lujv8qfpo

February 7, 2025

1 Machine Learning Part 4

[141]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from mlxtend.plotting import plot_decision_regions
from sklearn.preprocessing import PolynomialFeatures
from sklearn.datasets import load_iris
from sklearn.metrics import␣
 ↪confusion_matrix,precision_score,recall_score,f1_score
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
from mlxtend.plotting import plot_decision_regions
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
```

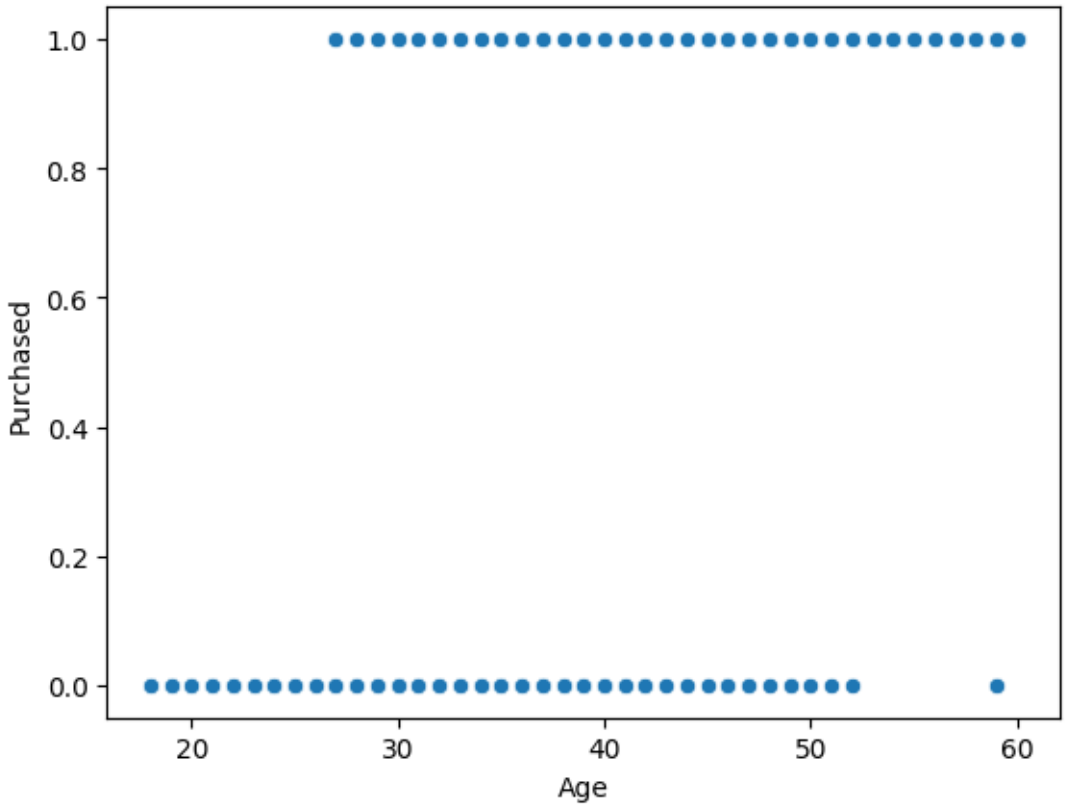
[7]:
```
dataset = pd.read_csv("/Users/ratnadeepgurav/Desktop/AIDS/Study for carrer␣
 ↪material/WSCUBE_Data_Analyst/ML/dataset/Social_Network_Ads.csv")
dataset.drop(columns= ["User ID","Gender","EstimatedSalary"],inplace = True)
dataset.head(3)
```

[7]:

	Age	Purchased
0	19	0
1	35	0
2	26	0

2 Logistic Regression (single Input)

[12]:
```
sns.scatterplot(x = "Age" ,y = "Purchased", data = dataset )
plt.show()
```

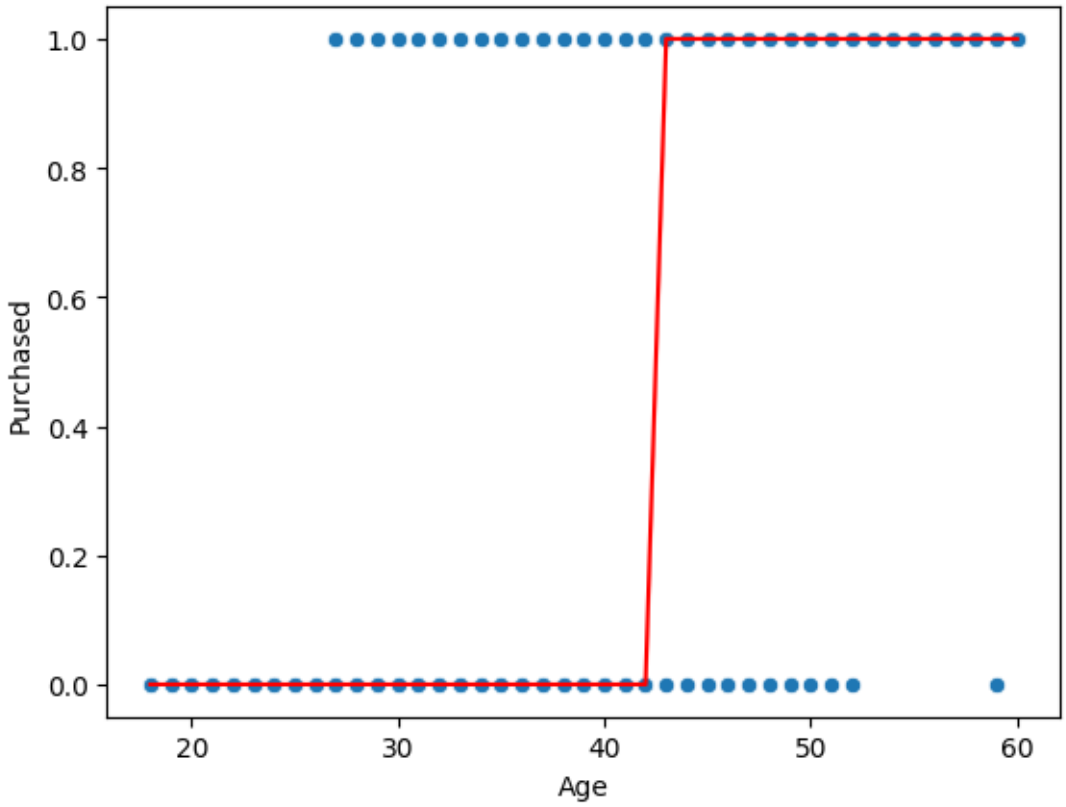


[26]:
```
x = data[["Age"]]
y = data["Purchased"]

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2 ,␣
 ↪random_state = 42)

lr =LogisticRegression()

lr.fit(x_train,y_train)

print(lr.score(x_test,y_test)*100)

print(lr.predict([[42]]))
```

```
91.25
[0]
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

[30]:
```
sns.scatterplot(x = "Age" ,y = "Purchased", data = dataset )
sns.lineplot(x = "Age" , y =lr.predict(x),data = dataset,color = "red")
plt.show()
```



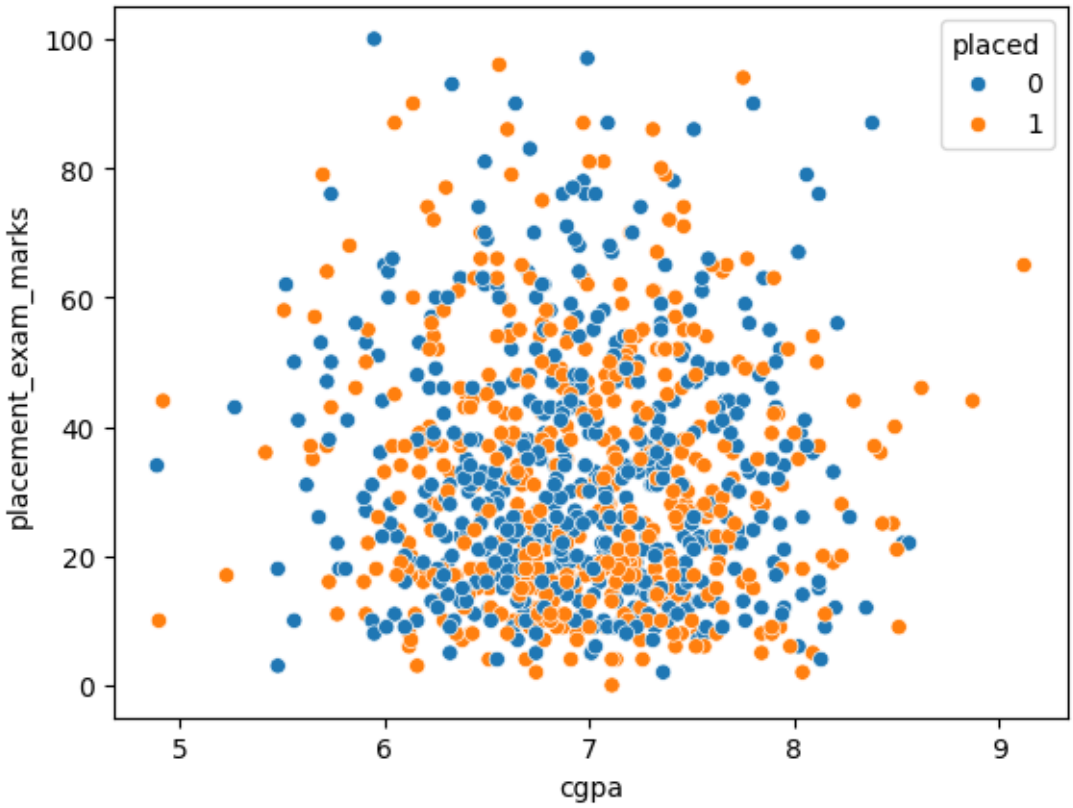
3 Logistic Regression (Multiple Input)

[40]:
```
data1 = pd.read_csv("/Users/ratnadeepgurav/Desktop/AIDS/Study for carrer␣
↪material/WSCUBE_Data_Analyst/ML/dataset/placement 3.csv")
data1.head(3)
```

[40]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1

[42]:
```
sns.scatterplot(x = "cgpa", y = "placement_exam_marks" , data = data1 , hue =␣
↪"placed")
plt.show()
```



[76]:
```
x1 = data1.iloc[:,:-1]
y1 = data1["placement_exam_marks"]


x_train,x_test,y_train,y_test = train_test_split(x1,y1,test_size = 0.2 ,␣
 ↪random_state = 42)

lr = LogisticRegression()

lr.fit(x_train,y_train)

lr.score(x_test,y_test)*100

lr.predict([[7.54,7.46]])
```

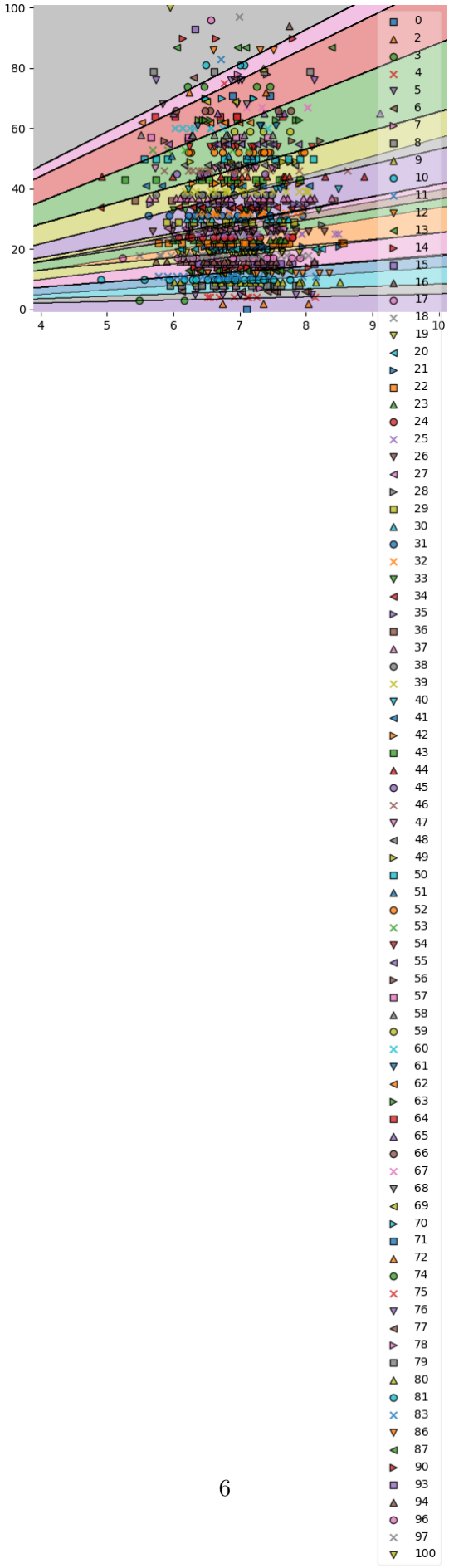
```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

[76]:
```
array([9.])
```

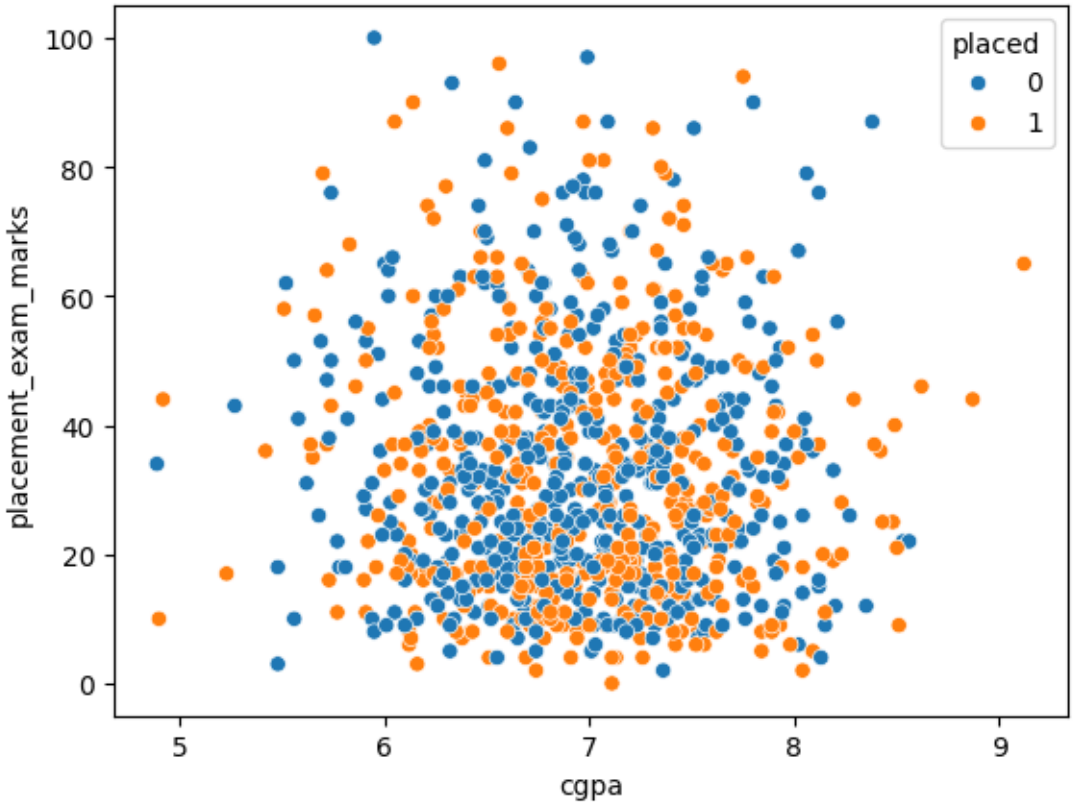
[84]:
```
plot_decision_regions(x1.to_numpy(), y1.to_numpy().astype(np.int_),clf = lr)
plt.show()
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/mlxtend/plotting/decision_regions.py:346: UserWarning: You passed a
edgecolor/edgecolors ('black') for an unfilled marker ('x').  Matplotlib is
ignoring the edgecolor in favor of the facecolor.  This behavior may change in
the future.
  ax.scatter(
```


100
80
60
40
20
0
4
5
6
7
8
9
10
0
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
74
75
76
77
78
79
80
81
83
86
87
90
93
94
96
97
100

4 Logical Regression (Polynomial Regression) :- used where data are not in linear form

[86]:
```
sns.scatterplot(x = "cgpa", y = "placement_exam_marks" , data = data1 , hue =␣
↪"placed")
plt.show()
```



[88]:
```
x1 = data1.iloc[:,:-1]
y1 = data1["placement_exam_marks"]


x_train,x_test,y_train,y_test = train_test_split(x1,y1,test_size = 0.2 ,␣
↪random_state = 42)

lr = LogisticRegression()

lr.fit(x_train,y_train)
```

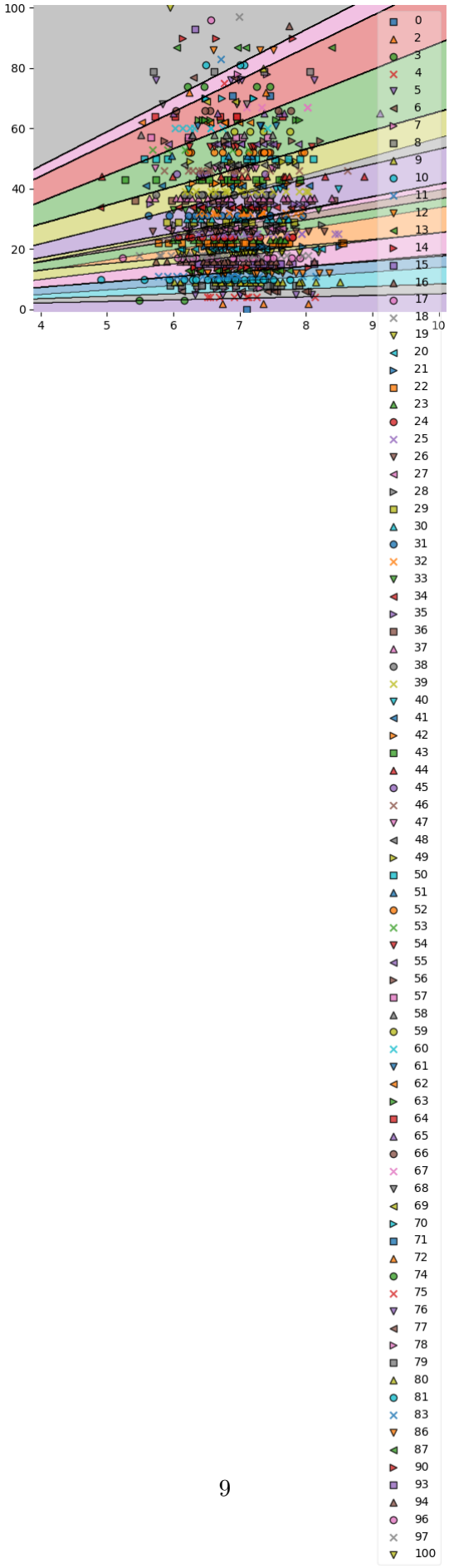
```
lr.score(x_test,y_test)*100

lr.predict([[7.54,7.46]])
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

[88]:
```
array([9.])
```

[93]:
```
plot_decision_regions(x1.to_numpy(), y1.to_numpy().astype(np.int_),clf = lr)
plt.show()
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/mlxtend/plotting/decision_regions.py:346: UserWarning: You passed a
edgecolor/edgecolors ('black') for an unfilled marker ('x').  Matplotlib is
ignoring the edgecolor in favor of the facecolor.  This behavior may change in
the future.
  ax.scatter(
```


100
80
60
40
20
0
4
5
6
7
8
9
10
0
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
74
75
76
77
78
79
80
81
83
86
87
90
93
94
96
97
100

[104]:
```
pf = PolynomialFeatures(degree =3)

pf.fit(x1)
df = pd.DataFrame(pf.transform(x1))

x_train,x_test,y_train,y_test = train_test_split(x1,y1,test_size = 0.2 ,␣
 ↪random_state = 42)

lr = LogisticRegression()

lr.fit(x_train,y_train)

lr.score(x_test,y_test)*100
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[104]:
```
14.000000000000002
```

5 Logistic Regression (Multiclass Classification)

[3]:
```
data3 = pd.read_csv("/Users/ratnadeepgurav/Desktop/AIDS/Study for carrer␣
 ↪material/WSCUBE_Data_Analyst/ML/dataset/Iris.csv")
data3.head(3)
```

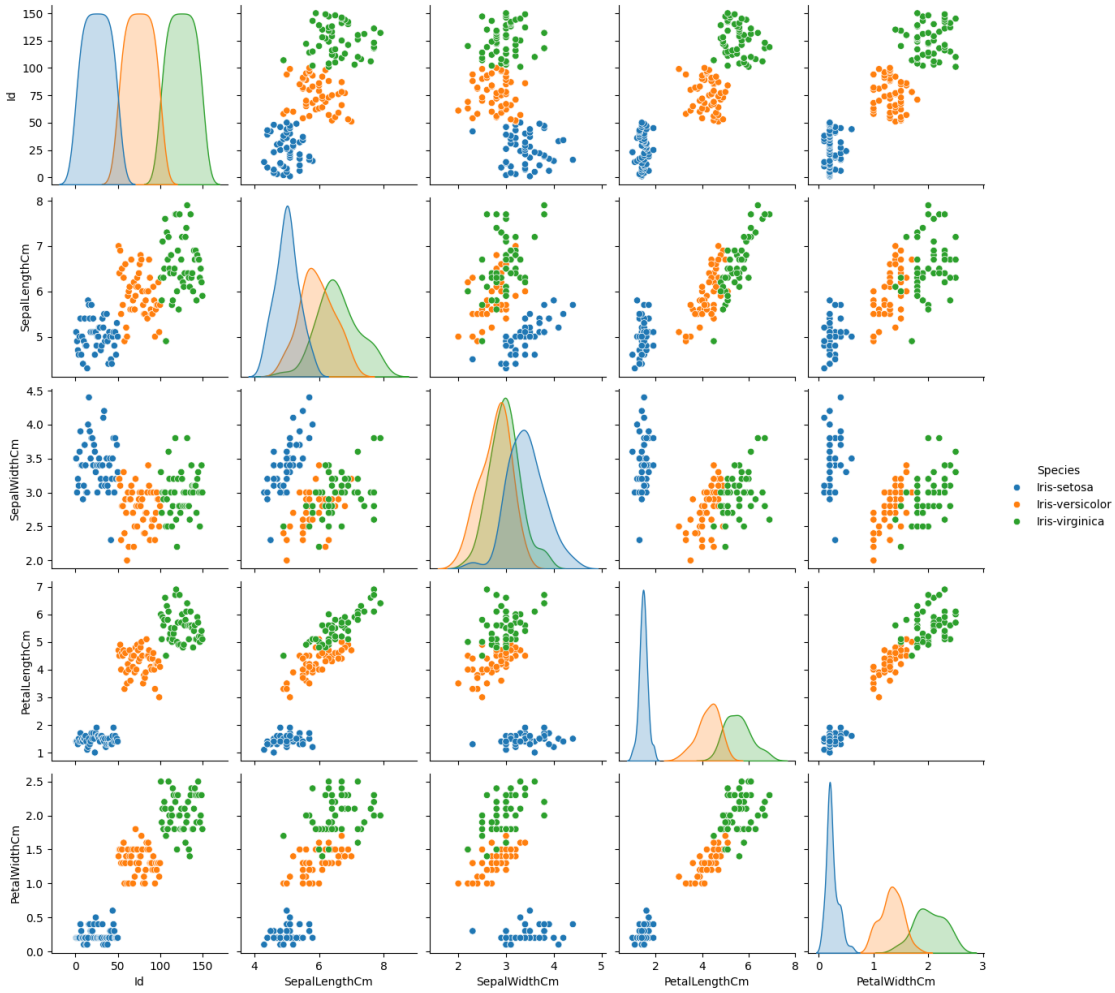
[3]:

	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

[117]:
```
data3["Species"].unique()
```

[117]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

[118]:
```
sns.pairplot(data = data3 , hue = "Species")
plt.show()
```



[4]:
```
iris = load_iris()
data3 = pd.DataFrame(data=iris.data, columns=iris.feature_names)
data3["Species"] = iris.target

x2 = data3.iloc[:, :-1]
y2 = data3["Species"]

x_train, x_test, y_train, y_test = train_test_split(x2, y2, test_size=0.2,␣
 ↪random_state=42)

lr1 = LogisticRegression(multi_class="ovr", max_iter=200)
lr1.fit(x_train, y_train)
print(lr1.score(x_test, y_test)*100)
```

```
96.66666666666667
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/linear_model/_logistic.py:1256: FutureWarning: 'multi_class'
was deprecated in version 1.5 and will be removed in 1.7. Use
OneVsRestClassifier(LogisticRegression(..)) instead. Leave it to its default
value to avoid this warning.
  warnings.warn(
```

6 Confusion Matrix

7 Precision :- TP/TP+FP

```
# It helps to measure the ability to classify positive samples in the model
```

8 Recall :- TP/TP+FN

```
# It Helps to measure how many positive samples were correctly classified by the ML Model
```

9 F1 Score :- 2 * precision * Recall

10 ——————————

11 Precision + Recall

```
# It is harmonic mean of precision and recall . it take both false positive and false negative
# Therfore it perform well an imbalanced dataset
```

[8]:
```
mat_data = pd.read_csv("/Users/ratnadeepgurav/Desktop/AIDS/Study for carrer␣
 ↪material/WSCUBE_Data_Analyst/ML/dataset/placement 3.csv")
mat_data.head(5)
```

[8]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0

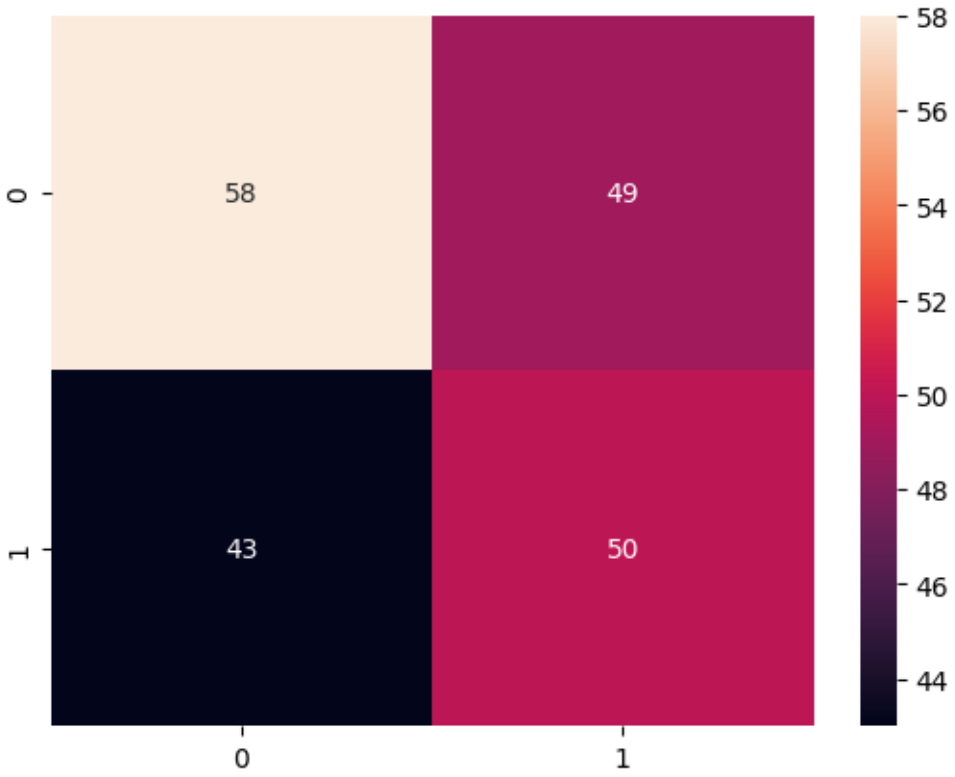
[21]:
```
x3 = mat_data.iloc[:,:-1]
y3 = mat_data["placed"]

x_train, x_test, y_train, y_test = train_test_split(x3, y3, test_size=0.2,␣
 ↪random_state=42)

lr = LogisticRegression()
```

```
lr.fit(x_train,y_train)

lr.score(x_test,y_test) * 100
```

[21]: 54.0

[31]:
```
cf = confusion_matrix(y_test,lr.predict(x_test))
```

[33]:
```
sns.heatmap(cf,annot= True)
plt.show()
```



[46]:
```
ps = precision_score(y_test,lr.predict(x_test))*100
print("\n\n Precision Score :- ",ps,"%\n\n")

rs = recall_score(y_test,lr.predict(x_test))*100
print("\n\n Recall Score    :- ",rs,"%\n\n")

fs = f1_score(y_test,lr.predict(x_test))*100
print("\n\n F1  Score       :- ",fs,"%\n\n")
```

```
Precision Score :-  50.505050505050505 %


Recall Score    :-  53.76344086021505 %


F1  Score       :-  52.083333333333336 %
```

12 Imbalanced Dataset

[47]:
```
# Two Types
#     1 . Random Under Sampling
#     2 . Random Over Sampling
```

[69]:
```
data_bal = pd.read_csv("/Users/ratnadeepgurav/Desktop/AIDS/Study for carrer material/WSCUBE_Data_Analyst/ML/dataset/Social_Network_Ads.csv")

data_bal.drop(columns = ["User ID","Gender"],inplace = True)

data_bal.head(12)
```

[69]:

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0
5	27	58000	0
6	27	84000	0
7	32	150000	1
8	25	33000	0
9	35	65000	0
10	26	80000	0
11	26	52000	0

[108]:
```
data_bal["Purchased"].value_counts()
```

[108]:
```
Purchased
0    257
1    143
Name: count, dtype: int64
```

13 Before Random Under Sample

[68]:
```
x5 = data_bal.iloc[:,:-1]
y5 = data_bal["Purchased"]

x_train,x_test,y_train,y_test = train_test_split(x5,y5,test_size = 0.
 ↪2,random_state = 42)

lr5 = LogisticRegression()

lr5.fit(x_train,y_train)

lr5.score(x_test,y_test) * 100
```

[68]:
```
88.75
```

[107]:
```
lr5.predict([[45,26000        ]])
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

[107]:
```
array([0])
```

14 After Use Random Under Sampler

[119]:
```
ru = RandomUnderSampler()

ru_x , ru_y = ru.fit_resample(x5,y5)

ru_y.value_counts()
```

[119]:
```
Purchased
0    143
1    143
Name: count, dtype: int64
```

[125]:
```
x_train,x_test,y_train,y_test = train_test_split(ru_x,ru_y,test_size = 0.
 ↪2,random_state = 42)

lr6 = LogisticRegression()

lr6.fit(x_train,y_train)

print("\n\n Random Under Sampler :- ",lr6.score(x_test,y_test) * 100,"\n\n")
```

```
Random Under Sampler :-  81.03448275862068
```

15 Random Over Sampling

[117]:
```
ro = RandomOverSampler()

ro_x , ro_y = ro.fit_resample(x5,y5)

ro_y.value_counts()
```

[117]:
```
Purchased
0    257
1    257
Name: count, dtype: int64
```

[123]:
```
x_train,x_test,y_train,y_test = train_test_split(ro_x,ro_y,test_size = 0.
 ↪2,random_state = 42)

lr7 = LogisticRegression()

lr7.fit(x_train,y_train)

print("\n\n Random Over Sampler :- ",lr7.score(x_test,y_test) * 100,"\n\n")
```

```
Random Over Sampler :-  85.43689320388349
```

16 Naive Bayes

17 Types :

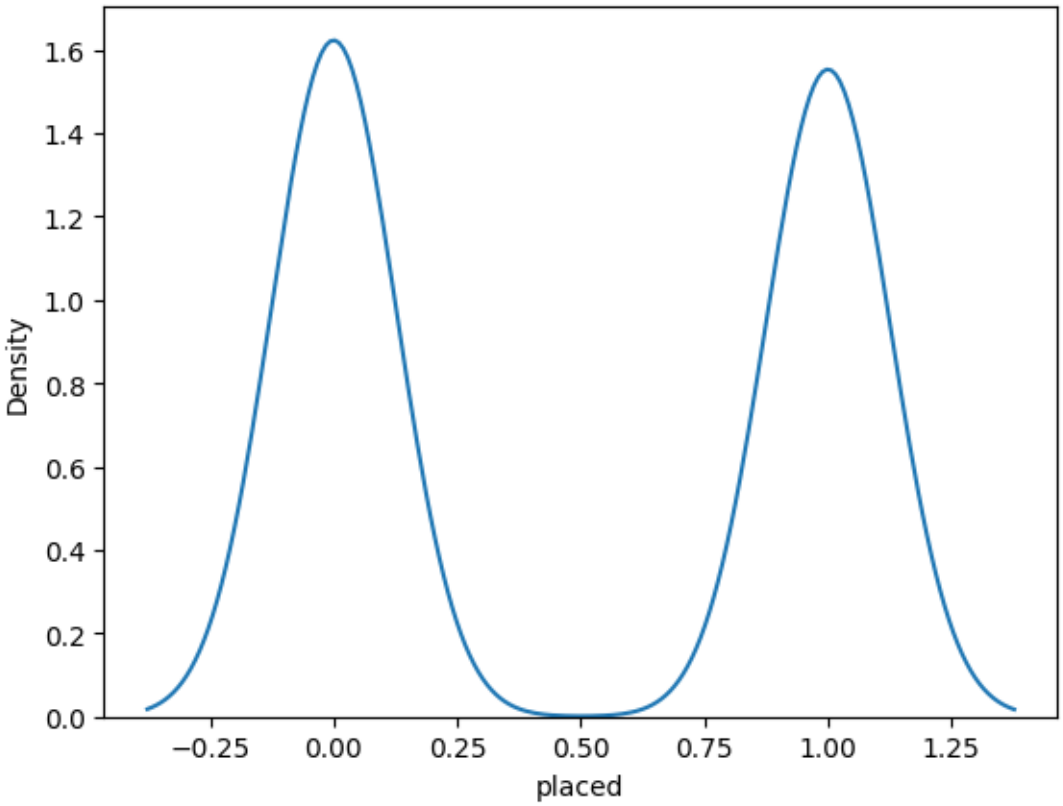
```
# 1 . Gaussian (used when data normal distribution)
# 2 . Multinomial (used text data, descrete data)
# 3 . Bernoulli (used Boolean Variable)
```

[134]:
```
dataa = pd.read_csv("/Users/ratnadeepgurav/Desktop/AIDS/Study for carrer␣
 ↪material/WSCUBE_Data_Analyst/ML/dataset/placement 3.csv")
dataa.head(5)
```

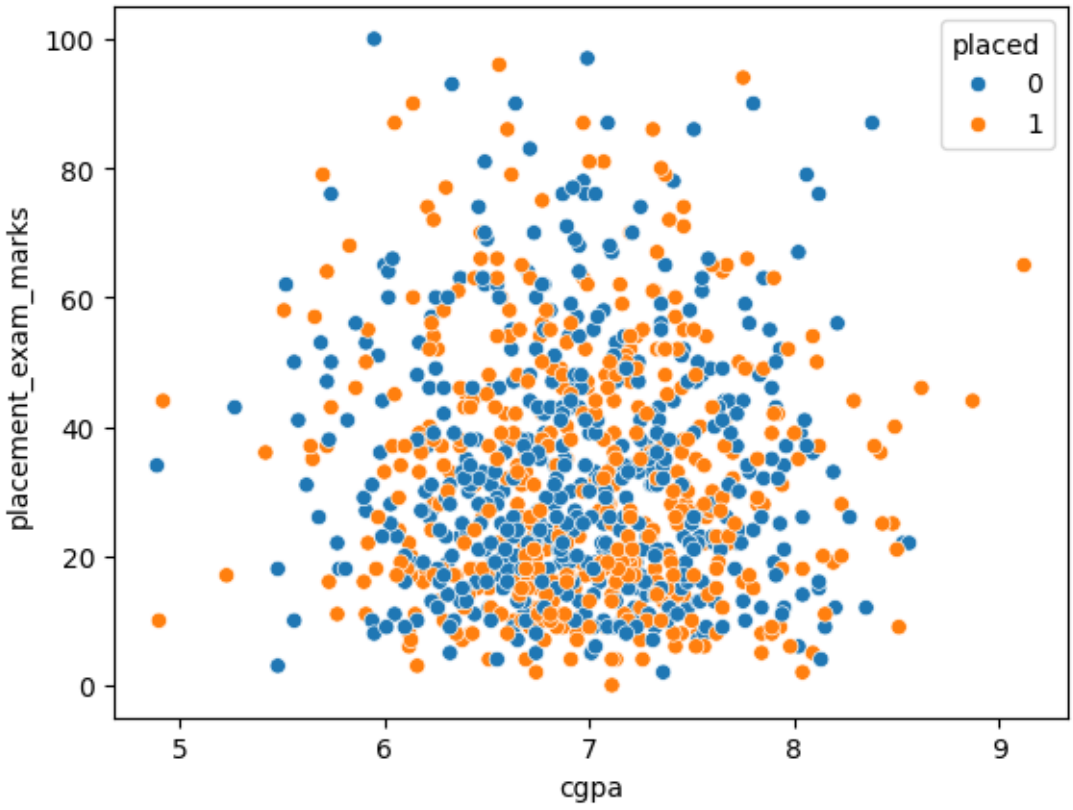
[134]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0

[137]:
```
sns.kdeplot(data=dataa["placed"])
plt.show()
```



[132]:
```
sns.scatterplot(x = "cgpa" , y = "placement_exam_marks" , data = dataa , hue =␣
↪"placed")
plt.show()
```



[145]:
```
x11 = dataa.iloc[:,:-1]
y11 = dataa["placed"]

x_train,x_test,y_train,y_test = train_test_split(x11,y11,test_size = 0.
↪2,random_state = 42)
```

[145]: 53.0

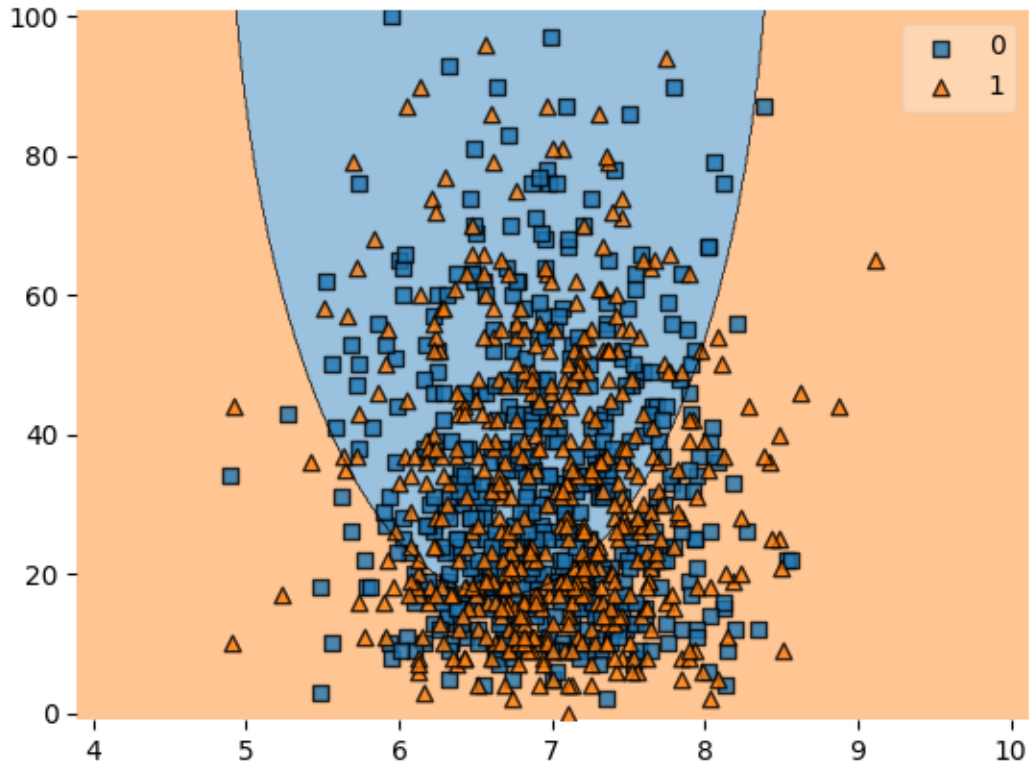
18 GaussianNB

[160]:
```
gs = GaussianNB()

gs.fit(x_train,y_train)

print("\n\n GaussianNB Score :- ",gs.score(x_test,y_test)*100,"\n\n")
```

```
 GaussianNB Score :-  53.0
```

[161]:
```
plot_decision_regions(x11.to_numpy(),y11.to_numpy(),clf = gs)
plt.show()
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



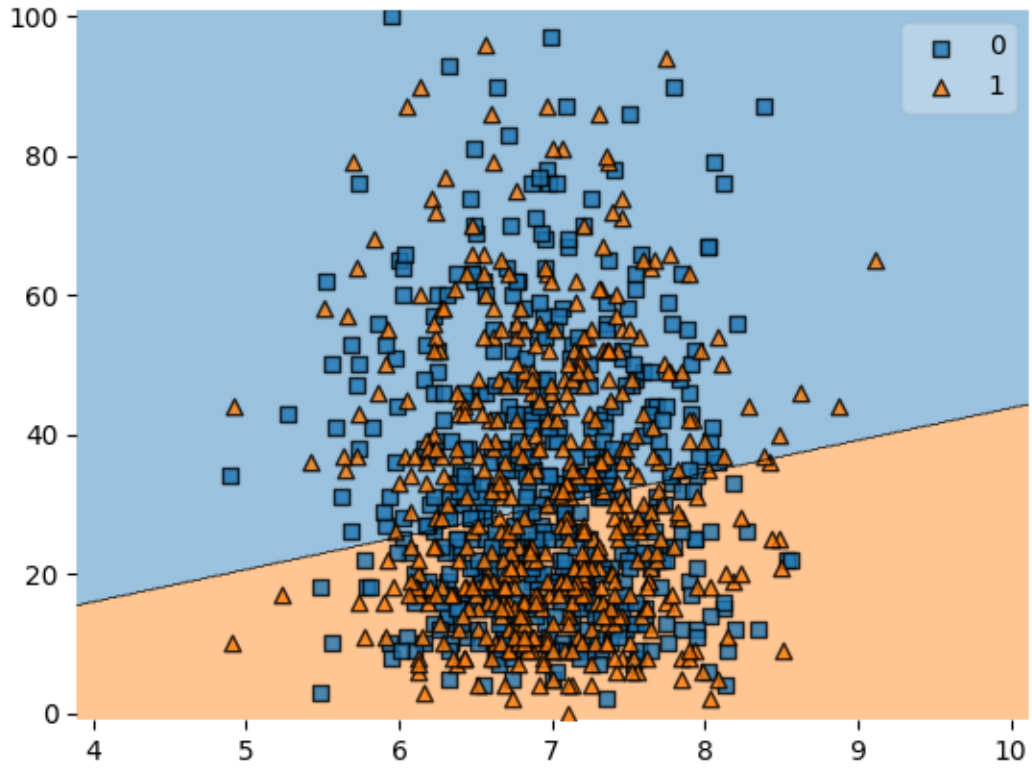
19 MultinomialNB

[163]:
```
mn = MultinomialNB()

mn.fit(x_train,y_train)

print("\n\n GaussianNB Score :- ",mn.score(x_test,y_test)*100,"\n\n")
```

```
 GaussianNB Score :-  53.5
```

[164]:
```
plot_decision_regions(x11.to_numpy(),y11.to_numpy(),clf = mn)
plt.show()
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but MultinomialNB was fitted with feature names
  warnings.warn(
```



20 BernoulliNB

[157]:
```
bs = BernoulliNB()

bs.fit(x_train,y_train)

print("\n\n GaussianNB Score :- ",bs.score(x_test,y_test)*100,"\n\n")
```

```
 GaussianNB Score :-  53.5
```

[159]:
```
plot_decision_regions(x11.to_numpy(),y11.to_numpy(),clf = bs)
plt.show()
```

```
/opt/homebrew/Cellar/jupyterlab/4.2.3/libexec/lib/python3.12/site-
packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid
feature names, but BernoulliNB was fitted with feature names
  warnings.warn(
```