KEERTHI.7.PANDA

July 30, 2025

[24]:
```
import pandas as pd
```

[2]:
```
df=pd.read_csv("employee_survey_data.csv")
```

[25]:
```
df
```

[25]:

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
0	1	3.0	4.0	2.0
1	2	3.0	2.0	4.0
2	3	2.0	2.0	1.0
3	4	4.0	4.0	3.0
4	5	4.0	1.0	3.0
…	…	…	…	…
4405	4406	4.0	1.0	3.0
4406	4407	4.0	4.0	3.0
4407	4408	1.0	3.0	3.0
4408	4409	4.0	1.0	3.0
4409	4410	1.0	3.0	NaN

[4410 rows x 4 columns]

[26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   EmployeeID              4410 non-null   int64  
 1   EnvironmentSatisfaction  4385 non-null   float64
 2   JobSatisfaction         4390 non-null   float64
 3   WorkLifeBalance         4372 non-null   float64
dtypes: float64(3), int64(1)
memory usage: 137.9 KB
```

[27]:
```
df.head()
```

[27]:

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
0	1	3.0	4.0	2.0
1	2	3.0	2.0	4.0
2	3	2.0	2.0	1.0
3	4	4.0	4.0	3.0
4	5	4.0	1.0	3.0

[29]:
```
df.tail()
```

[29]:

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
4405	4406	4.0	1.0	3.0
4406	4407	4.0	4.0	3.0
4407	4408	1.0	3.0	3.0
4408	4409	4.0	1.0	3.0
4409	4410	1.0	3.0	NaN

[31]:
```
df.columns
```

[31]:
```
Index(['EmployeeID', 'EnvironmentSatisfaction', 'JobSatisfaction',
       'WorkLifeBalance'],
      dtype='object')
```

[32]:
```
df.isnull().sum()
```

[32]:
```
EmployeeID                 0
EnvironmentSatisfaction    25
JobSatisfaction            20
WorkLifeBalance            38
dtype: int64
```

[ ]:
```
import pandas as pd
```

[15]:
```
df=pd.read_csv("employee_survey_data.csv")
```

[73]:
```
df
```

[73]:

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
0	1	3.0	4.0	2.0
1	2	3.0	2.0	4.0
2	3	2.0	2.0	1.0
3	4	4.0	4.0	3.0
4	5	4.0	1.0	3.0
…	…	…	…	…
4405	4406	4.0	1.0	3.0
4406	4407	4.0	4.0	3.0
4407	4408	1.0	3.0	3.0
4408	4409	4.0	1.0	3.0

```
4409        4410                    1.0              3.0              NaN

[4410 rows x 4 columns]
```

[89]:
```
df["EnvironmentSatisfaction"]=df["EnvironmentSatisfaction"].
 ↪fillna(df["EnvironmentSatisfaction"].median())
```

[90]:
```
df.isnull().sum()
```

[90]:
```
EmployeeID                 0
EnvironmentSatisfaction     0
JobSatisfaction             0
WorkLifeBalance            38
dtype: int64
```

[85]:
```
df["JobSatisfaction"]=df["JobSatisfaction"].fillna(df["JobSatisfaction"].
 ↪median())
```

[86]:
```
df.isnull().sum()
```

[86]:
```
EmployeeID                 0
EnvironmentSatisfaction    25
JobSatisfaction             0
WorkLifeBalance            38
dtype: int64
```

[91]:
```
df["WorkLifeBalance"]=df["WorkLifeBalance"].fillna(df["WorkLifeBalance"].
 ↪median())
```

[92]:
```
df.isnull().sum()
```

[92]:
```
EmployeeID                 0
EnvironmentSatisfaction    0
JobSatisfaction            0
WorkLifeBalance            0
dtype: int64
```

[93]:
```
import pandas as pd
```

[94]:
```
df=pd.read_csv("employee_survey_data.csv")
```

[95]:
```
df
```

[95]:

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
0	1	3.0	4.0	2.0
1	2	3.0	2.0	4.0
2	3	2.0	2.0	1.0

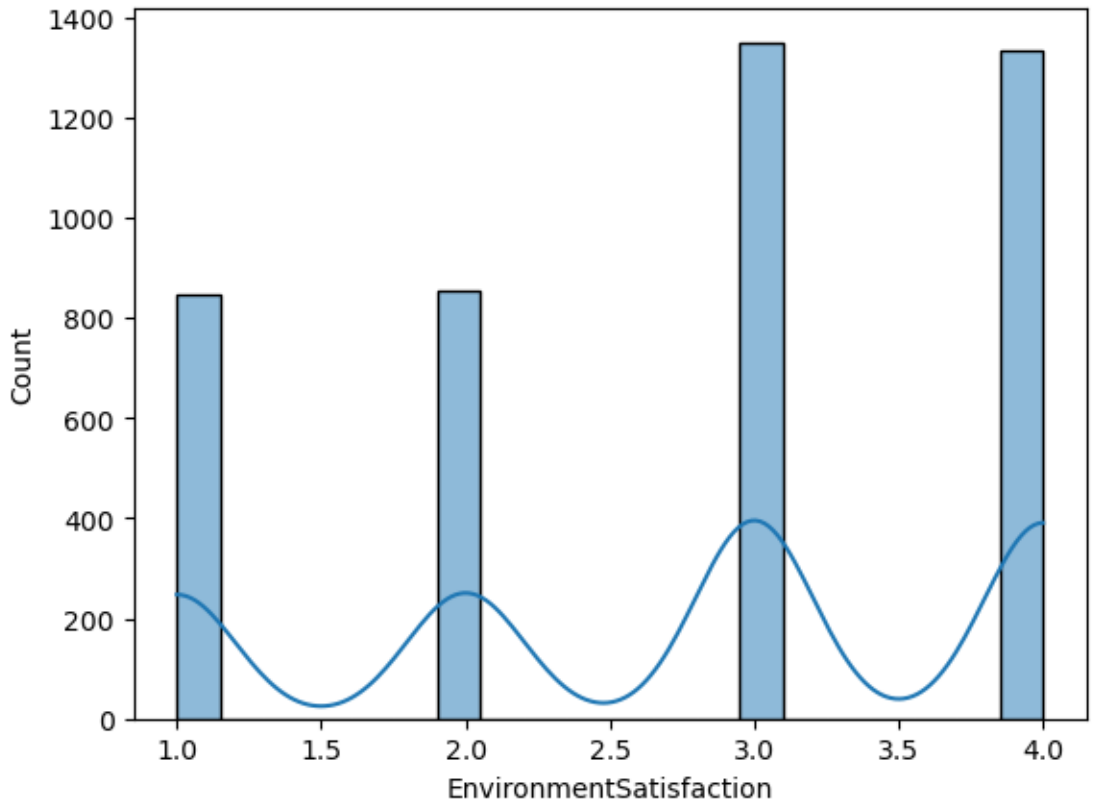
```
3              4                    4.0              4.0              3.0
4              5                    4.0              1.0              3.0
…              …                    …                …                …
4405        4406                    4.0              1.0              3.0
4406        4407                    4.0              4.0              3.0
4407        4408                    1.0              3.0              3.0
4408        4409                    4.0              1.0              3.0
4409        4410                    1.0              3.0              NaN

[4410 rows x 4 columns]
```

[96]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[97]:
```
numeric_cols=df.select_dtypes(include="number").columns
```

[99]:
```
sns.histplot(df["EnvironmentSatisfaction"],bins=20,kde=True)
```

[99]:
```
<Axes: xlabel='EnvironmentSatisfaction', ylabel='Count'>
```



[82]:
```
import pandas as pd
```

[83]:
```
df=pd.read_csv("employee_survey_data.csv")
```

[35]:
```
df
```

[35]:

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
0	1	3.0	4.0	2.0
1	2	3.0	2.0	4.0
2	3	2.0	2.0	1.0
3	4	4.0	4.0	3.0
4	5	4.0	1.0	3.0
…	…	…	…	…
4405	4406	4.0	1.0	3.0
4406	4407	4.0	4.0	3.0
4407	4408	1.0	3.0	3.0
4408	4409	4.0	1.0	3.0
4409	4410	1.0	3.0	NaN

[4410 rows x 4 columns]

[36]:
```
df.isnull().sum()
```

[36]:
```
EmployeeID                 0
EnvironmentSatisfaction    25
JobSatisfaction            20
WorkLifeBalance            38
dtype: int64
```

[37]:
```
df["JobSatisfaction"]=df["JobSatisfaction"].fillna(df["JobSatisfaction"].
↪median())
```

[38]:
```
df.isnull().sum()
```

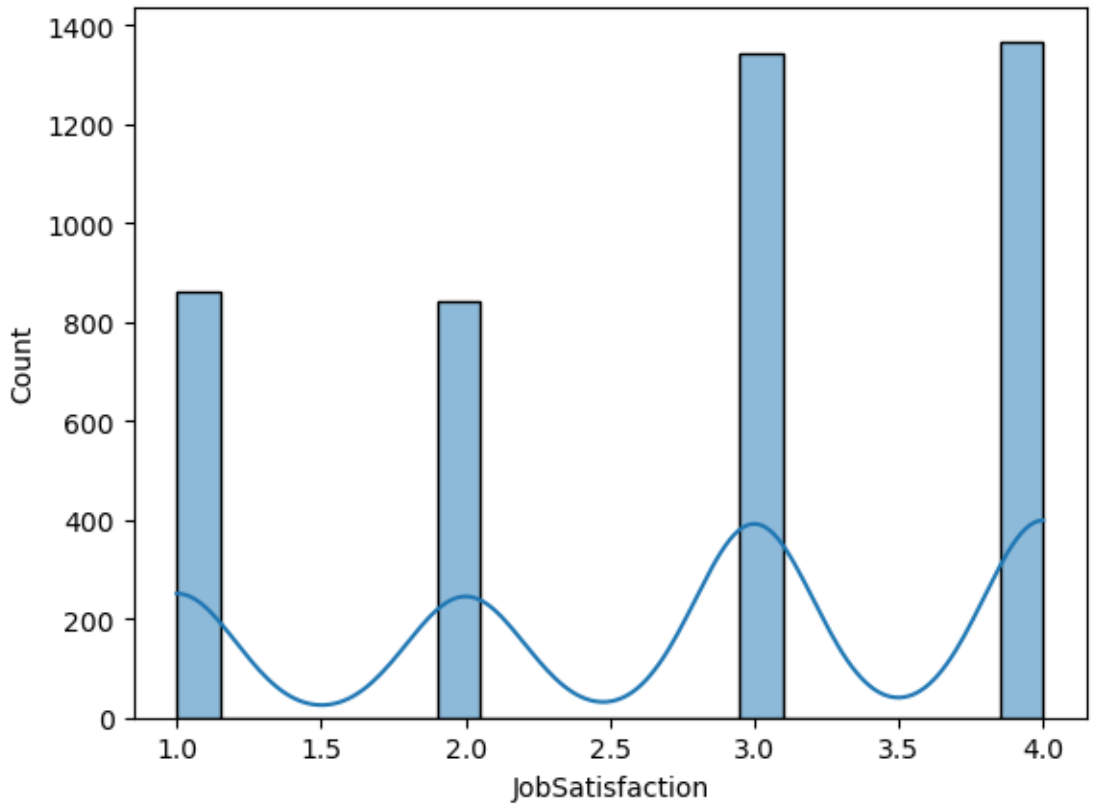
[38]:
```
EmployeeID                 0
EnvironmentSatisfaction    25
JobSatisfaction             0
WorkLifeBalance            38
dtype: int64
```

[39]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[40]:
```
numeric_cols=df.select_dtypes(include="number").columns
```

[42]:
```
sns.histplot(df["JobSatisfaction"],bins=20,kde=True)
```

[42]: <Axes: xlabel='JobSatisfaction', ylabel='Count'>



[ ]:
```
import pandas as pd
```

[46]:
```
df=pd.read_csv("car_data.csv")
```

[47]:
```
df
```

[47]:

	User ID	Gender	Age	AnnualSalary	Purchased
0	385	Male	35	20000	0
1	681	Male	40	43500	0
2	353	Male	49	74000	0
3	895	Male	40	107500	1
4	661	Male	25	79000	0
..	…	…	…	…	…
995	863	Male	38	59000	0
996	800	Female	47	23500	0
997	407	Female	28	138500	1
998	299	Female	48	134000	1
999	687	Female	44	73500	0

```
[1000 rows x 5 columns]
```

[54]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   User ID       1000 non-null   int64
 1   Gender        1000 non-null   object
 2   Age           1000 non-null   int64
 3   AnnualSalary  1000 non-null   int64
 4   Purchased     1000 non-null   int64
dtypes: int64(4), object(1)
memory usage: 39.2+ KB
```

[48]:
```
df.head()
```

[48]:

	User ID	Gender	Age	AnnualSalary	Purchased
0	385	Male	35	20000	0
1	681	Male	40	43500	0
2	353	Male	49	74000	0
3	895	Male	40	107500	1
4	661	Male	25	79000	0

[49]:
```
df.tail()
```

[49]:

	User ID	Gender	Age	AnnualSalary	Purchased
995	863	Male	38	59000	0
996	800	Female	47	23500	0
997	407	Female	28	138500	1
998	299	Female	48	134000	1
999	687	Female	44	73500	0

[57]:
```
df.columns
```

[57]:
```
Index(['User ID', 'Gender', 'Age', 'AnnualSalary', 'Purchased'], dtype='object')
```

[59]:
```
df
```

[59]:

	User ID	Gender	Age	AnnualSalary	Purchased
0	385	Male	35	20000	0
1	681	Male	40	43500	0
2	353	Male	49	74000	0
3	895	Male	40	107500	1
4	661	Male	25	79000	0

```
..       …     …  …           …        …
995      863    Male   38         59000          0
996      800  Female   47         23500          0
997      407  Female   28        138500          1
998      299  Female   48        134000          1
999      687  Female   44         73500          0

[1000 rows x 5 columns]
```

[60]:
```
df.isnull().sum()
```

[60]:
```
User ID         0
Gender          0
Age             0
AnnualSalary    0
Purchased       0
dtype: int64
```

[83]:
```
import pandas as pd
```

[ ]:
```
df=pd.read_csv("KNNAlgorithmDataset.csv")
```

[84]:
```
df
```

[84]:
```
           id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0      842302         M        17.99         10.38          122.80     1001.0
1      842517         M        20.57         17.77          132.90     1326.0
2    84300903         M        19.69         21.25          130.00     1203.0
3    84348301         M        11.42         20.38           77.58      386.1
4    84358402         M        20.29         14.34          135.10     1297.0
..        …       …           …             …               …        …
564    926424         M        21.56         22.39          142.00     1479.0
565    926682         M        20.13         28.25          131.20     1261.0
566    926954         M        16.60         28.08          108.30      858.1
567    927241         M        20.60         29.33          140.10     1265.0
568     92751         B         7.76         24.54           47.92      181.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0            0.11840           0.27760         0.30010              0.14710
1            0.08474           0.07864         0.08690              0.07017
2            0.10960           0.15990         0.19740              0.12790
3            0.14250           0.28390         0.24140              0.10520
4            0.10030           0.13280         0.19800              0.10430
..               …                 …               …                    …
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
```

```
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     …  radius_worst  texture_worst  perimeter_worst  area_worst  \
0    …        25.380          17.33           184.60      2019.0
1    …        24.990          23.41           158.80      1956.0
2    …        23.570          25.53           152.50      1709.0
3    …        14.910          26.50            98.87       567.7
4    …        22.540          16.67           152.20      1575.0
..   …           …              …                …          …
564  …        25.450          26.40           166.10      2027.0
565  …        23.690          38.25           155.00      1731.0
566  …        18.980          34.12           126.70      1124.0
567  …        25.740          39.42           184.60      1821.0
568  …         9.456          30.37            59.16       268.6

     smoothness_worst  compactness_worst  concavity_worst  \
0             0.16220            0.66560           0.7119
1             0.12380            0.18660           0.2416
2             0.14440            0.42450           0.4504
3             0.20980            0.86630           0.6869
4             0.13740            0.20500           0.4000
..                …                  …                …
564           0.14100            0.21130           0.4107
565           0.11660            0.19220           0.3215
566           0.11390            0.30940           0.3403
567           0.16500            0.86810           0.9387
568           0.08996            0.06444           0.0000

     concave points_worst  symmetry_worst  fractal_dimension_worst
0                  0.2654          0.4601                  0.11890
1                  0.1860          0.2750                  0.08902
2                  0.2430          0.3613                  0.08758
3                  0.2575          0.6638                  0.17300
4                  0.1625          0.2364                  0.07678
..                    …               …                       …
564                0.2216          0.2060                  0.07115
565                0.1628          0.2572                  0.06637
566                0.1418          0.2218                  0.07820
567                0.2650          0.4087                  0.12400
568                0.0000          0.2871                  0.07039

[569 rows x 32 columns]
```

[85]:
```
df.head()
```

[85]:
```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   …  radius_worst  texture_worst  perimeter_worst  area_worst  \
0  …         25.38          17.33           184.60      2019.0
1  …         24.99          23.41           158.80      1956.0
2  …         23.57          25.53           152.50      1709.0
3  …         14.91          26.50            98.87       567.7
4  …         22.54          16.67           152.20      1575.0

   smoothness_worst  compactness_worst  concavity_worst  concave points_worst  \
0            0.1622             0.6656           0.7119                0.2654
1            0.1238             0.1866           0.2416                0.1860
2            0.1444             0.4245           0.4504                0.2430
3            0.2098             0.8663           0.6869                0.2575
4            0.1374             0.2050           0.4000                0.1625

   symmetry_worst  fractal_dimension_worst
0          0.4601                  0.11890
1          0.2750                  0.08902
2          0.3613                  0.08758
3          0.6638                  0.17300
4          0.2364                  0.07678

[5 rows x 32 columns]
```

[86]:
```
df.tail()
```

[86]:
```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
564  926424         M        21.56         22.39          142.00     1479.0
565  926682         M        20.13         28.25          131.20     1261.0
566  926954         M        16.60         28.08          108.30      858.1
567  927241         M        20.60         29.33          140.10     1265.0
568   92751         B         7.76         24.54           47.92      181.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
```

```
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     …  radius_worst  texture_worst  perimeter_worst  area_worst  \
564  …        25.450          26.40           166.10      2027.0
565  …        23.690          38.25           155.00      1731.0
566  …        18.980          34.12           126.70      1124.0
567  …        25.740          39.42           184.60      1821.0
568  …         9.456          30.37            59.16       268.6

     smoothness_worst  compactness_worst  concavity_worst  \
564           0.14100            0.21130           0.4107
565           0.11660            0.19220           0.3215
566           0.11390            0.30940           0.3403
567           0.16500            0.86810           0.9387
568           0.08996            0.06444           0.0000

     concave points_worst  symmetry_worst  fractal_dimension_worst
564                0.2216          0.2060                  0.07115
565                0.1628          0.2572                  0.06637
566                0.1418          0.2218                  0.07820
567                0.2650          0.4087                  0.12400
568                0.0000          0.2871                  0.07039

[5 rows x 32 columns]
```

[87]:
```
df.columns
```

[87]:
```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst'],
      dtype='object')
```

[88]:
```
df.isnull().sum()
```

[88]:
```
id                         0
diagnosis                  0
radius_mean                0
```

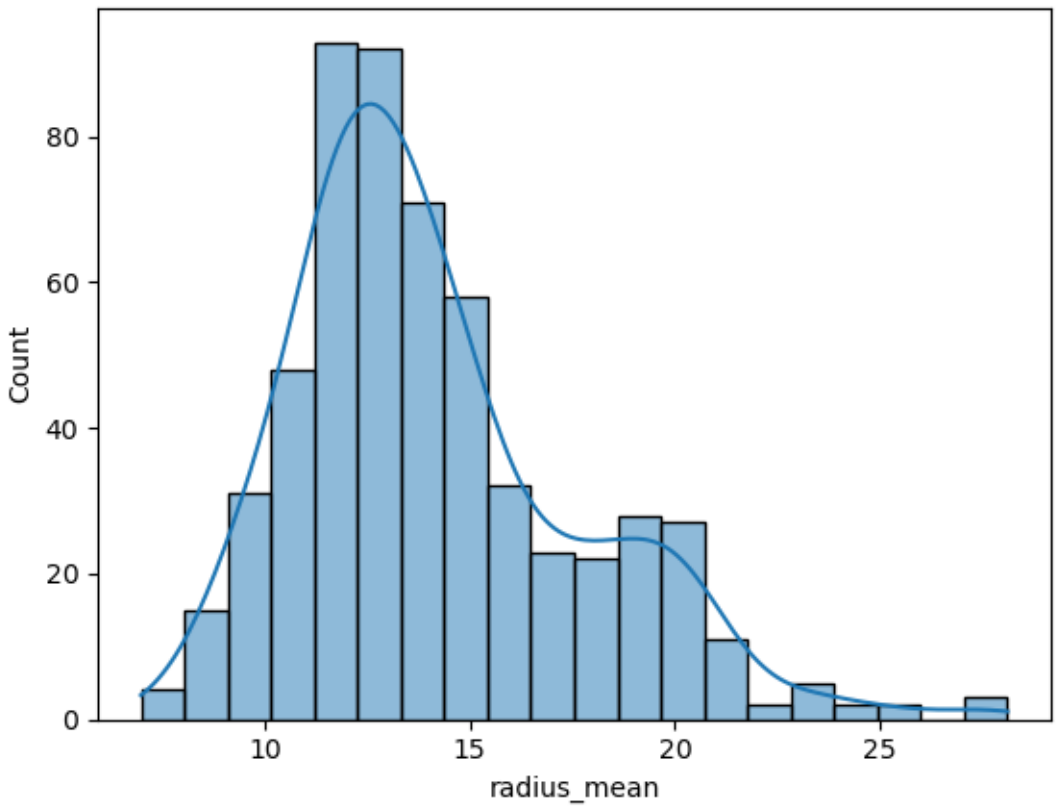
```
texture_mean              0
perimeter_mean            0
area_mean                 0
smoothness_mean           0
compactness_mean          0
concavity_mean            0
concave points_mean       0
symmetry_mean             0
fractal_dimension_mean    0
radius_se                 0
texture_se                0
perimeter_se              0
area_se                   0
smoothness_se             0
compactness_se            0
concavity_se              0
concave points_se         0
symmetry_se               0
fractal_dimension_se      0
radius_worst              0
texture_worst             0
perimeter_worst           0
area_worst                0
smoothness_worst          0
compactness_worst         0
concavity_worst           0
concave points_worst      0
symmetry_worst            0
fractal_dimension_worst   0
dtype: int64
```

[89]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

[90]:
```
numeric_cols=df.select_dtypes(include="number").columns
```

[91]:
```
sns.histplot(df["radius_mean"],bins=20,kde=True)
```

[91]:
```
<Axes: xlabel='radius_mean', ylabel='Count'>
```



[92]:
```
columns=["radius_mean","texture_mean","perimeter_mean","area_mean","smoothness_mean","compactn
  ↪points_mean","symmetry_mean","fractal_dimension_mean","radius_se","texture_se","perimeter_s
  ↪points_se","symmetry_se","fractal_dimension_se"]
for col in columns:
    Q1=df[col].quantile(0.25)
    Q3=df[col].quantile(0.75)
    IQR=Q3-Q1

    LOWER_BOUND=Q1-1.5*IQR
    UPPER_BOUND=Q3+1.5*IQR
    outliers = df[(df[col] < LOWER_BOUND)|(df[col] > UPPER_BOUND)]
    print(outliers.shape[0])
```

```
14
7
13
25
6
16
18
10
```

```
15
15
38
20
38
65
6
28
22
19
27
28
```

[93]:
```
df.to_csv("algorithm_mean.csv",index=False)
```

[ ]:
```
dff=pd.read_csv("algorithm_mean.csv")
```