cancer-prediction-system

July 3, 2024

Import Libraries

[55]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use("seaborn-whitegrid")
```

[4]:
```
# Importing our Dataset

cancer_patient = pd.read_csv("cancer patient data sets.csv")
cancer_patient.head()
```

[4]:

	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	\
0	P1	33	1	2	4	5	
1	P10	17	1	3	1	5	
2	P100	35	1	4	5	6	
3	P1000	37	1	7	7	7	
4	P101	46	1	6	8	7	

	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	\
0	4	3	2	2	
1	3	4	2	2	
2	5	5	4	6	
3	7	6	7	7	
4	7	7	6	7	

	…	Fatigue	Weight Loss	Shortness of Breath	Wheezing	\
0	…	3	4	2	2	
1	…	1	3	7	8	
2	…	8	7	9	2	
3	…	4	2	3	1	
4	…	3	2	4	1	

	Swallowing Difficulty	Clubbing of Finger Nails	Frequent Cold	Dry Cough	\
0	3	1	2	3	
1	6	2	1	7	

```
2                    1                    4            6          7
3                    4                    5            6          7
4                    4                    2            4          2

   Snoring   Level
0        4     Low
1        2  Medium
2        2    High
3        5    High
4        3    High

[5 rows x 25 columns]
```

[5]:
```
len(cancer_patient)
```

[5]:
```
1000
```

[6]:
```
cancer_patient.info();
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 25 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Patient Id                1000 non-null   object
 1   Age                       1000 non-null   int64
 2   Gender                    1000 non-null   int64
 3   Air Pollution             1000 non-null   int64
 4   Alcohol use               1000 non-null   int64
 5   Dust Allergy              1000 non-null   int64
 6   OccuPational Hazards      1000 non-null   int64
 7   Genetic Risk              1000 non-null   int64
 8   chronic Lung Disease      1000 non-null   int64
 9   Balanced Diet             1000 non-null   int64
 10  Obesity                   1000 non-null   int64
 11  Smoking                   1000 non-null   int64
 12  Passive Smoker            1000 non-null   int64
 13  Chest Pain                1000 non-null   int64
 14  Coughing of Blood         1000 non-null   int64
 15  Fatigue                   1000 non-null   int64
 16  Weight Loss               1000 non-null   int64
 17  Shortness of Breath       1000 non-null   int64
 18  Wheezing                  1000 non-null   int64
 19  Swallowing Difficulty     1000 non-null   int64
 20  Clubbing of Finger Nails  1000 non-null   int64
 21  Frequent Cold             1000 non-null   int64
 22  Dry Cough                 1000 non-null   int64
```

```
 23  Snoring                  1000 non-null   int64
 24  Level                    1000 non-null   object
dtypes: int64(23), object(2)
memory usage: 195.4+ KB
```

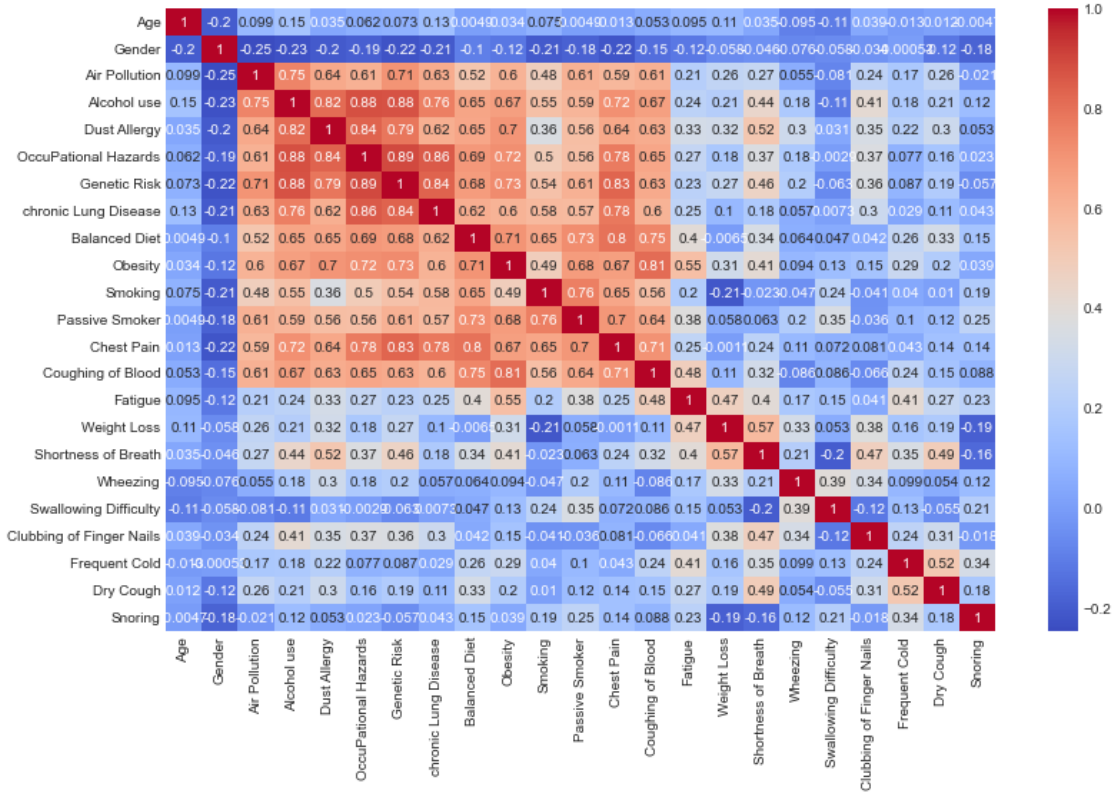
[7]:
```
cancer_patient.describe().T
```

[7]:

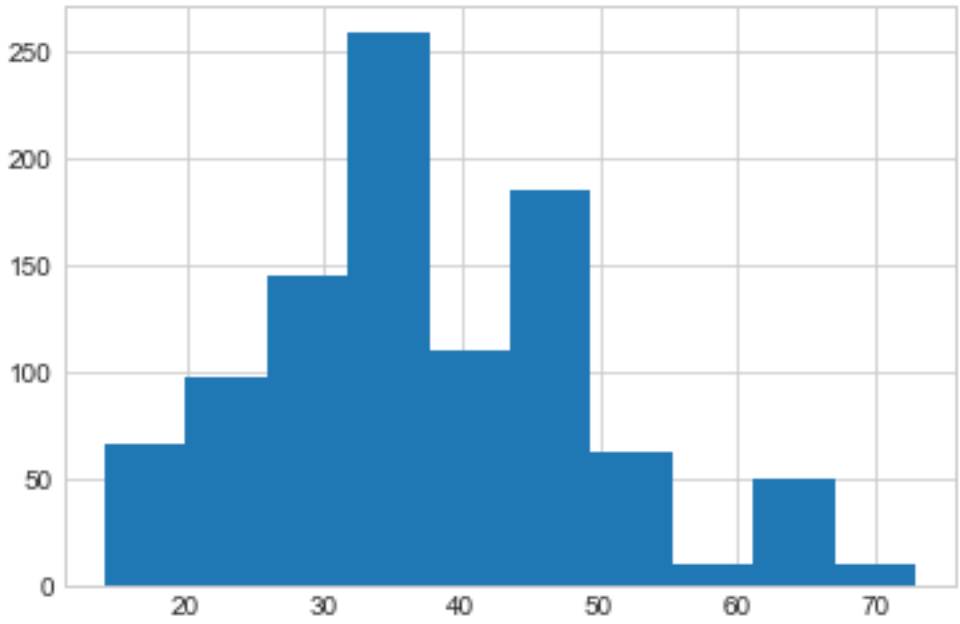
	count	mean	std	min	25%	50%	75%	max
Age	1000.0	37.174	12.005493	14.0	27.75	36.0	45.0	73.0
Gender	1000.0	1.402	0.490547	1.0	1.00	1.0	2.0	2.0
Air Pollution	1000.0	3.840	2.030400	1.0	2.00	3.0	6.0	8.0
Alcohol use	1000.0	4.563	2.620477	1.0	2.00	5.0	7.0	8.0
Dust Allergy	1000.0	5.165	1.980833	1.0	4.00	6.0	7.0	8.0
OccuPational Hazards	1000.0	4.840	2.107805	1.0	3.00	5.0	7.0	8.0
Genetic Risk	1000.0	4.580	2.126999	1.0	2.00	5.0	7.0	7.0
chronic Lung Disease	1000.0	4.380	1.848518	1.0	3.00	4.0	6.0	7.0
Balanced Diet	1000.0	4.491	2.135528	1.0	2.00	4.0	7.0	7.0
Obesity	1000.0	4.465	2.124921	1.0	3.00	4.0	7.0	7.0
Smoking	1000.0	3.948	2.495902	1.0	2.00	3.0	7.0	8.0
Passive Smoker	1000.0	4.195	2.311778	1.0	2.00	4.0	7.0	8.0
Chest Pain	1000.0	4.438	2.280209	1.0	2.00	4.0	7.0	9.0
Coughing of Blood	1000.0	4.859	2.427965	1.0	3.00	4.0	7.0	9.0
Fatigue	1000.0	3.856	2.244616	1.0	2.00	3.0	5.0	
Weight Loss	1000.0	3.855	2.206546	1.0	2.00	3.0	6.0	
Shortness of Breath	1000.0	4.240	2.285087	1.0	2.00	4.0	6.0	
Wheezing	1000.0	3.777	2.041921	1.0	2.00	4.0	5.0	
Swallowing Difficulty	1000.0	3.746	2.270383	1.0	2.00	4.0	5.0	
Clubbing of Finger Nails	1000.0	3.923	2.388048	1.0	2.00	4.0	5.0	
Frequent Cold	1000.0	3.536	1.832502	1.0	2.00	3.0	5.0	
Dry Cough	1000.0	3.853	2.039007	1.0	2.00	4.0	6.0	
Snoring	1000.0	2.926	1.474686	1.0	2.00	3.0	4.0	

```
Fatigue                     9.0
Weight Loss                 8.0
Shortness of Breath         9.0
Wheezing                    8.0
Swallowing Difficulty       8.0
Clubbing of Finger Nails    9.0
Frequent Cold               7.0
Dry Cough                   7.0
Snoring                     7.0
```

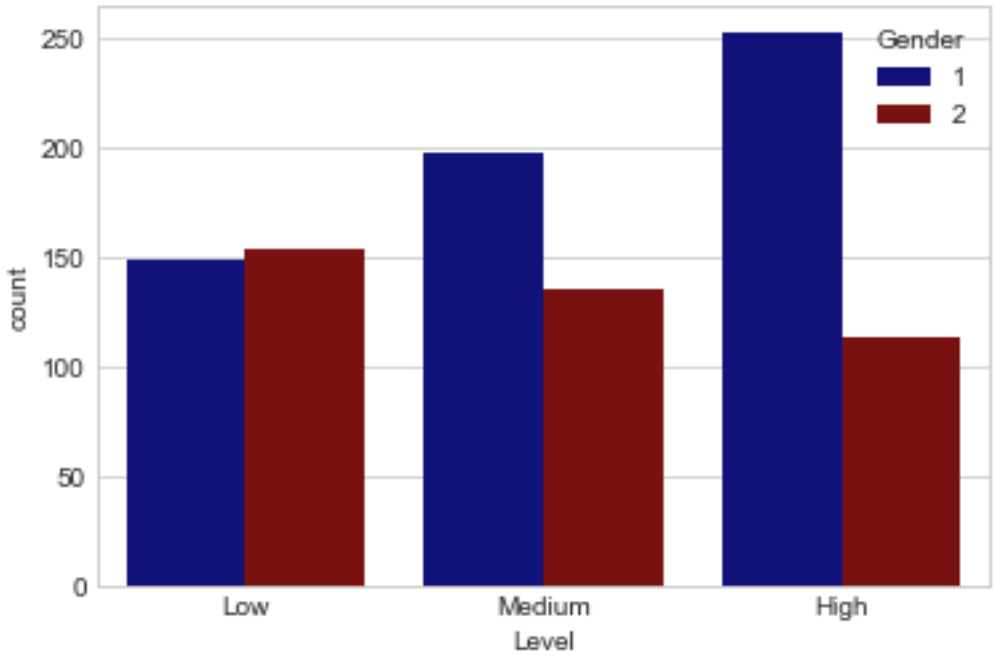
[8]:
```
fig = plt.figure(figsize = (13,8))
sns.heatmap(cancer_patient.corr(),cmap='coolwarm',annot=True);
```



[9]:
```
fig, ax = plt.subplots()
hist = ax.hist(x = cancer_patient["Age"]);
```



[10]:
```
#Required outside of function. This needs to be activated first when plotting␣
 ↪in every code block
fig, ax = plt.subplots()

#Count plot
plot = sns.countplot(data = cancer_patient, x='Level', hue='Gender',␣
 ↪palette=['darkblue','darkred'])
```



[11]:
```
cancer_patient.columns
```

[11]:
```
Index(['Patient Id', 'Age', 'Gender', 'Air Pollution', 'Alcohol use',
       'Dust Allergy', 'OccuPational Hazards', 'Genetic Risk',
       'chronic Lung Disease', 'Balanced Diet', 'Obesity', 'Smoking',
       'Passive Smoker', 'Chest Pain', 'Coughing of Blood', 'Fatigue',
       'Weight Loss', 'Shortness of Breath', 'Wheezing',
       'Swallowing Difficulty', 'Clubbing of Finger Nails', 'Frequent Cold',
       'Dry Cough', 'Snoring', 'Level'],
      dtype='object')
```

Cancer found in people age over 50

[12]:
```
cancer_over50 = cancer_patient[cancer_patient["Age"] > 50]
cancer_over50.head()
```

[12]:

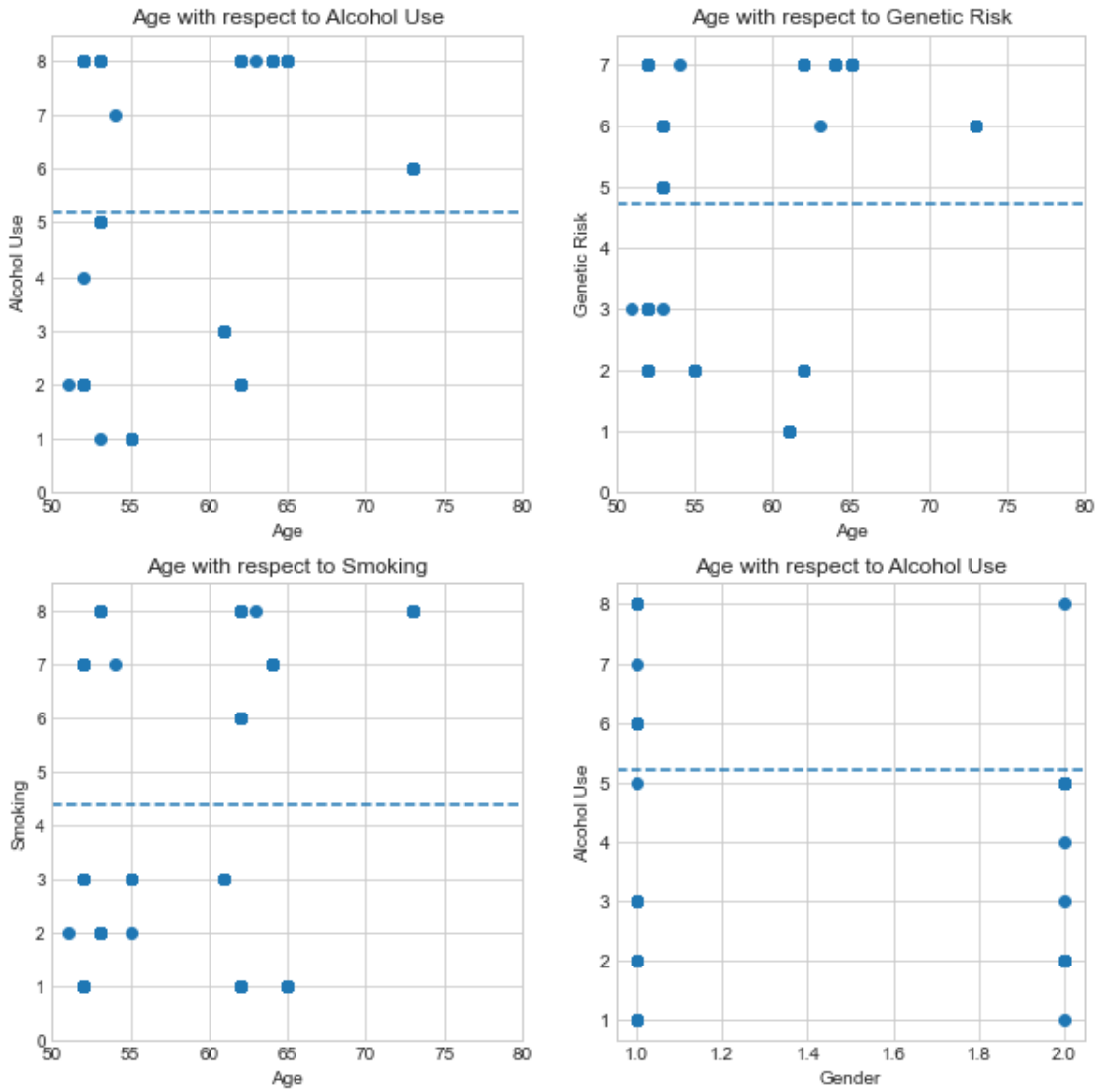
	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	\
6	P103	52	2	2	4	5	
11	P108	64	2	6	8	7	
15	P111	73	1	5	6	6	
21	P117	53	2	4	5	6	
22	P118	62	1	6	8	7	

	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	\
6	4	3	2	2	

```
11                    7              7                    6              7
15                    5              6                    5              6
21                    5              5                    4              6
22                    7              7                    6              7

    …  Fatigue  Weight Loss  Shortness of Breath  Wheezing  \
6   …        3            4                    2         2
11  …        9            6                    5         7
15  …        4            3                    6         2
21  …        8            7                    9         2
22  …        3            2                    4         1

    Swallowing Difficulty  Clubbing of Finger Nails  Frequent Cold  Dry Cough  \
6                       3                         1              2          3
11                      2                         4              3          1
15                      1                         2              1          6
21                      1                         4              6          7
22                      4                         2              4          2

    Snoring   Level
6         4     Low
11        4    High
15        2  Medium
21        2    High
22        3    High

[5 rows x 25 columns]
```

[13]:
```
# Making Subplots
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows = 2, ncols= 2, figsize=(10,␣
 ↪10))

# Adding Data to the plot
scatter = ax1.scatter(x = cancer_over50["Age"], y = cancer_over50["Alcohol␣
 ↪use"], cmap = "winter")

# For Plot ax1
ax1.set(title = "Age with respect to Alcohol Use",
        xlabel = "Age",
        ylabel = "Alcohol Use")
ax1.axhline(cancer_over50["Alcohol use"].mean(),
           linestyle = "--");
ax1.set_xlim([50, 80])
ax1.set_ylim([0, 8.5])

# For Plot ax2
```

```
scatter = ax2.scatter(x = cancer_over50["Age"], y = cancer_over50["Genetic␣
 ↪Risk"])
ax2.set(title = "Age with respect to Genetic Risk", xlabel = "Age", ylabel =␣
 ↪"Genetic Risk")
ax2.axhline(cancer_over50["Genetic Risk"].mean(),
           linestyle = "--");
ax2.set_xlim([50, 80])
ax2.set_ylim([0, 7.5])

# For Plot ax3
scatter = ax3.scatter(x = cancer_over50["Age"], y = cancer_over50["Smoking"])
ax3.set(title = "Age with respect to Smoking", xlabel = "Age", ylabel =␣
 ↪"Smoking")
ax3.axhline(cancer_over50["Smoking"].mean(),
           linestyle = "--");
ax3.set_xlim([50, 80])
ax3.set_ylim([0, 8.5])

# For Plot ax4
scatter = ax4.scatter(x = cancer_over50["Gender"], y = cancer_over50["Alcohol␣
 ↪use"])
ax4.set(title = "Age with respect to Alcohol Use", xlabel = "Gender", ylabel =␣
 ↪"Alcohol Use")
ax4.axhline(cancer_over50["Alcohol use"].mean(),
           linestyle = "--");
```



[14]:
```
cancer_over50.head()
```

[14]:

	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet
6	P103	52	2	2	4	5	4	3	2	2
11	P108	64	2	6	8	7	7	7	6	7
15	P111	73	1	5	6	6	5	6	5	6
21	P117	53	2	4	5	6				
22	P118	62	1	6	8	7				

```
21                    5             5                    4             6
22                    7             7                    6             7

    …  Fatigue  Weight Loss  Shortness of Breath  Wheezing  \
6   …        3            4                    2         2
11  …        9            6                    5         7
15  …        4            3                    6         2
21  …        8            7                    9         2
22  …        3            2                    4         1

    Swallowing Difficulty  Clubbing of Finger Nails  Frequent Cold  Dry Cough  \
6                       3                         1              2          3
11                      2                         4              3          1
15                      1                         2              1          6
21                      1                         4              6          7
22                      4                         2              4          2

    Snoring   Level
6         4     Low
11        4    High
15        2  Medium
21        2    High
22        3    High

[5 rows x 25 columns]
```

[15]:
```
len(cancer_patient), len(cancer_over50)
```

[15]:
```
(1000, 134)
```

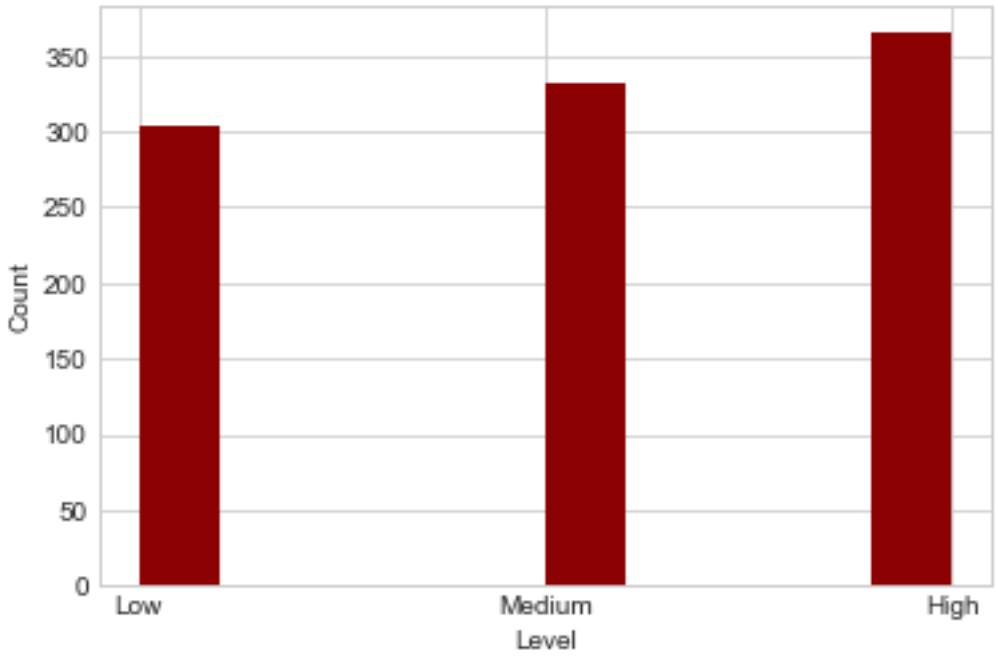
There are only 134 patients who are Over 50 so we analyse the entire data irresespective of age to achieve fruitfull results later.

[16]:
```
cancer_patient.columns
```

[16]:
```
Index(['Patient Id', 'Age', 'Gender', 'Air Pollution', 'Alcohol use',
       'Dust Allergy', 'OccuPational Hazards', 'Genetic Risk',
       'chronic Lung Disease', 'Balanced Diet', 'Obesity', 'Smoking',
       'Passive Smoker', 'Chest Pain', 'Coughing of Blood', 'Fatigue',
       'Weight Loss', 'Shortness of Breath', 'Wheezing',
       'Swallowing Difficulty', 'Clubbing of Finger Nails', 'Frequent Cold',
       'Dry Cough', 'Snoring', 'Level'],
      dtype='object')
```

[17]:
```
fig, ax = plt.subplots()

histt = ax.hist(x = cancer_patient["Level"], bins = 10, color ='darkred')
```

```
ax.set(xlabel = "Level", ylabel = "Count");
```



[18]:
```
cancer_patient.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 25 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Patient Id                1000 non-null   object
 1   Age                       1000 non-null   int64
 2   Gender                    1000 non-null   int64
 3   Air Pollution             1000 non-null   int64
 4   Alcohol use               1000 non-null   int64
 5   Dust Allergy              1000 non-null   int64
 6   OccuPational Hazards      1000 non-null   int64
 7   Genetic Risk              1000 non-null   int64
 8   chronic Lung Disease      1000 non-null   int64
 9   Balanced Diet             1000 non-null   int64
 10  Obesity                   1000 non-null   int64
 11  Smoking                   1000 non-null   int64
 12  Passive Smoker            1000 non-null   int64
 13  Chest Pain                1000 non-null   int64
 14  Coughing of Blood         1000 non-null   int64
```

```
 15  Fatigue                   1000 non-null   int64
 16  Weight Loss               1000 non-null   int64
 17  Shortness of Breath       1000 non-null   int64
 18  Wheezing                  1000 non-null   int64
 19  Swallowing Difficulty     1000 non-null   int64
 20  Clubbing of Finger Nails  1000 non-null   int64
 21  Frequent Cold             1000 non-null   int64
 22  Dry Cough                 1000 non-null   int64
 23  Snoring                   1000 non-null   int64
 24  Level                     1000 non-null   object
dtypes: int64(23), object(2)
memory usage: 195.4+ KB
```

As we can see Level dtype is not int so first we replace it with numbers then into type int

[20]:
```
cancer_patient["Level"].replace(["Low", "Medium", "High"], ["0", "1", "2"],␣
↪inplace=True)
```

[21]:
```
cancer_patient["Level"] = cancer_patient["Level"].astype(int)
```

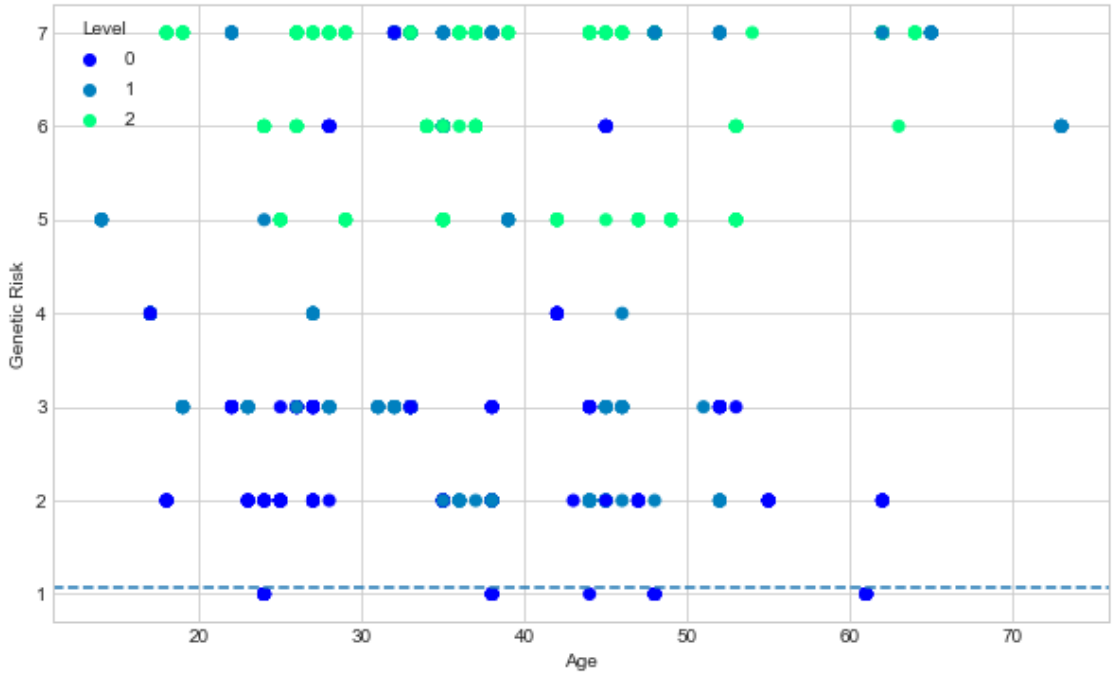
[22]:
```
cancer_patient.head().info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 25 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Patient Id                5 non-null      object
 1   Age                       5 non-null      int64
 2   Gender                    5 non-null      int64
 3   Air Pollution             5 non-null      int64
 4   Alcohol use               5 non-null      int64
 5   Dust Allergy              5 non-null      int64
 6   OccuPational Hazards      5 non-null      int64
 7   Genetic Risk              5 non-null      int64
 8   chronic Lung Disease      5 non-null      int64
 9   Balanced Diet             5 non-null      int64
 10  Obesity                   5 non-null      int64
 11  Smoking                   5 non-null      int64
 12  Passive Smoker            5 non-null      int64
 13  Chest Pain                5 non-null      int64
 14  Coughing of Blood         5 non-null      int64
 15  Fatigue                   5 non-null      int64
 16  Weight Loss               5 non-null      int64
 17  Shortness of Breath       5 non-null      int64
 18  Wheezing                  5 non-null      int64
 19  Swallowing Difficulty     5 non-null      int64
 20  Clubbing of Finger Nails  5 non-null      int64
```

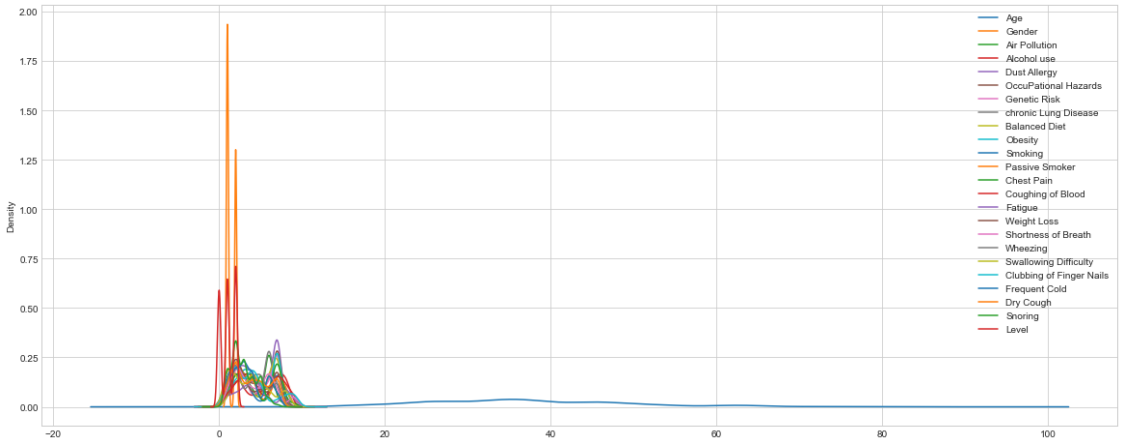
```
 21  Frequent Cold              5 non-null      int64
 22  Dry Cough                  5 non-null      int64
 23  Snoring                    5 non-null      int64
 24  Level                      5 non-null      int32
dtypes: int32(1), int64(23), object(1)
memory usage: 1.1+ KB
```

Plotting with respect to Age and Genetic Risk

[23]:
```
fig, ax = plt.subplots(figsize = (10, 6));

scatter = ax.scatter(x = cancer_patient["Age"],
                     y = cancer_patient["Genetic Risk"],
                     c = cancer_patient["Level"],
                     cmap = "winter")

ax.set(xlabel = "Age",
       ylabel = "Genetic Risk");

ax.legend(*scatter.legend_elements(), title = "Level");

ax.axhline(cancer_patient["Level"].mean(),
           linestyle = "--");
```



[24]:
```
cancer_patient.plot.kde(figsize = (20,8));
```



[25]:
```
np.array([cancer_patient["Gender"][:10]])
```

[25]:
```
array([[1, 1, 1, 1, 1, 1, 2, 2, 2, 1]], dtype=int64)
```

Number of Male & Females

[26]:
```
male = 0
female = 0
for i in cancer_patient["Gender"]:
    if i == 1:
        male += 1
    elif i == 2:
        female += 1
f"Number of Male: {male}, Number of females: {female}"
```

[26]:
```
'Number of Male: 598, Number of females: 402'
```

[27]:
```
# Make a histogram here
cancer_patient_male = cancer_patient[cancer_patient["Gender"] == 1]
cancer_patient_male.head()
```

[27]:

	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	\
0	P1	33	1	2	4	5	
1	P10	17	1	3	1	5	
2	P100	35	1	4	5	6	
3	P1000	37	1	7	7	7	
4	P101	46	1	6	8	7	

	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	\
0	4	3	2	2	
1	3	4	2	2	
2	5	5	4	6	

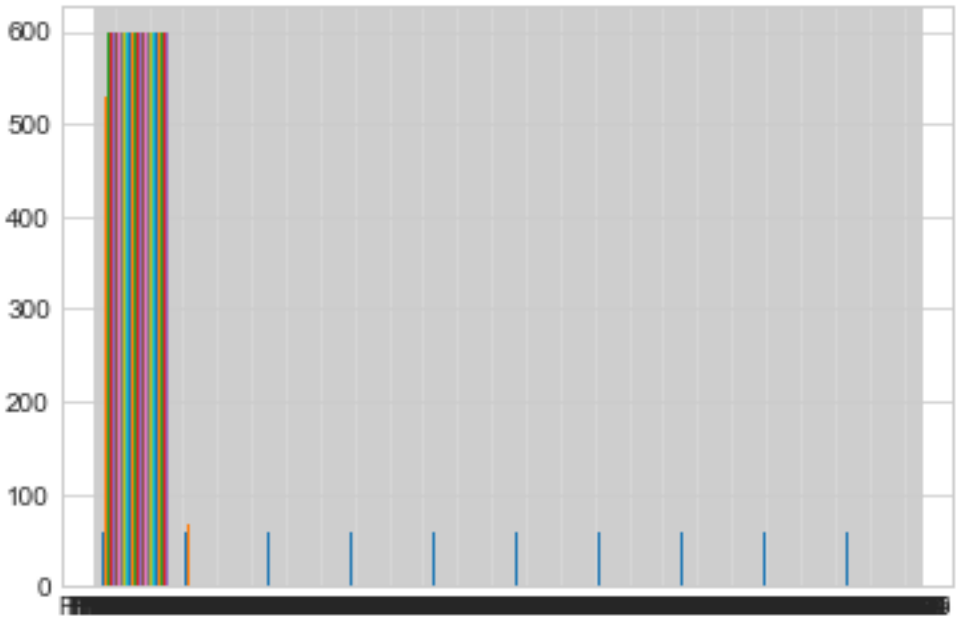
```
3                    7             6                    7             7
4                    7             7                    6             7

   …  Fatigue  Weight Loss  Shortness of Breath  Wheezing  \
0  …        3            4                    2         2
1  …        1            3                    7         8
2  …        8            7                    9         2
3  …        4            2                    3         1
4  …        3            2                    4         1

   Swallowing Difficulty  Clubbing of Finger Nails  Frequent Cold  Dry Cough  \
0                      3                         1              2          3
1                      6                         2              1          7
2                      1                         4              6          7
3                      4                         5              6          7
4                      4                         2              4          2

   Snoring  Level
0        4      0
1        2      1
2        2      2
3        5      2
4        3      2

[5 rows x 25 columns]
```

[28]:
```
cancer_patient_female = cancer_patient[cancer_patient["Gender"] == 2]
cancer_patient_female.head()
```

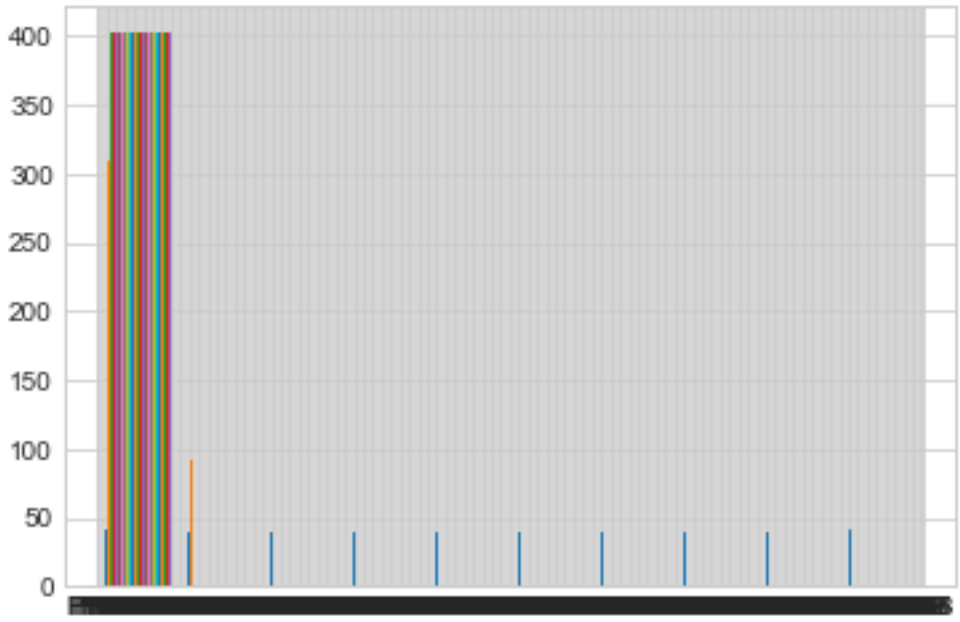
[28]:
```
   Patient Id  Age  Gender  Air Pollution  Alcohol use  Dust Allergy  \
6        P103   52       2              2            4             5
7        P104   28       2              3            1             4
8        P105   35       2              4            5             6
11       P108   64       2              6            8             7
12       P109   39       2              4            5             6

    OccuPational Hazards  Genetic Risk  chronic Lung Disease  Balanced Diet  \
6                      4             3                     2              2
7                      3             2                     3              4
8                      5             6                     5              5
11                     7             7                     6              7
12                     6             5                     4              6

    …  Fatigue  Weight Loss  Shortness of Breath  Wheezing  \
6   …        3            4                    2         2
7   …        3            2                    2         4
8   …        1            4                    3         2
```

```
11  …        9            6                   5         7
12  …        5            3                   2         4

    Swallowing Difficulty  Clubbing of Finger Nails  Frequent Cold  Dry Cough  \
6                       3                         1              2          3
7                       2                         2              3          4
8                       4                         6              2          4
11                      2                         4              3          1
12                      3                         1              7          5

    Snoring  Level
6         4      0
7         3      0
8         1      1
11        4      2
12        6      1

[5 rows x 25 columns]
```

[29]:
```
plt.hist(cancer_patient_male);
```



[30]:
```
plt.hist(cancer_patient_female);
```



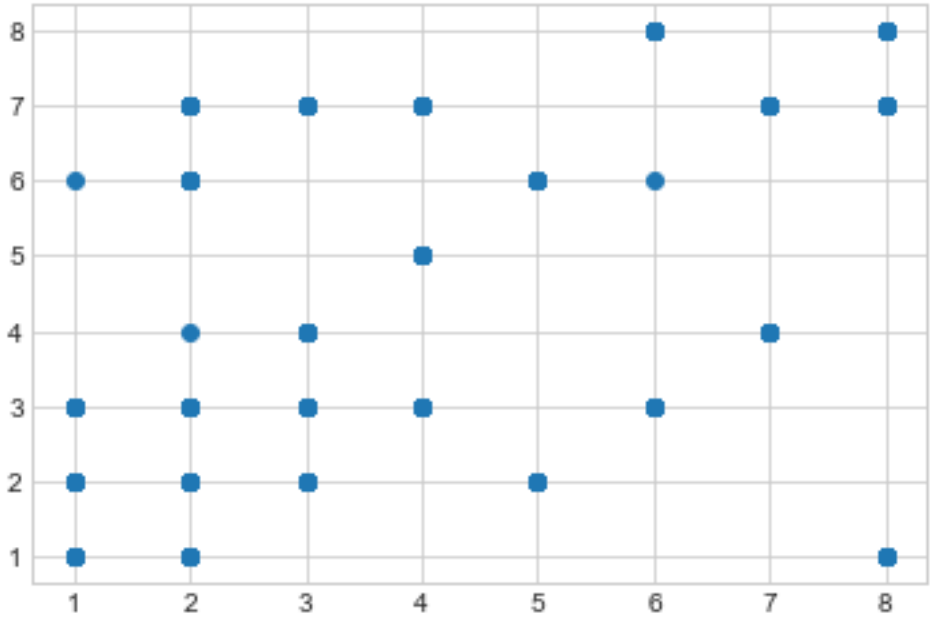
[31]:
```
cancer_patient_male.plot.hist(figsize = (15, 50), subplots = True);
```


Age
Gender
Air Pollution
Alcohol use
Dust Allergy
OccuPational Hazards
Genetic Risk
chronic Lung Disease
Balanced Diet
Obesity
Smoking
Passive Smoker
Chest Pain
Coughing of Blood
Fatigue
Weight Loss
Shortness of Breath
Wheezing
Swallowing Difficulty
Clubbing of Finger Nails
Frequent Cold
Dry Cough
Snoring
Level
Frequency

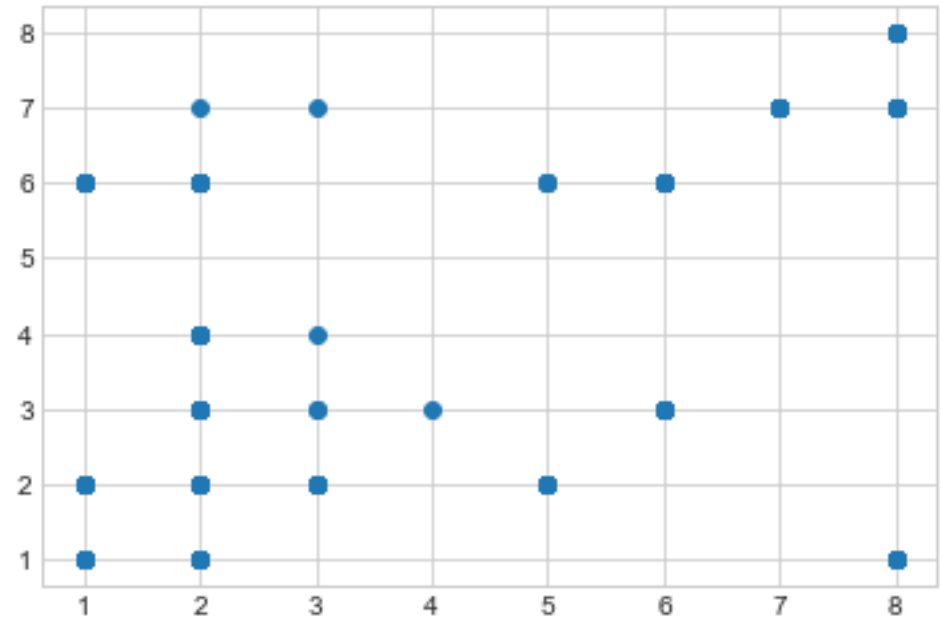
[32]:
```
cancer_patient_female.plot.hist(figsize = (15, 50), subplots = True);
```


Age
Gender
Air Pollution
Alcohol use
Dust Allergy
OccuPational Hazards
Genetic Risk
chronic Lung Disease
Balanced Diet
Obesity
Smoking
Passive Smoker
Chest Pain
Coughing of Blood
Fatigue
Weight Loss
Shortness of Breath
Wheezing
Swallowing Difficulty
Clubbing of Finger Nails
Frequent Cold
Dry Cough
Snoring
Level
Frequency

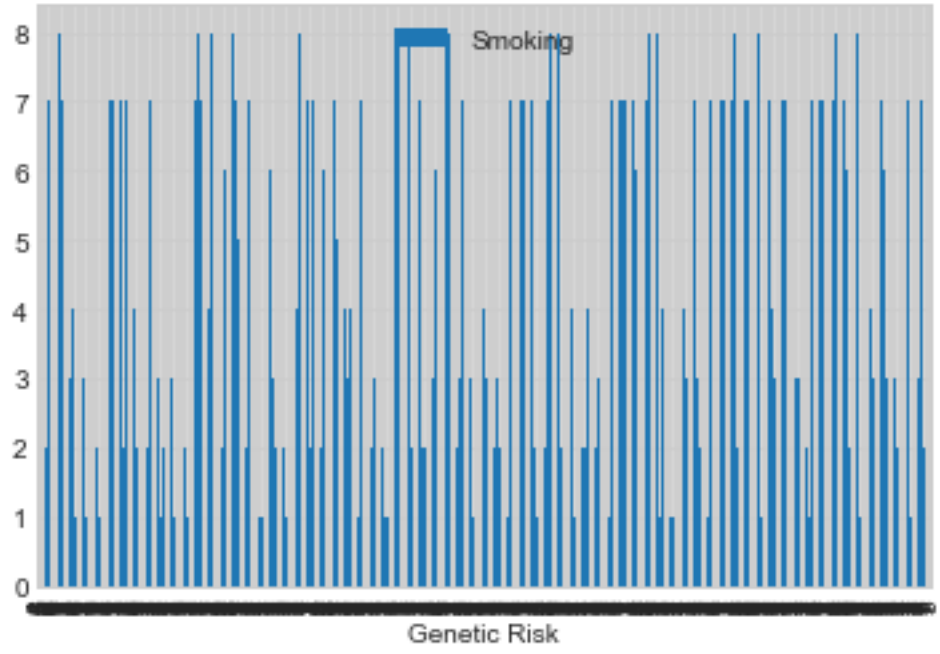
[33]:
```
fig, ax = plt.subplots()
scatter = ax.scatter(x = cancer_patient_male["Alcohol use"], y =␣
 ↪cancer_patient_male["Smoking"])
# cancer_patient_male.plot(x = cancer_patient_male["Alcohol use"], y =␣
 ↪cancer_patient_male["Age"], kind = "scatter");
```



[34]:
```
fig, ax = plt.subplots()
scatter = ax.scatter(x = cancer_patient_female["Alcohol use"], y =␣
 ↪cancer_patient_female["Smoking"]);
```



[35]:
```
fig, ax = plt.subplots()
cancer_patient_male.plot(kind = "bar", x = "Genetic Risk", y = "Smoking", ax =␣
↪ax);
```



[36]:
```
len(cancer_patient_male), len(cancer_patient_female)
```

[36]:
```
(598, 402)
```

[37]:
```
cancer_patient.head()
```

[37]:

	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	\
0	P1	33	1	2	4	5	
1	P10	17	1	3	1	5	
2	P100	35	1	4	5	6	
3	P1000	37	1	7	7	7	
4	P101	46	1	6	8	7	

	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	\
0	4	3	2	2	
1	3	4	2	2	
2	5	5	4	6	
3	7	6	7	7	
4	7	7	6	7	

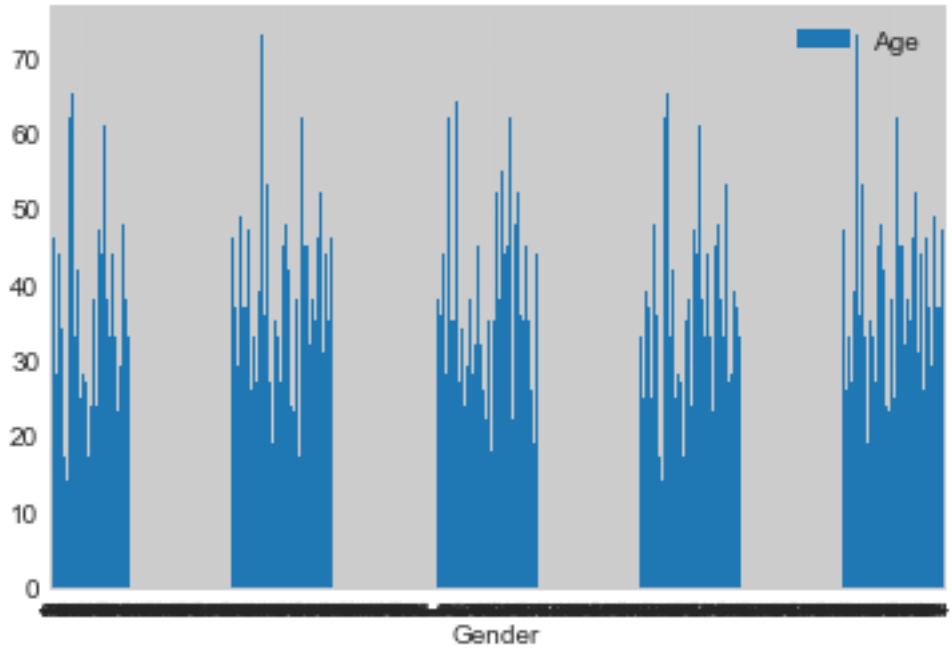
	…	Fatigue	Weight Loss	Shortness of Breath	Wheezing	\
0	…	3	4	2	2	
1	…	1	3	7	8	
2	…	8	7	9	2	
3	…	4	2	3	1	
4	…	3	2	4	1	

	Swallowing Difficulty	Clubbing of Finger Nails	Frequent Cold	Dry Cough	\
0	3	1	2	3	
1	6	2	1	7	
2	1	4	6	7	
3	4	5	6	7	
4	4	2	4	2	

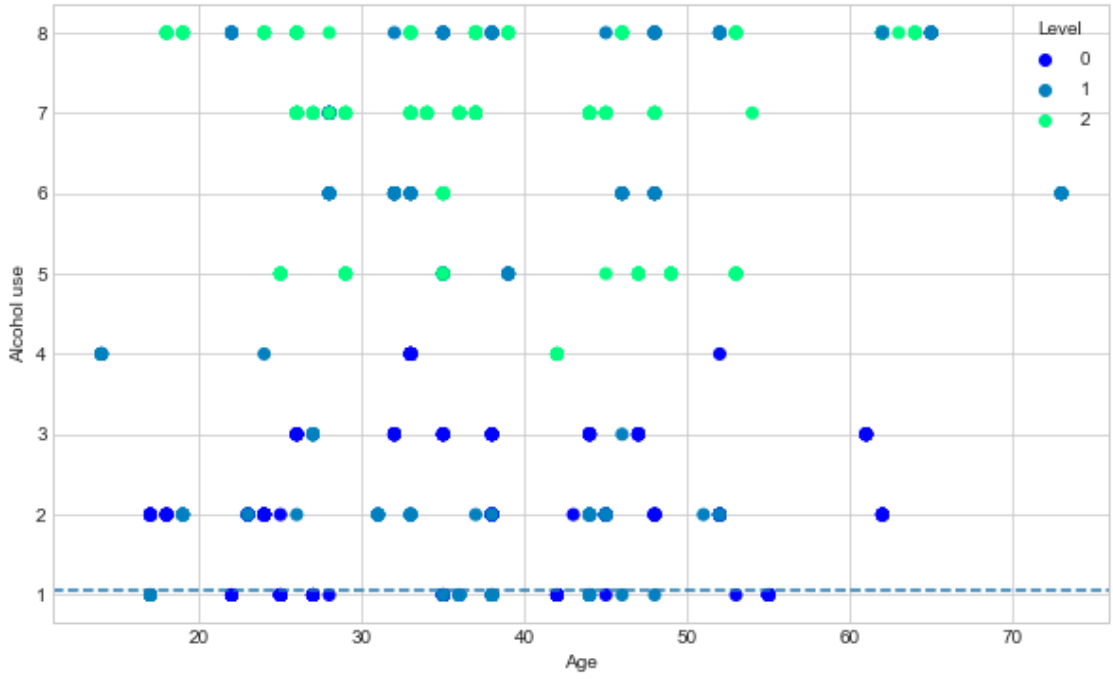
	Snoring	Level
0	4	0
1	2	1
2	2	2
3	5	2
4	3	2

[5 rows x 25 columns]

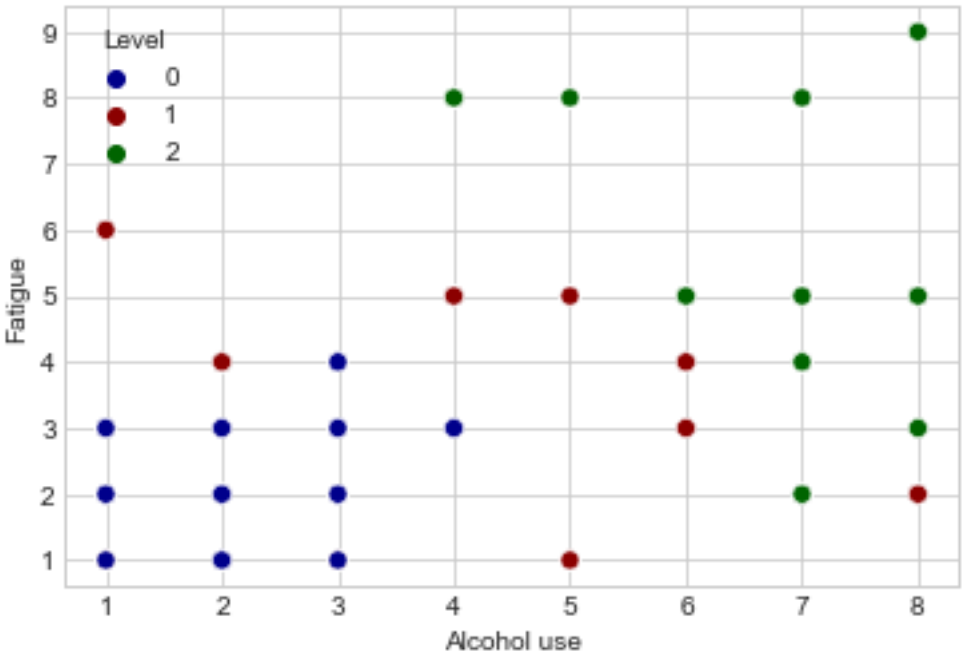
[38]:
```
fig, ax = plt.subplots()
cancer_patient.plot(kind = "bar", x = "Gender", y = "Age", ax = ax);
```



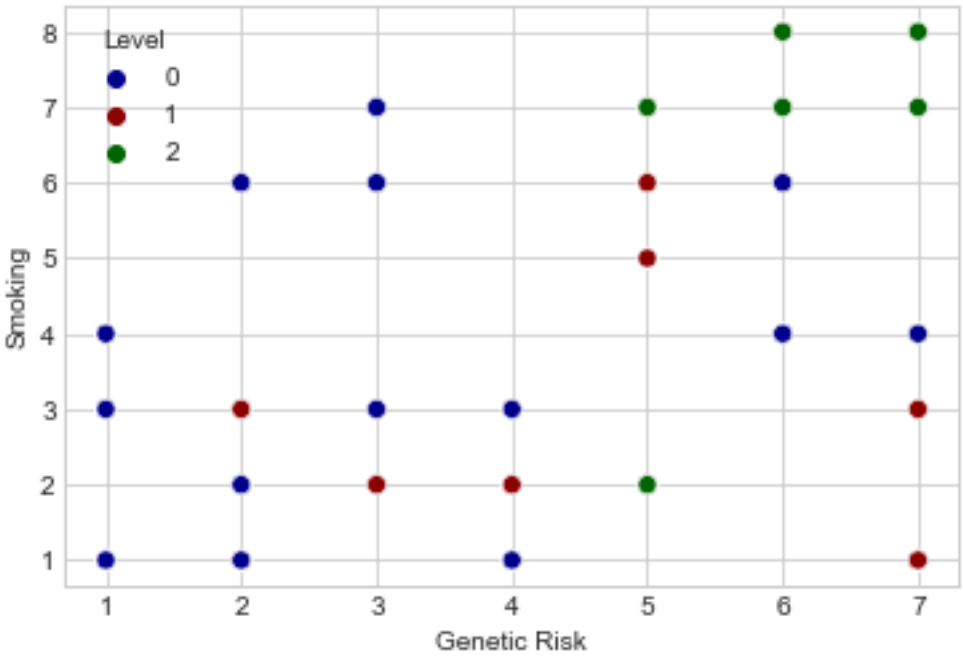
[39]:
```
fig, ax = plt.subplots(figsize = (10, 6))
scatter = ax.scatter(x = cancer_patient["Age"],
                     y = cancer_patient["Alcohol use"],
                     c = cancer_patient["Level"],
                     cmap = "winter")

ax.set(xlabel = "Age",
       ylabel = "Alcohol use");

ax.legend(*scatter.legend_elements(), title = "Level");

ax.axhline(cancer_patient["Level"].mean(),
           linestyle = "--");
```



[40]:
```
fig, ax=plt.subplots()#Required outside of function. This needs to be activated␣
 ↪first when plotting in every code block
plot=sns.scatterplot(data=cancer_patient,
                     x='Alcohol use',
                     y='Fatigue',
                     hue='Level',
                     palette=['darkblue','darkred','darkgreen'],
                     s=50,
                     marker='o')#Count plot
```



[41]:
```
fig, ax=plt.subplots()#Required outside of function. This needs to be activated␣
 ↪first when plotting in every code block
plot=sns.scatterplot(data=cancer_patient,
                     x='Genetic Risk',
                     y='Smoking',
                     hue='Level',
                     palette=['darkblue','darkred','darkgreen'],
                     s=50,
                     marker='o')#Count plot
```



Our data is analyzed and ready for Model Training and Machine Learning

[56]:
```
cancer_patient.head()
```

[56]:

	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	\
0	33	1	2	4	5	
1	17	1	3	1	5	
2	35	1	4	5	6	
3	37	1	7	7	7	
4	46	1	6	8	7	

	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	\
0	4	3	2	2	
1	3	4	2	2	
2	5	5	4	6	
3	7	6	7	7	
4	7	7	6	7	

	Obesity	…	Fatigue	Weight Loss	Shortness of Breath	Wheezing	\
0	4	…	3	4	2	2	
1	2	…	1	3	7	8	
2	7	…	8	7	9	2	
3	7	…	4	2	3	1	
4	7	…	3	2	4	1	

```
   Swallowing Difficulty  Clubbing of Finger Nails  Frequent Cold  Dry Cough  \
0                      3                         1              2          3
1                      6                         2              1          7
2                      1                         4              6          7
3                      4                         5              6          7
4                      4                         2              4          2

   Snoring  Level
0        4      0
1        2      1
2        2      2
3        5      2
4        3      2

[5 rows x 24 columns]
```

[43]:
```
cancer_patient.drop(["Patient Id"], axis = 1, inplace= True)
```

[44]:
```
cancer_patient.head()
```

[44]:
```
   Age  Gender  Air Pollution  Alcohol use  Dust Allergy  \
0   33       1              2            4             5
1   17       1              3            1             5
2   35       1              4            5             6
3   37       1              7            7             7
4   46       1              6            8             7

   OccuPational Hazards  Genetic Risk  chronic Lung Disease  Balanced Diet  \
0                     4             3                     2              2
1                     3             4                     2              2
2                     5             5                     4              6
3                     7             6                     7              7
4                     7             7                     6              7

   Obesity  …  Fatigue  Weight Loss  Shortness of Breath  Wheezing  \
0        4  …        3            4                    2         2
1        2  …        1            3                    7         8
2        7  …        8            7                    9         2
3        7  …        4            2                    3         1
4        7  …        3            2                    4         1

   Swallowing Difficulty  Clubbing of Finger Nails  Frequent Cold  Dry Cough  \
0                      3                         1              2          3
1                      6                         2              1          7
2                      1                         4              6          7
3                      4                         5              6          7
4                      4                         2              4          2
```

```
   Snoring  Level
0        4      0
1        2      1
2        2      2
3        5      2
4        3      2

[5 rows x 24 columns]
```

Fitting the model/algorithm and use it to make predictions on our data.

First we use Support Vector Machine Estimator

[45]:
```
from sklearn import svm
from sklearn.model_selection import train_test_split

X = cancer_patient.drop(["Level"], axis = 1)
y = cancer_patient["Level"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

sv = svm.SVC()
sv.fit(X_train, y_train)
sv.score(X_test, y_test)
```

[45]:
```
0.98
```

[46]:
```
y_preds = sv.predict(X_test)
y_preds[:10]
```

[46]:
```
array([2, 1, 2, 2, 2, 2, 0, 1, 2, 1])
```

[47]:
```
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score
print(classification_report(y_test, y_preds))
```

```
              precision    recall  f1-score   support

           0       1.00      0.95      0.97        61
           1       0.96      0.98      0.97        65
           2       0.99      1.00      0.99        74

    accuracy                           0.98       200
   macro avg       0.98      0.98      0.98       200
weighted avg       0.98      0.98      0.98       200
```

[48]:
```
confusion_matrix(y_test, y_preds)
```

[48]:
```
array([[58,  3,  0],
       [ 0, 64,  1],
       [ 0,  0, 74]], dtype=int64)
```

[49]:
```
accuracy_score(y_test, y_preds)
```

[49]:
```
0.98
```

Checking accuracy with other model

[50]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split

X = cancer_patient.drop(["Level"], axis = 1)
y = cancer_patient["Level"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
knn.score(X_test, y_test)
```

[50]:
```
1.0
```

RandomForestClassifier

[51]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split

X = cancer_patient.drop(["Level"], axis = 1)
y = cancer_patient["Level"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

rfr = RandomForestRegressor()
rfr.fit(X_train, y_train)
rfr.score(X_test, y_test)
```

[51]:
```
1.0
```

Cross val for all the above algorithms to make sure for scores accuracy

[52]:
```
from sklearn.model_selection import cross_val_score

crossVal_sv = cross_val_score(sv, X, y)
crossVal_knn = cross_val_score(knn, X, y)
```

```
crossVal_rfr = cross_val_score(rfr, X, y)

print(f"For SupportVectorMachine: {crossVal_sv}, \nFor KNeighborClassifier:␣
  ↪{crossVal_knn}, \nFor RandomForestRegressor: {crossVal_rfr}")
```

```
For SupportVectorMachine: [0.98  0.975 0.985 0.97  0.97 ],
For KNeighborClassifier: [0.995 1.    1.    1.    0.995],
For RandomForestRegressor: [0.9999937  1.         0.99999919 1.
0.99999852]
```

[53]:
```
# For SupportVectorMachine

np.random.seed(42)

sv_single_score = sv.score(X_test, y_test)

sv_cross_val_score = np.mean(cross_val_score(sv, X, y))

sv_single_score, sv_cross_val_score
```

[53]:
```
(0.985, 0.976)
```

[54]:
```
# For KNeighborClassifier

np.random.seed(42)

knn_single_score = knn.score(X_test, y_test)

knn_cross_val_score = np.mean(cross_val_score(knn, X, y))

knn_single_score, knn_cross_val_score
```

[54]:
```
(1.0, 0.998)
```