

Lets Grow More (VIP)

Beginner level Task 01

Iris Flowers Classification Project

Name- Priyanka Jagdale

```
In [6]: import pandas as pd
import numpy as np
import matplotlib.pyplot
import seaborn as sns
import os
```

```
In [7]: os.getcwd()
```

```
Out[7]: 'C:\\Users\\KANIFNATH'
```

```
In [9]: os.chdir("F:\\Lets grow more")
```

```
In [22]: iris = pd.read_csv(r"C:\Users\KANIFNATH\AppData\Local\Temp\Temp1_archive (1).zip\Iris.csv")
```

```
In [24]: iris.shape
```

```
Out[24]: (150, 6)
```

```
In [30]: iris.head()
```

```
Out[30]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
In [33]: iris.mean()
```

```
Out[33]: Id                75.500000  
SepalLengthCm          5.843333  
SepalWidthCm           3.054000  
PetalLengthCm          3.758667  
PetalWidthCm           1.198667  
dtype: float64
```

In [34]: `iris.mode()`

Out[34]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.0	3.0	1.5	0.2	Iris-setosa
1	2	NaN	NaN	NaN	NaN	Iris-versicolor
2	3	NaN	NaN	NaN	NaN	Iris-virginica
3	4	NaN	NaN	NaN	NaN	NaN
4	5	NaN	NaN	NaN	NaN	NaN
5	6	NaN	NaN	NaN	NaN	NaN
6	7	NaN	NaN	NaN	NaN	NaN
7	8	NaN	NaN	NaN	NaN	NaN
8	9	NaN	NaN	NaN	NaN	NaN
9	10	NaN	NaN	NaN	NaN	NaN
10	11	NaN	NaN	NaN	NaN	NaN
11	12	NaN	NaN	NaN	NaN	NaN
12	13	NaN	NaN	NaN	NaN	NaN
13	14	NaN	NaN	NaN	NaN	NaN
14	15	NaN	NaN	NaN	NaN	NaN
15	16	NaN	NaN	NaN	NaN	NaN
16	17	NaN	NaN	NaN	NaN	NaN
17	18	NaN	NaN	NaN	NaN	NaN
18	19	NaN	NaN	NaN	NaN	NaN
19	20	NaN	NaN	NaN	NaN	NaN
20	21	NaN	NaN	NaN	NaN	NaN
21	22	NaN	NaN	NaN	NaN	NaN
22	23	NaN	NaN	NaN	NaN	NaN
23	24	NaN	NaN	NaN	NaN	NaN
24	25	NaN	NaN	NaN	NaN	NaN
25	26	NaN	NaN	NaN	NaN	NaN
26	27	NaN	NaN	NaN	NaN	NaN
27	28	NaN	NaN	NaN	NaN	NaN
28	29	NaN	NaN	NaN	NaN	NaN
29	30	NaN	NaN	NaN	NaN	NaN
...
120	121	NaN	NaN	NaN	NaN	NaN
121	122	NaN	NaN	NaN	NaN	NaN
122	123	NaN	NaN	NaN	NaN	NaN
123	124	NaN	NaN	NaN	NaN	NaN

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
124	125	NaN	NaN	NaN	NaN	NaN
125	126	NaN	NaN	NaN	NaN	NaN
126	127	NaN	NaN	NaN	NaN	NaN
127	128	NaN	NaN	NaN	NaN	NaN
128	129	NaN	NaN	NaN	NaN	NaN
129	130	NaN	NaN	NaN	NaN	NaN
130	131	NaN	NaN	NaN	NaN	NaN
131	132	NaN	NaN	NaN	NaN	NaN
132	133	NaN	NaN	NaN	NaN	NaN
133	134	NaN	NaN	NaN	NaN	NaN
134	135	NaN	NaN	NaN	NaN	NaN
135	136	NaN	NaN	NaN	NaN	NaN
136	137	NaN	NaN	NaN	NaN	NaN
137	138	NaN	NaN	NaN	NaN	NaN
138	139	NaN	NaN	NaN	NaN	NaN
139	140	NaN	NaN	NaN	NaN	NaN
140	141	NaN	NaN	NaN	NaN	NaN
141	142	NaN	NaN	NaN	NaN	NaN
142	143	NaN	NaN	NaN	NaN	NaN
143	144	NaN	NaN	NaN	NaN	NaN
144	145	NaN	NaN	NaN	NaN	NaN
145	146	NaN	NaN	NaN	NaN	NaN
146	147	NaN	NaN	NaN	NaN	NaN
147	148	NaN	NaN	NaN	NaN	NaN
148	149	NaN	NaN	NaN	NaN	NaN
149	150	NaN	NaN	NaN	NaN	NaN

150 rows × 6 columns

In [35]: iris.median()

Out[35]: Id 75.50
 SepalLengthCm 5.80
 SepalWidthCm 3.00
 PetalLengthCm 4.35
 PetalWidthCm 1.30
 dtype: float64

```
In [36]: iris.std()
```

```
Out[36]: Id                43.445368
SepalLengthCm      0.828066
SepalWidthCm       0.433594
PetalLengthCm      1.764420
PetalWidthCm       0.763161
dtype: float64
```

```
In [37]: iris.min()
```

```
Out[37]: Id                1
SepalLengthCm      4.3
SepalWidthCm       2
PetalLengthCm      1
PetalWidthCm       0.1
Species            Iris-setosa
dtype: object
```

```
In [38]: iris.max()
```

```
Out[38]: Id                150
SepalLengthCm      7.9
SepalWidthCm       4.4
PetalLengthCm      6.9
PetalWidthCm       2.5
Species            Iris-virginica
dtype: object
```

```
In [39]: iris.count()
```

```
Out[39]: Id                150
SepalLengthCm      150
SepalWidthCm       150
PetalLengthCm      150
PetalWidthCm       150
Species            150
dtype: int64
```

```
In [41]: iris.tail()
```

```
Out[41]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
In [42]: print("Target Labels", iris["Species"].unique())
```

```
Target Labels ['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
```

Training and Testing Model

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

```
In [46]: x=iris.drop(columns=['Species'])  
y=iris['Species']  
x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.2)
```

```
In [47]: from sklearn.linear_model import LogisticRegression
```


F:\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:35: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

eps=np.finfo(np.float).eps,

F:\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:597: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,

F:\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:836: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,

F:\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:862: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

eps=np.finfo(np.float).eps, positive=False):

F:\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1097: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,

F:\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1344: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,

F:\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1480: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```

    eps=np.finfo(np.float).eps, copy_X=True, positive=False):
F:\Anaconda3\lib\site-packages\sklearn\linear_model\ransomized_l1.py:152: Dep
recationWarning: `np.float` is a deprecated alias for the builtin `float`. To
silence this warning, use `float` by itself. Doing this will not modify any b
ehavior and is safe. If you specifically wanted the numpy scalar type, use `n
p.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
    precompute=False, eps=np.finfo(np.float).eps,
F:\Anaconda3\lib\site-packages\sklearn\linear_model\ransomized_l1.py:320: Dep
recationWarning: `np.float` is a deprecated alias for the builtin `float`. To
silence this warning, use `float` by itself. Doing this will not modify any b
ehavior and is safe. If you specifically wanted the numpy scalar type, use `n
p.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
    eps=np.finfo(np.float).eps, random_state=None,
F:\Anaconda3\lib\site-packages\sklearn\linear_model\ransomized_l1.py:580: Dep
recationWarning: `np.float` is a deprecated alias for the builtin `float`. To
silence this warning, use `float` by itself. Doing this will not modify any b
ehavior and is safe. If you specifically wanted the numpy scalar type, use `n
p.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/de
vdocs/release/1.20.0-notes.html#deprecations
    eps=4 * np.finfo(np.float).eps, n_jobs=None,

```

```
In [48]: log_reg=LogisticRegression()
```

```
In [49]: log_reg.fit(x_train,y_train)
```

```

F:\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWa
rning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to
silence this warning.
    FutureWarning)
F:\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:460: FutureWa
rning: Default multi_class will be changed to 'auto' in 0.22. Specify the mul
ti_class option to silence this warning.
    "this warning.", FutureWarning)

```

```

Out[49]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
    intercept_scaling=1, max_iter=100, multi_class='warn',
    n_jobs=None, penalty='l2', random_state=None, solver='warn',
    tol=0.0001, verbose=0, warm_start=False)

```

```
In [50]: log_reg.score(x_test,y_test)
```

```
Out[50]: 0.9333333333333333
```

```
In [51]: log_reg.score(x,y)
```

```
Out[51]: 0.9333333333333333
```

```
In [ ]:
```