               22083 non-null  object
 12  Test 2                                               22083 non-null  object
 13  Test 3                                               22083 non-null  object
 14  Test 4                                               22083 non-null  object
 15  Test 5                                               22083 non-null  object
 16  Parental consent                                     22083 non-null  int64
 17  Follow-up                                            22083 non-null  int64
 18  Gender                                               22083 non-null  int64
 19  Birth asphyxia                                       22083 non-null  int64
 20  Autopsy shows birth defect (if applicable)           22083 non-null  int64
 21  Folic acid details (peri-conceptional)               22083 non-null  int64
 22  H/O serious maternal illness                         22083 non-null  int64
 23  H/O radiation exposure (x-ray)                       22083 non-null  int64
 24  H/O substance abuse                                  22083 non-null  int64
 25  Assisted conception IVF/ART                          22083 non-null  int64
 26  History of anomalies in previous pregnancies         22083 non-null  int64
 27  No. of previous abortion                             22083 non-null  object
 28  Birth defects                                        22083 non-null  int64
 29  White Blood cell count (thousand per microliter)     22083 non-null  object
 30  Blood test result                                    22083 non-null  int64
 31  Symptom 1                                            22083 non-null  object
 32  Symptom 2                                            22083 non-null  object
 33  Symptom 3                                            22083 non-null  object
 34  Symptom 4                                            22083 non-null  object
 35  Symptom 5                                            22083 non-null  object
 36  Genetic Disorder                                     22083 non-null  int64
 37  Disorder Subclass                                    22083 non-null  int64
dtypes: float64(1), int64(22), object(15)
memory usage: 6.4+ MB
```

```
# Changing the datatype to float
df_train = df_train.apply(pd.to_numeric,downcast="float")
```

```
#total symptom
df_train["total symptom"]=(df_train["Symptom 1"]+df_train["Symptom 2"]+df_train["Symptom 3"]+df_train
df_train["sum of Mother's and fathers age avg"]=(df_train["Mother's age"]+df_train["Father's age"]) /
```

```
#Dropping Symptom Columns
df_train.drop(["Symptom 1","Symptom 2","Symptom 3","Symptom 4","Symptom 5"],axis=1,inplace=True)
```

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22083 entries, 0 to 22082
Data columns (total 35 columns):
 #   Column                                               Non-Null Count  Dtype
---  ------                                               --------------  -----
 0   Patient Age                                          22083 non-null  float32
 1   Genes in mother's side                               22083 non-null  float32
 2   Inherited from father                                22083 non-null  float32
 3   Maternal gene                                        22083 non-null  float32
 4   Paternal gene                                        22083 non-null  float32
 5   Blood cell count (mcL)                               22083 non-null  float32
 6   Mother's age                                         22083 non-null  float32
 7   Father's age                                         22083 non-null  float32
 8   Status                                               22083 non-null  float32
```

```
 9   Respiratory Rate (breaths/min)                    22083 non-null  float32
 10  Heart Rate (rates/min                             22083 non-null  float32
 11  Test 1                                            22083 non-null  float32
 12  Test 2                                            22083 non-null  float32
 13  Test 3                                            22083 non-null  float32
 14  Test 4                                            22083 non-null  float32
 15  Test 5                                            22083 non-null  float32
 16  Parental consent                                  22083 non-null  float32
 17  Follow-up                                         22083 non-null  float32
 18  Gender                                            22083 non-null  float32
 19  Birth asphyxia                                    22083 non-null  float32
 20  Autopsy shows birth defect (if applicable)        22083 non-null  float32
 21  Folic acid details (peri-conceptional)            22083 non-null  float32
 22  H/O serious maternal illness                      22083 non-null  float32
 23  H/O radiation exposure (x-ray)                    22083 non-null  float32
 24  H/O substance abuse                               22083 non-null  float32
 25  Assisted conception IVF/ART                       22083 non-null  float32
 26  History of anomalies in previous pregnancies      22083 non-null  float32
 27  No. of previous abortion                          22083 non-null  float32
 28  Birth defects                                     22083 non-null  float32
 29  White Blood cell count (thousand per microliter)  22083 non-null  float32
 30  Blood test result                                 22083 non-null  float32
 31  Genetic Disorder                                  22083 non-null  float32
 32  Disorder Subclass                                 22083 non-null  float32
 33  total symptom                                     22083 non-null  float32
 34  sum of Mother's and fathers age avg               22083 non-null  float32
dtypes: float32(35)
memory usage: 2.9 MB
```

```
df_train.head()
```

	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Mother's age	Father's age	Statu
0	2.0	1.0	0.0	1.0	0.0	4.760603	23.0	20.0	1.
1	4.0	1.0	1.0	0.0	0.0	4.910669	23.0	23.0	0.
2	6.0	1.0	0.0	0.0	0.0	4.893297	41.0	22.0	1.
3	12.0	1.0	0.0	1.0	0.0	4.705280	21.0	20.0	0.
4	11.0	1.0	0.0	1.0	1.0	4.720703	32.0	20.0	1.

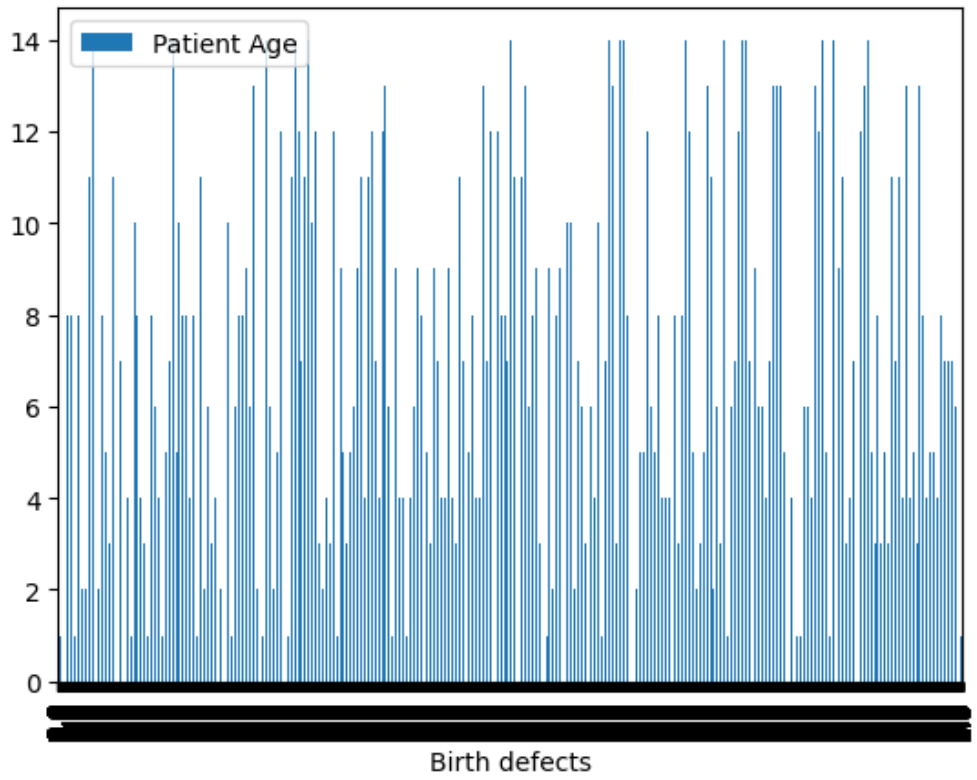
5 rows × 35 columns

```
df_train.Status.value_counts()
```

```
1.0    11083
0.0    11000
Name: Status, dtype: int64
```

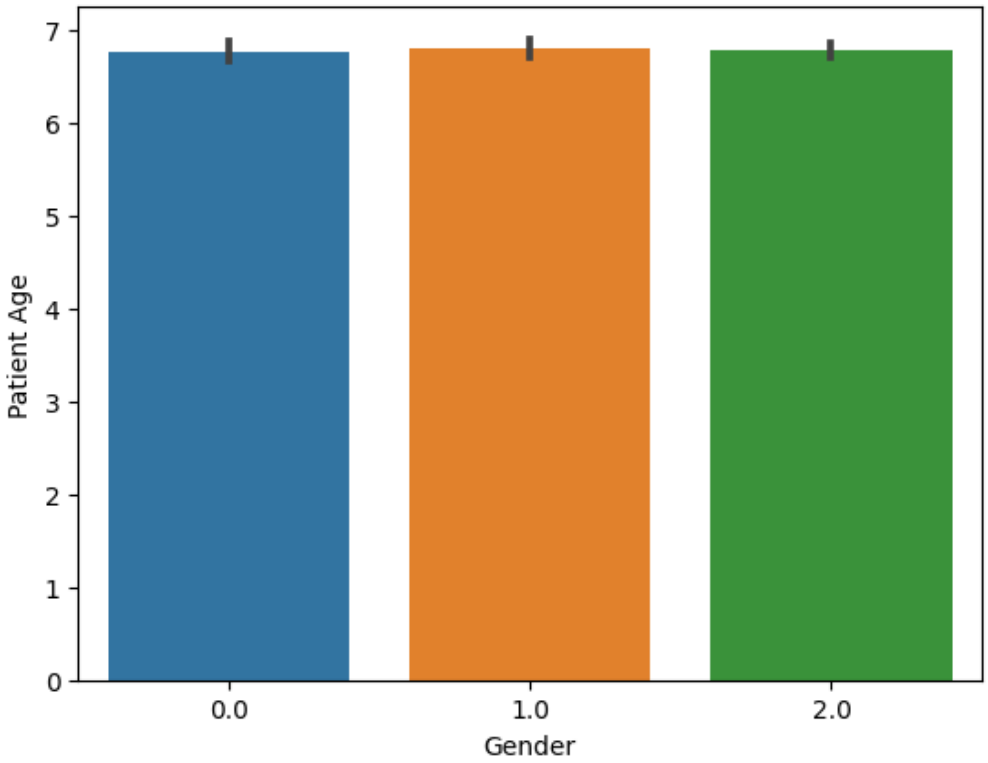
```
df_train.plot.bar(y="Patient Age",x="Birth defects")
```

```
<Axes: xlabel='Birth defects'>
```



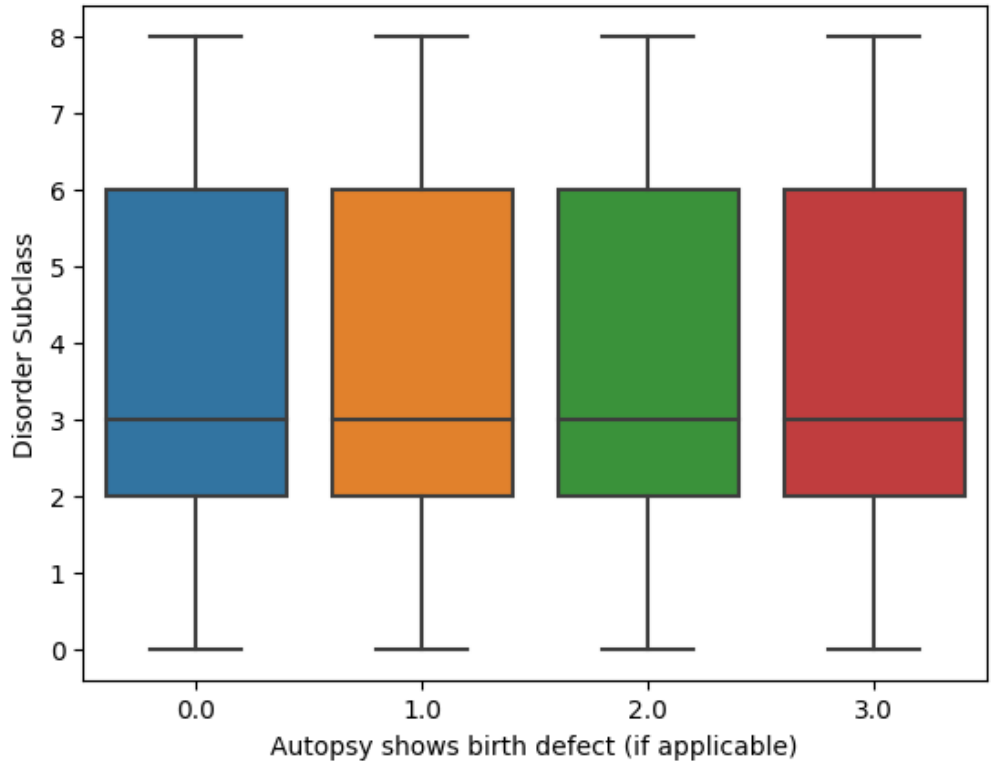

```
sns.barplot(x="Gender",y="Patient Age",data=df_train)
```

```
<Axes: xlabel='Gender', ylabel='Patient Age'>
```



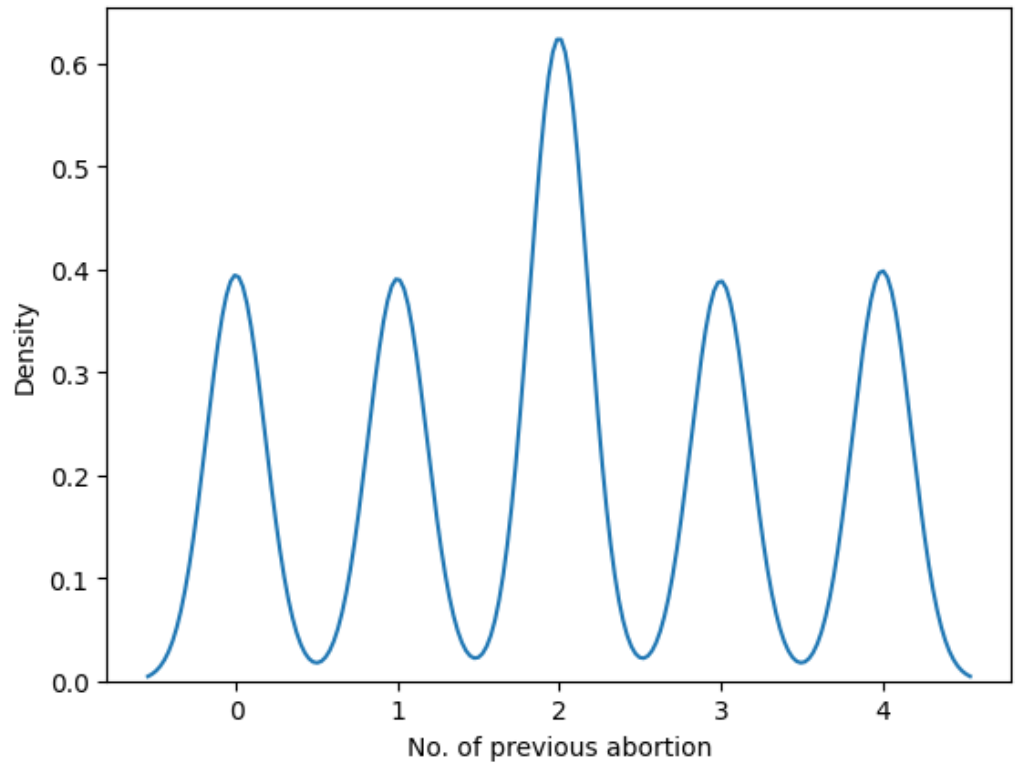

```
sns.boxplot(x="Autopsy shows birth defect (if applicable)",y="Disorder Subclass",data=df_train)
```

```
<Axes: xlabel='Autopsy shows birth defect (if applicable)', ylabel='Disorder Subclass'>
```



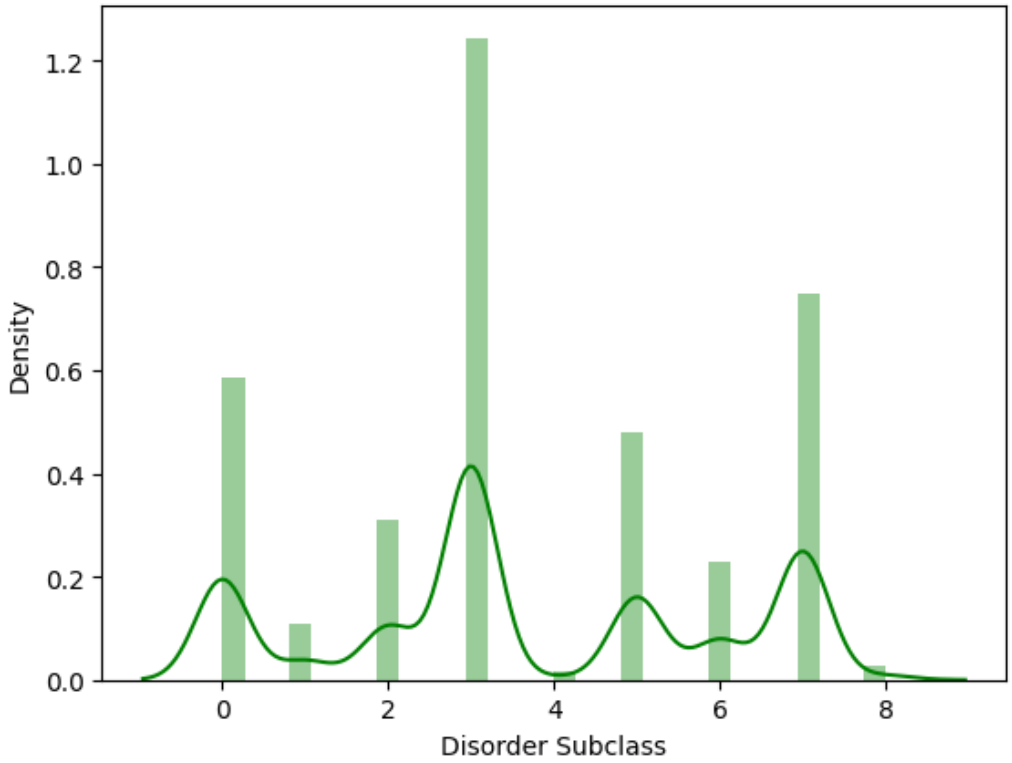

```
sns.kdeplot(df_train["No. of previous abortion"],palette="dark")
```

```
<Axes: xlabel='No. of previous abortion', ylabel='Density'>
```



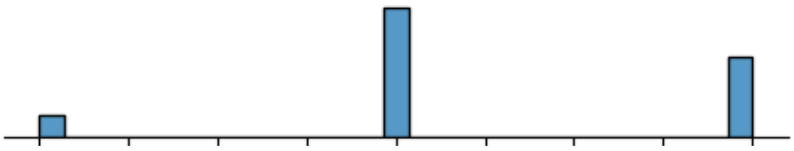

```
# Distplot
sns.distplot(df_train['Disorder Subclass'],color="green",bins=30)
```

```
<Axes: xlabel='Disorder Subclass', ylabel='Density'>
```




```
#JointPlot
plt.figure(figsize=(12,6))
sns.jointplot(x=df_train["Genetic Disorder"],y=df_train['Patient Age'],kind="hex")
```

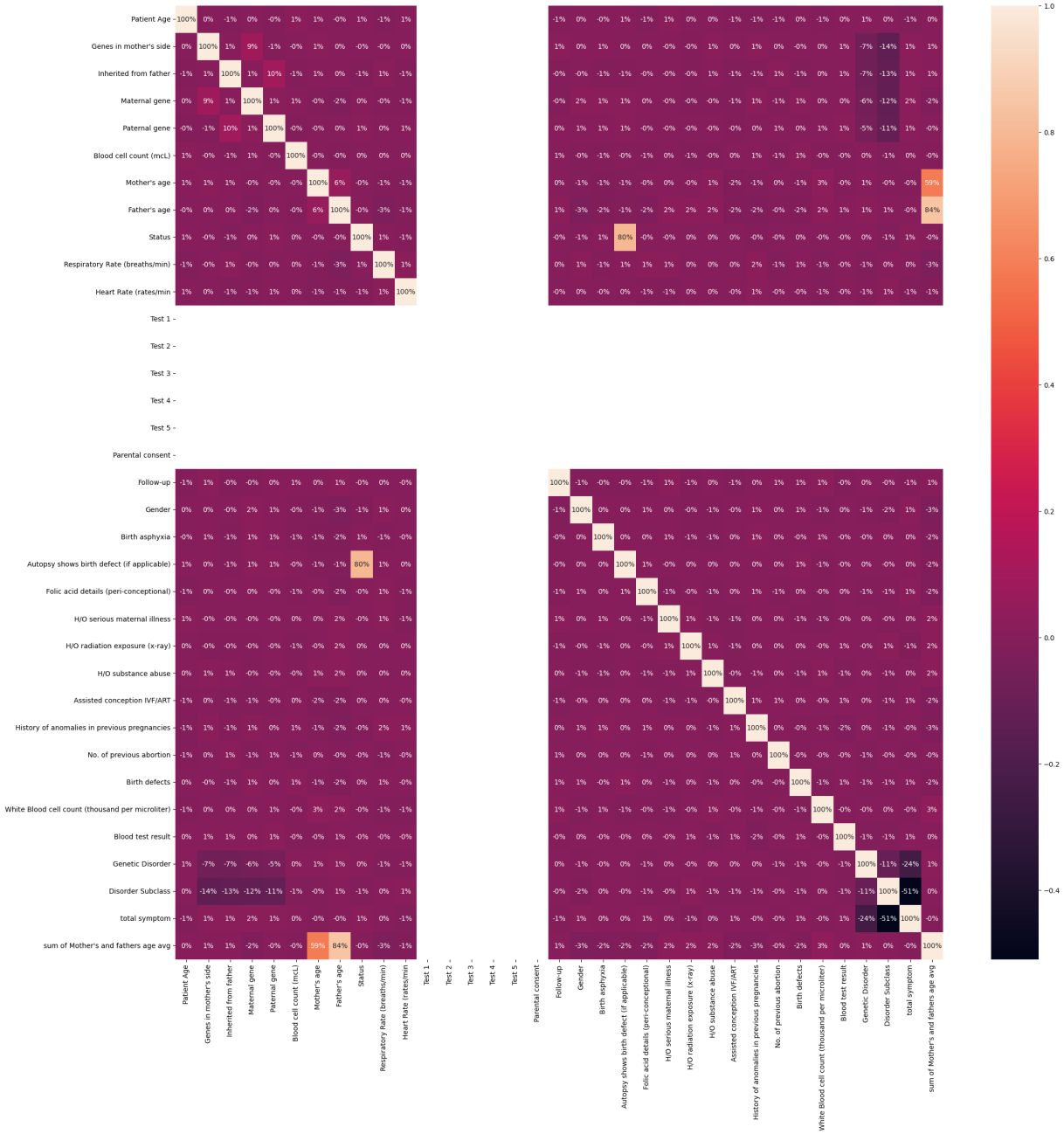
```
<seaborn.axisgrid.JointGrid at 0x7e1592780ac0>
<Figure size 1200x600 with 0 Axes>
```



```
df_train.corr()
```

	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Mother's age	F
Patient Age	1.000000	0.003452	-0.008275	0.001492	-0.004422	0.010155	0.008203	-0
Genes in mother's side	0.003452	1.000000	0.008960	0.089605	-0.007389	-0.002403	0.010247	0
Inherited from father	-0.008275	0.008960	1.000000	0.009222	0.095115	-0.007677	0.006349	0
Maternal gene	0.001492	0.089605	0.009222	1.000000	0.011829	0.008119	-0.004247	-0
Paternal gene	-0.004422	-0.007389	0.095115	0.011829	1.000000	-0.003494	-0.001070	0
Blood cell count (mcL)	0.010155	-0.002403	-0.007677	0.008119	-0.003494	1.000000	-0.001129	-0
Mother's age	0.008203	0.010247	0.006349	-0.004247	-0.001070	-0.001129	1.000000	0
Father's age	-0.000949	0.000520	0.003769	-0.022747	0.000636	-0.003498	0.059002	1
Status	0.007764	-0.000221	-0.012293	0.003566	0.013799	0.003149	-0.001519	-0
Respiratory Rate (breaths/min)	-0.011186	-0.001917	0.011765	-0.003921	0.000572	0.002200	-0.008097	-0
Heart Rate (rates/min	0.008489	0.001019	-0.010575	-0.005682	0.005119	0.000185	-0.008515	-0
Test 1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Test 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Test 3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Test 4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Test 5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Parental consent	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Follow-up	-0.005463	0.009939	-0.003019	-0.003337	0.003752	0.005366	0.002463	0
Gender	0.000485	0.004320	-0.002209	0.020249	0.006780	-0.004604	-0.011823	-0
Birth asphyxia	-0.001758	0.005706	-0.009471	0.010482	0.006934	-0.007196	-0.014797	-0
Autopsy shows birth defect (if applicable)	0.007851	0.000360	-0.008126	0.009456	0.013202	-0.000833	-0.013375	-0
Folic acid details (peri-conceptional)	-0.008526	0.000115	-0.001314	0.003564	-0.000652	-0.005324	-0.002391	-0

```
plt.figure(figsize=(25,25))
sns.heatmap(df_train.iloc[:,0:39].corr(),annot=True,fmt=".0%")
plt.show()
```




```
df_train.columns
```

```
Index(['Patient Age', 'Genes in mother's side', 'Inherited from father',
       'Maternal gene', 'Paternal gene', 'Blood cell count (mcL)',
       'Mother's age', 'Father's age', 'Status',
       'Respiratory Rate (breaths/min)', 'Heart Rate (rates/min', 'Test 1',
```

```
       'Test 2', 'Test 3', 'Test 4', 'Test 5', 'Parental consent', 'Follow-up',
       'Gender', 'Birth asphyxia',
       'Autopsy shows birth defect (if applicable)',
       'Folic acid details (peri-conceptional)',
       'H/O serious maternal illness', 'H/O radiation exposure (x-ray)',
       'H/O substance abuse', 'Assisted conception IVF/ART',
       'History of anomalies in previous pregnancies',
       'No. of previous abortion', 'Birth defects',
       'White Blood cell count (thousand per microliter)', 'Blood test result',
       'Genetic Disorder', 'Disorder Subclass', 'total symptom',
       'sum of Mother's and fathers age avg'],
      dtype='object')
```

```
df_test.head()
```

	Patient Id	Patient Age	Genes in mother's side	Inherited from father	Maternal gene	Paternal gene	Blood cell count (mcL)	Patient First Name	Family Name	Father na
0	PID0x4175	6	No	Yes	No	No	4.981655	Charles	NaN	Kc
1	PID0x21f5	10	Yes	No	NaN	Yes	5.118890	Catherine	NaN	Home
2	PID0x49b8	5	No	NaN	No	No	4.876204	James	NaN	Danie
3	PID0x2d97	13	No	Yes	Yes	No	4.687767	Brian	NaN	Orvi
4	PID0x58da	5	No	NaN	NaN	Yes	5.152362	Gary	NaN	Issi

5 rows × 43 columns

```
df_train['Genetic Disorder'].head()
```

```
0    1.0
1    1.0
2    0.0
3    1.0
4    0.0
Name: Genetic Disorder, dtype: float32
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 3)
x,y = df_train.loc[:,df_train.columns != 'Genetic Disorder'], df_train.loc[:,'Genetic Disorder']
knn.fit(x,y)
prediction = knn.predict(x)
print('Prediction: {}'.format(prediction))
```

```
Prediction: [2. 1. 0. ... 1. 1. 1.]
```

```
x.columns
```

```
Index(['Patient Age', 'Genes in mother's side', 'Inherited from father',
       'Maternal gene', 'Paternal gene', 'Blood cell count (mcL)',
```

```
       'Mother's age', 'Father's age', 'Status',
       'Respiratory Rate (breaths/min)', 'Heart Rate (rates/min', 'Test 1',
       'Test 2', 'Test 3', 'Test 4', 'Test 5', 'Parental consent', 'Follow-up',
       'Gender', 'Birth asphyxia',
       'Autopsy shows birth defect (if applicable)',
       'Folic acid details (peri-conceptional)',
       'H/O serious maternal illness', 'H/O radiation exposure (x-ray)',
       'H/O substance abuse', 'Assisted conception IVF/ART',
       'History of anomalies in previous pregnancies',
       'No. of previous abortion', 'Birth defects',
       'White Blood cell count (thousand per microliter)', 'Blood test result',
       'Disorder Subclass', 'total symptom',
       'sum of Mother's and fathers age avg'],
      dtype='object')
```

```
y
```

```
0        1.0
1        1.0
2        0.0
3        1.0
4        0.0
        ...
22078    1.0
22079    0.0
22080    1.0
22081    1.0
22082    0.0
Name: Genetic Disorder, Length: 22083, dtype: float32
```

KNN

```
# train test split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2,random_state = 1)
knn = KNeighborsClassifier(n_neighbors = 3)
x,y = df_train.loc[:,df_train.columns != 'Status'], df_train.loc[:,'Status']
knn.fit(x_train,y_train)
prediction = knn.predict(x_test)
#print('Prediction: {}'.format(prediction))
print('With KNN (K=3) accuracy is: ',knn.score(x_test,y_test)) # accuracy
```

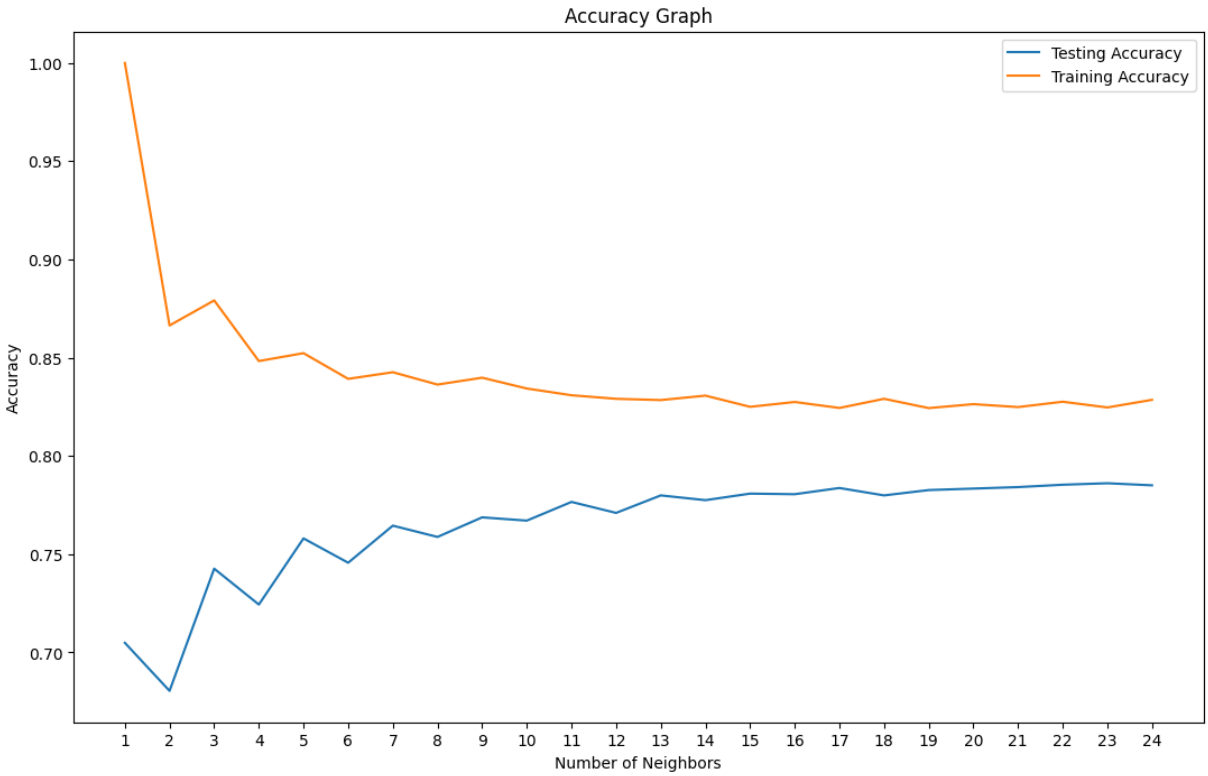
```
With KNN (K=3) accuracy is:  0.7502829975096219
```

```
# train test split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,random_state = 1)
knn = KNeighborsClassifier(n_neighbors = 3)
x,y = df_train.loc[:,df_train.columns != 'Genetic Disorder'], df_train.loc[:,'Genetic Disorder']
knn.fit(x_train,y_train)
prediction = knn.predict(x_test)
#print('Prediction: {}'.format(prediction))
print('With KNN (K=3) accuracy is: ',knn.score(x_test,y_test)) # accuracy
```

```
With KNN (K=3) accuracy is:  0.7426415094339622
```

```
neig = np.arange(1, 25)
train_accuracy = []
test_accuracy = []
# Loop over different values of k
for i, k in enumerate(neig):
    # k from 1 to 25(exclude)
    knn = KNeighborsClassifier(n_neighbors=k)
    # Fit with knn
    knn.fit(x_train,y_train)
    #train accuracy
    train_accuracy.append(knn.score(x_train, y_train))
    # test accuracy
    test_accuracy.append(knn.score(x_test, y_test))

# Plot
plt.figure(figsize=[13,8])
plt.plot(neig, test_accuracy, label = 'Testing Accuracy')
plt.plot(neig, train_accuracy, label = 'Training Accuracy')
plt.legend()
plt.title('Accuracy Graph')
plt.xlabel('Number of Neighbors')
plt.ylabel('Accuracy')
plt.xticks(neig)
plt.savefig('graph.png')
plt.show()
print("Best accuracy is {} with K = {}".format(np.max(test_accuracy),1+test_accuracy.index(np.max(tes
```




```
Best accuracy is 0.7861132075471698 with K = 23
```

Random Forest

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

x, y = df_train.loc[:, df_train.columns != 'Genetic Disorder'], df_train.loc[:, 'Genetic Disorder']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.8, random_state=1)

# Create a Random Forest classifier
rf = RandomForestClassifier(n_estimators=120, random_state=3)  # You can adjust n_estimators as neede

rf.fit(x_train, y_train)

prediction = rf.predict(x_test)
accuracy = rf.score(x_test, y_test)
print('Random Forest accuracy for predicting Genetic Disorder is:', accuracy)
```

```
Random Forest accuracy for predicting Genetic Disorder is: 0.8540782249391521
```

```
x, y = df_train.loc[:, df_train.columns != 'Genetic Disorder'], df_train.loc[:, 'Genetic Disorder']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.8, random_state=1)

# Create a Random Forest classifier
rf = RandomForestClassifier(n_estimators=120, random_state=1)  # You can adjust n_estimators as neede

rf.fit(x_train, y_train)

prediction = rf.predict(x_test)
accuracy = rf.score(x_test, y_test)
print('Random Forest accuracy for predicting Disorder Subclass is:', accuracy)
```

```
Random Forest accuracy for predicting Disorder Subclass is: 0.8486443652006566
```

```
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt
import numpy as np

n_estimators = np.arange(1, 25)
train_accuracy = []
test_accuracy = []

for n in n_estimators:
    # Create a Random Forest classifier with 'n' estimators
    rf = RandomForestClassifier(n_estimators=n, random_state=1)
    rf.fit(x_train, y_train)

    # Train accuracy
    train_accuracy.append(rf.score(x_train, y_train))

    # Test accuracy
    test_accuracy.append(rf.score(x_test, y_test))
```

```
# Plot
plt.figure(figsize=[13, 8])
plt.plot(n_estimators, test_accuracy, label='Testing Accuracy')
plt.plot(n_estimators, train_accuracy, label='Training Accuracy')
plt.legend()
plt.title('Number of Estimators VS Accuracy')
plt.xlabel('Number of Estimators')
plt.ylabel('Accuracy')
plt.xticks(n_estimators)
plt.savefig('rf_graph.png')
plt.show()

best_accuracy = max(test_accuracy)
best_n_estimators = n_estimators[test_accuracy.index(best_accuracy)]

print("Best accuracy is {} with n_estimators = {}".format(best_accuracy, best_n_estimators))
```

Extra Tree Classifier

```
# Import necessary libraries
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score


#  Split the data into training and testing sets
# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

x, y = df_train.loc[:, df_train.columns != 'Genetic Disorder'], df_train.loc[:, 'Genetic Disorder']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1)


extra_trees_classifier = ExtraTreesClassifier(n_estimators=100, random_state=42)


extra_trees_classifier.fit(x_train,y_train)


y_pred = extra_trees_classifier.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.8750282997509622
```

```
n_estimators = np.arange(1, 25)
train_accuracy = []
test_accuracy = []

for n in n_estimators:
    # Create a Extra Trees Classifier with 'n' estimators
    rf = ExtraTreesClassifier(n_estimators=n, random_state=1)
    rf.fit(x_train, y_train)

    # Train accuracy
    train_accuracy.append(rf.score(x_train, y_train))

    # Test accuracy
    test_accuracy.append(rf.score(x_test, y_test))

# Plot
plt.figure(figsize=[13, 8])
plt.plot(n_estimators, test_accuracy, label='Testing Accuracy')
plt.plot(n_estimators, train_accuracy, label='Training Accuracy')
```