

Tushar A. Kamble
Roll No. 29
B.Tech - A (CSE)

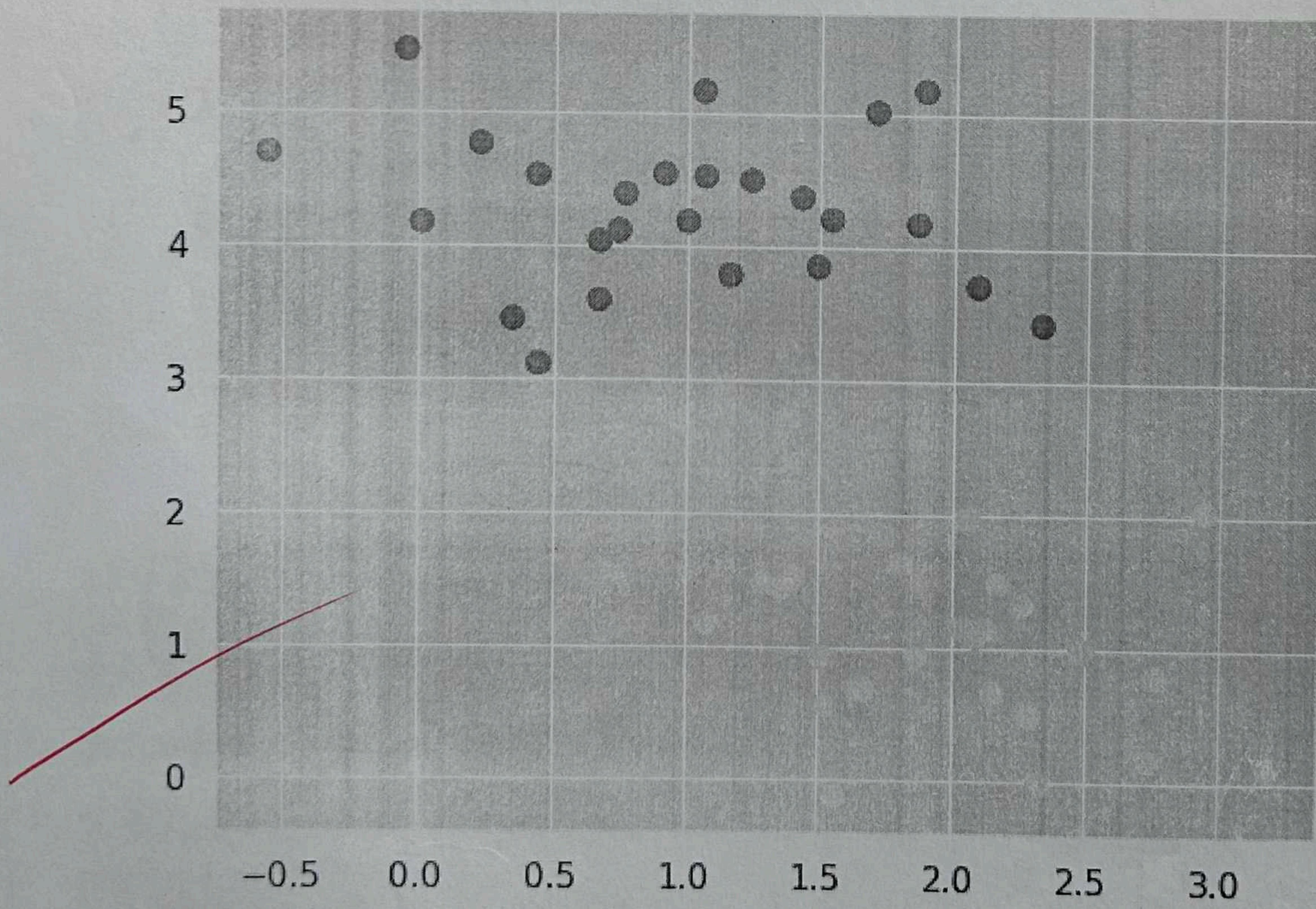
10 4em
10

SVM

November 1, 2023

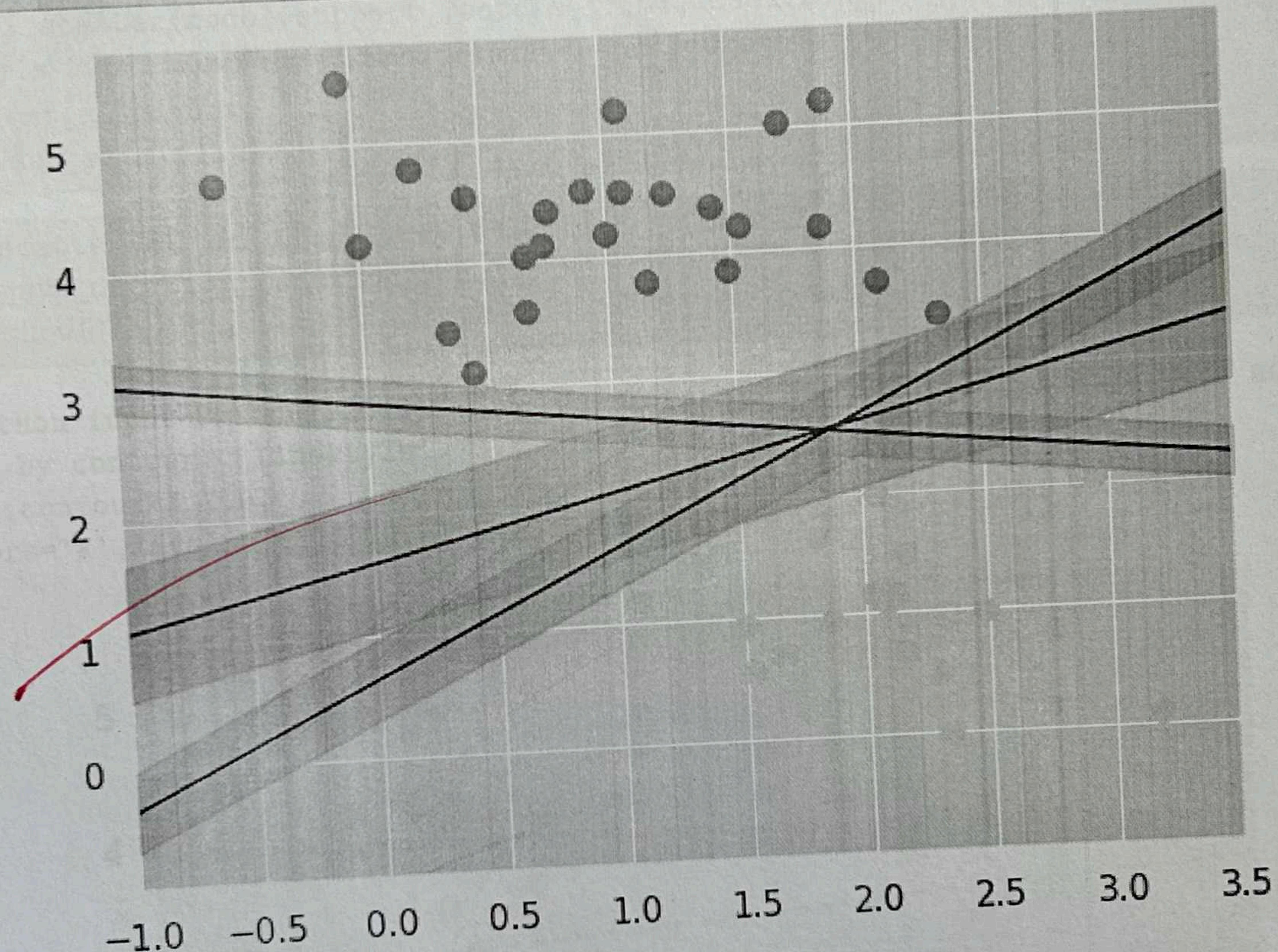
```
[ ]: import numpy as np  
import matplotlib.pyplot as plt  
from scipy import stats  
import seaborn as sns ;  
sns.set()
```

```
[ ]: from sklearn.datasets import make_blobs  
import matplotlib.pyplot as plt  
  
X, y = make_blobs(n_samples=50,centers=2,random_state=0,cluster_std=0.60)  
plt.scatter(X[:,0],X[:,1],c=y,s=50,cmap='autumn')  
plt.show()
```



```
[ ]: xfit = np.linspace(-1,3.5)
plt.scatter(X[:,0],X[:,1],c=y,s=50,cmap='autumn')
for m,b,d in [(1,0.65,0.33),(0.5,1.6,0.55),(-0.2,2.9,0.2)]:
    yfit = m*xfit+b
    plt.plot(xfit,yfit,'-k')
    plt.fill_between(xfit,yfit-d,yfit+d,edgecolor='none',color="#AAAAAA",alpha=0.4)

plt.xlim(-1,3.5)
plt.show()
```



```
[ ]: from sklearn.svm import SVC
model = SVC(kernel='linear',C=1E10)
model.fit(X,y)

[ ]: SVC(C=10000000000.0, kernel='linear')

[ ]: def plot_svc_decision_function(model,ax=None,plot_support=True):
    if ax is None:
        ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()
```

```

x = np.linspace(xlim[0], xlim[1], 30)
y = np.linspace(ylim[0], ylim[1], 30)
Y, X = np.meshgrid(y, x)
xy = np.vstack([X.ravel(), Y.ravel()]).T
P = model.decision_function(xy).reshape(X.shape)

ax.contour(X, Y, P, colors='k', levels=[-1, 0, 1], alpha=0.5, linestyle=['--', '-'])
if plot_support:
    ax.scatter(model.support_vectors_[:, 0], model.support_vectors_[:, 1], s=300, linewidth=1, facecolors='none', edgecolors='k')
ax.set_xlim(xlim)
ax.set_ylim(ylim)

[ ]: plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')
plot_svc_decision_function(model);
plt.show()

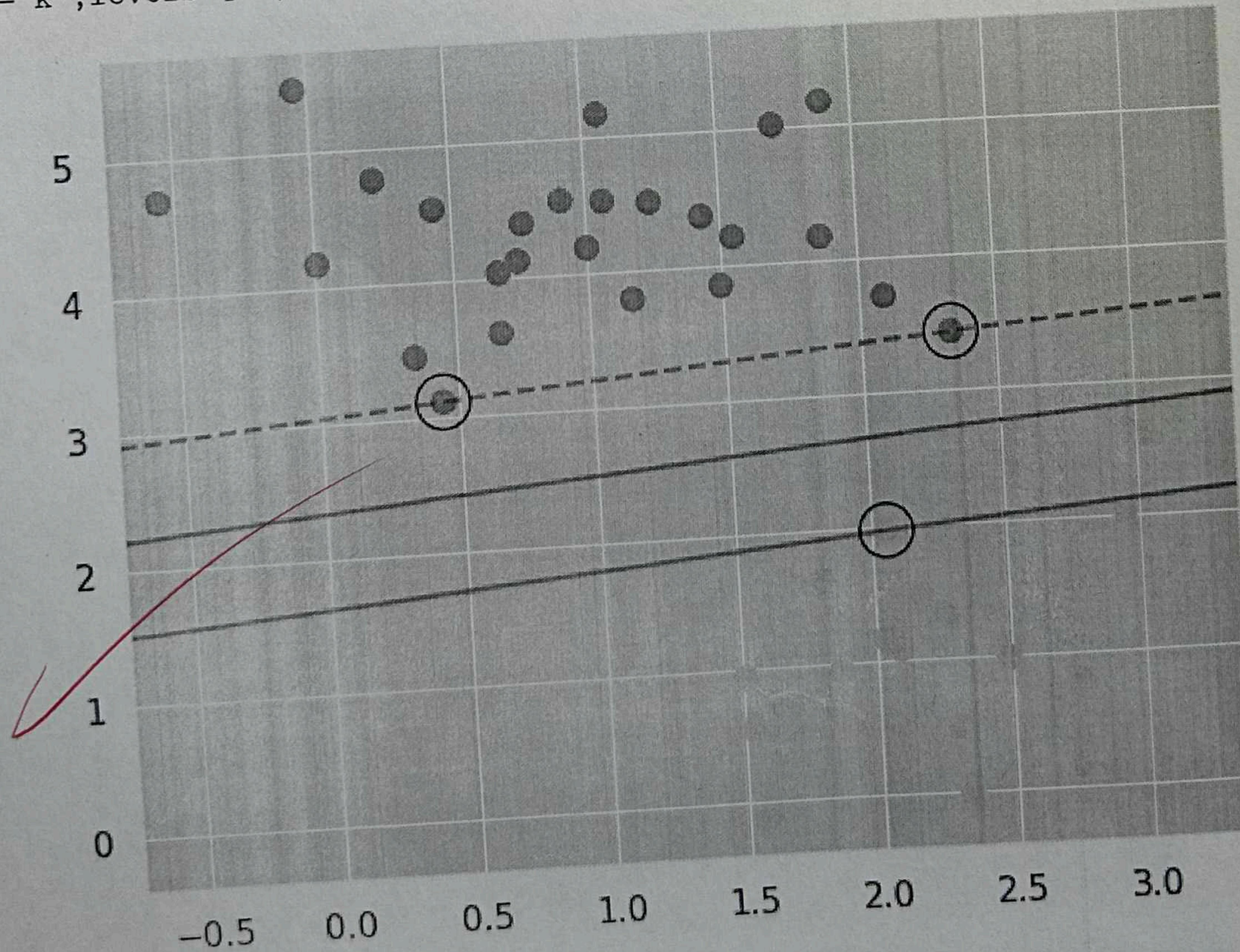
```

<ipython-input-9-fbd67126d64f>:13: UserWarning: The following kwargs were not used by contour: 'linestyle'

```

    ax.contour(X, Y, P,
               colors='k', levels=[-1, 0, 1], alpha=0.5, linestyle=['--', '-'])

```



```

[ ]: model.support_vectors_
[ ]: array([[0.44359863, 3.11530945],
           [2.33812285, 3.43116792],
           [2.06156753, 1.96918596]])

[ ]: def plot_svc_decision_function(model,ax=None,plot_support=True):
    if ax is None:
        ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()

    x = np.linspace(xlim[0],xlim[1],30)
    y = np.linspace(ylim[0],ylim[1],30)
    Y, X = np.meshgrid(y,x)
    xy = np.vstack([X.ravel(),Y.ravel()]).T
    P = model.decision_function(xy).reshape(X.shape)

    ax.contour(X,Y,P, colors='k',levels=[-1,0,1],alpha=0.5,linestyle=['--','-' '--'])
    if plot_support:
        ax.scatter(model.support_vectors_[:,0],model.support_vectors_[:,1],s=300,linewidth=1,facecolors='none',edgecolors='k')

def plot_svm(N=10,ax=None):
    X,y = make_blobs(n_samples=50,centers=2,random_state=0,cluster_std=0.60)
    X = X[:N]
    y = y[:N]
    model = SVC(kernel='linear',C=1E10)
    model.fit(X,y)

    ax = ax or plt.gca()
    ax.scatter(X[:,0],X[:,1],c=y,s=50,cmap='autumn')
    ax.set_xlim(-1,4)
    ax.set_ylim(-1,6)
    plot_svc_decision_function(model,ax)

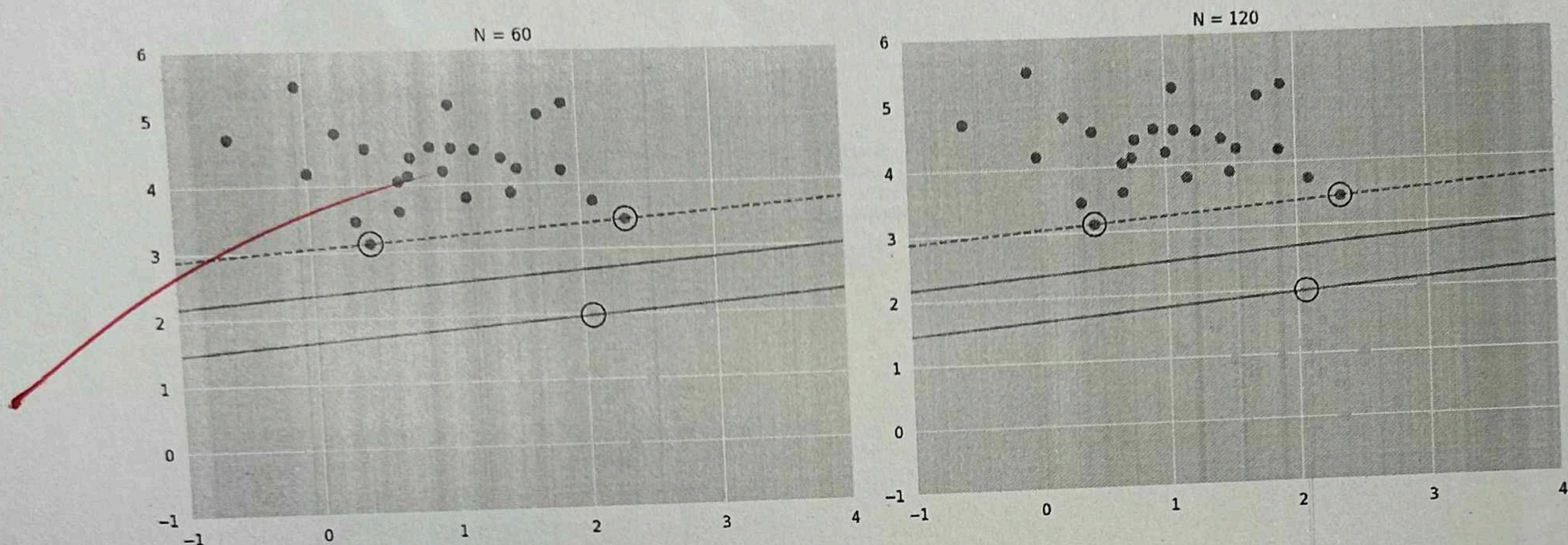
fig,ax = plt.subplots(1,2,figsize=(16,6))
fig.subplots_adjust(left=0.0625,right=0.95,wspace=0.1)

for axi, N in zip(ax,[60,120]):
    plot_svm(N,axi)
    axi.set_title('N = {}'.format(N))

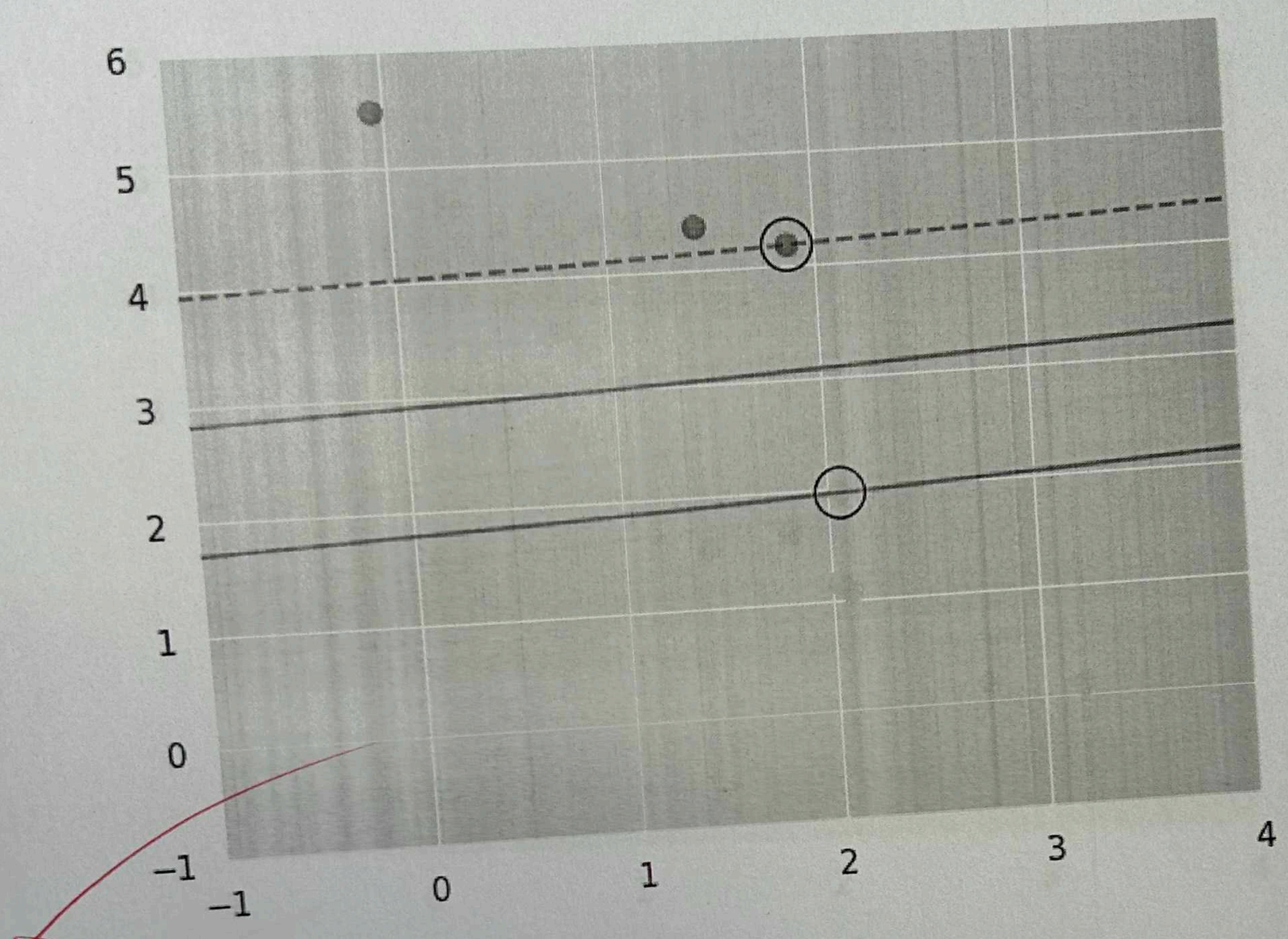
<ipython-input-15-cc8b90336398>:13: UserWarning: The following kwargs were not

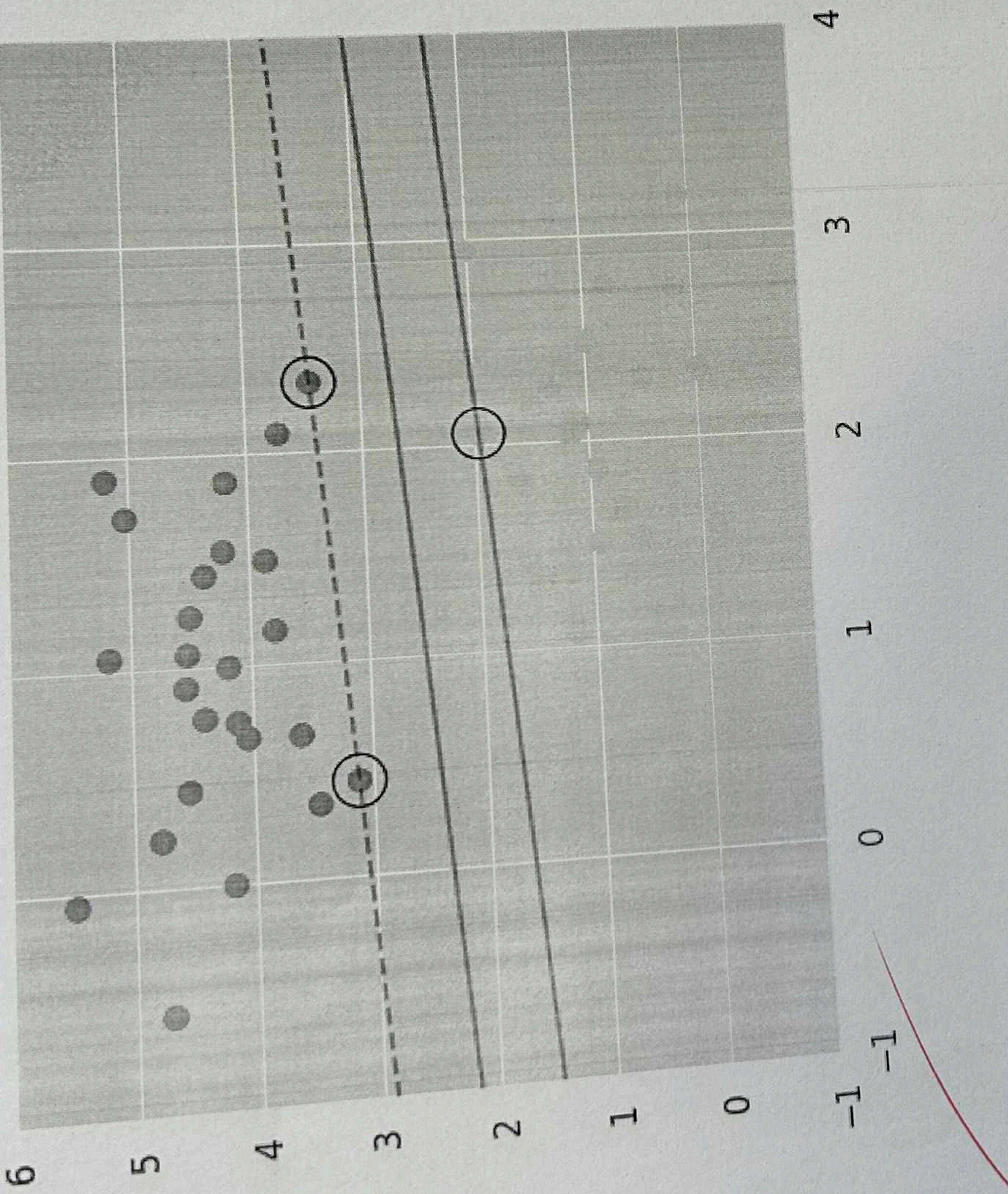
```

```
used by contour: 'linestyle'
    ax.contour(X,Y,P,
colors='k',levels=[-1,0,1],alpha=0.5,linestyle=['--','-' '--'])
<ipython-input-15-cc8b90336398>:13: UserWarning: The following kwargs were not
used by contour: 'linestyle'
    ax.contour(X,Y,P,
colors='k',levels=[-1,0,1],alpha=0.5,linestyle=['--','-' '--'])
```

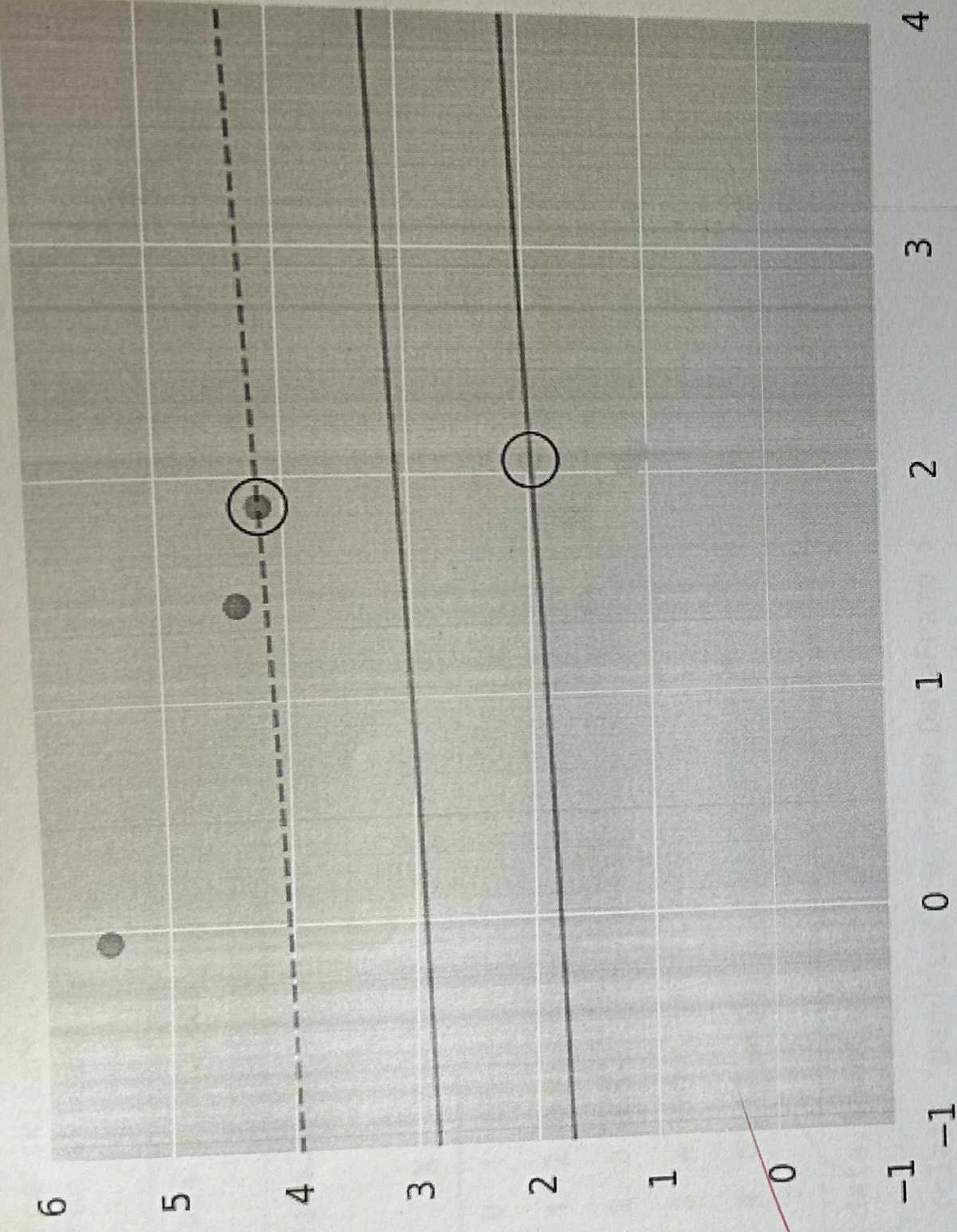


```
[ ]: from ipywidgets import interact,fixed
      interact(plot_svm,N=[10,200],ax=fixed(None))
      interactive(children=(Dropdown(description='N', options=(10, 200), value=10),
      Output()), _dom_classes=('widget...'))
[ ]: <function __main__.plot_svm(N=10, ax=None)>
```

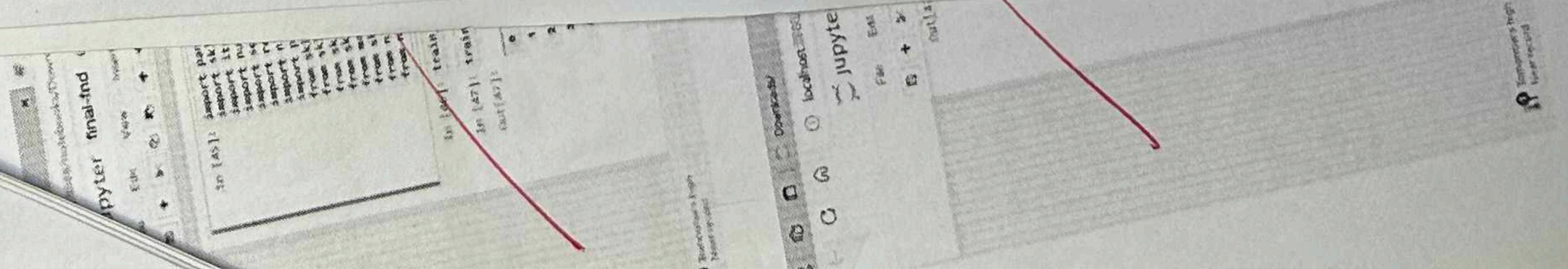




7



~~Final 22
22~~



```
In [12]: import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
In [12]: data = pd.read_csv('./lrData.csv')
```

```
In [3]: data.head()
```

```
Out[3]:
```

	X	Y
0	1	2.5
1	2	3.5
2	3	2.8
3	4	5.0
4	5	4.7

```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype  
--- 
 0   X               30 non-null    int64  
 1   Y               30 non-null    float64
dtypes: float64(1), int64(1)
memory usage: 612.0 bytes
```

```
In [18]: x1 = data.iloc[:,0]
y = data.iloc[:,1]
```

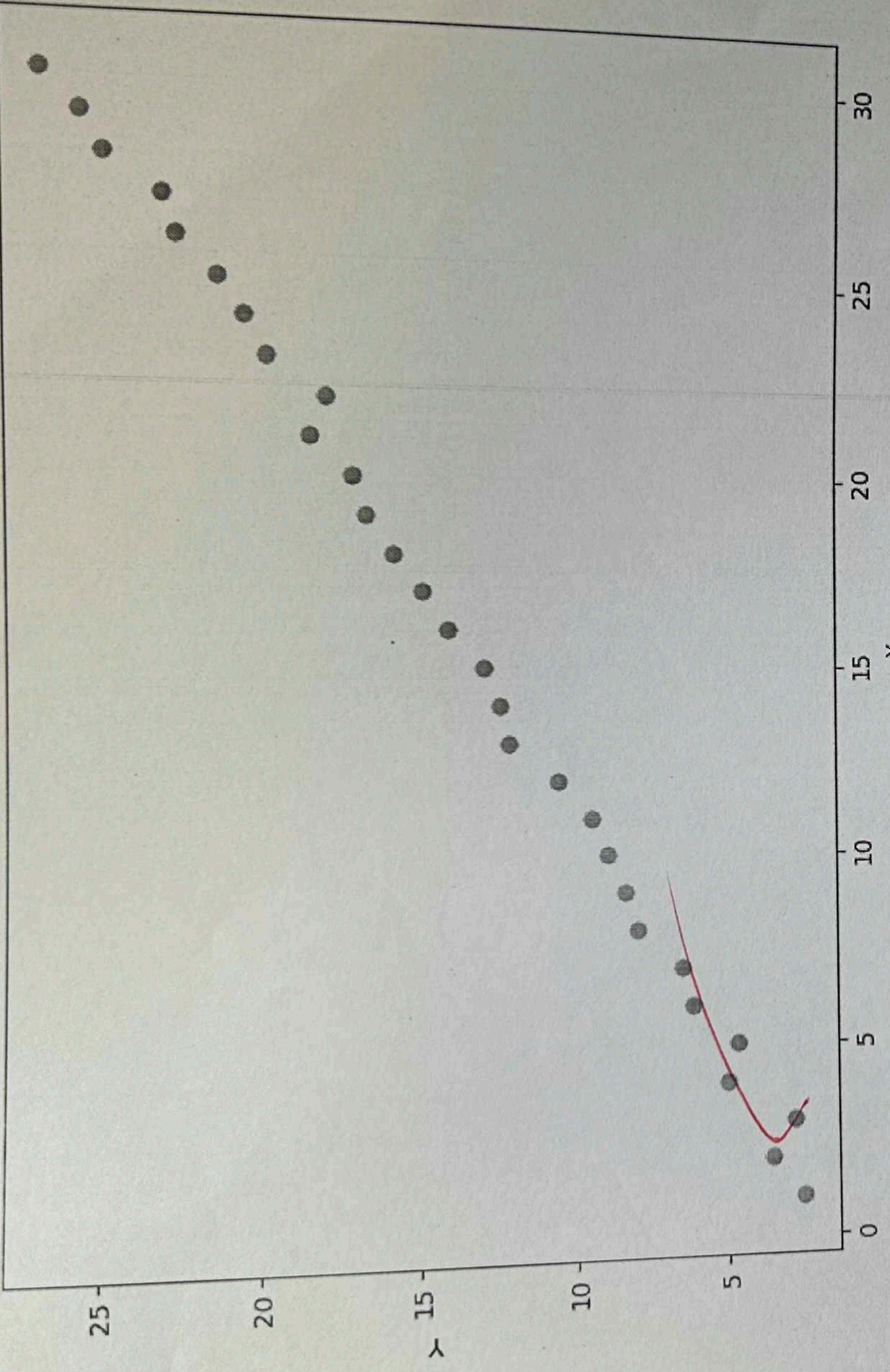
```
# In[18]: x1 = data.iloc[:,0]
# y = data.iloc[:,1]
```

Tushar A. Kamble
Roll No. 29
B.Tech - A (ESE)

8/10

```
In [19]: plt.figure(figsize=(9, 6))
plt.scatter(x1, y, color='red', label='')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Linear regression')
plt.show()
```

Linear regression



```
In [36]: X_train, X_test, y_train, y_test = train_test_split(x1, y, test_size=0.2, r
```

```
In [37]: linear_regressor = LinearRegression()
```

```
In [38]: linear_regressor.fit(X_train, y_train)
```

```
Out[38]:
```

```
LinearRegression()
LinearRegression()
```

```
In [39]: y_pred = linear_regressor.predict(X_test)
```

```
In [40]: y_pred
```

```
Out[40]: array([24.05986729, 14.21229748, 20.77734402, 15.85355912, 8.46788176,
9.28851258])
```

```
In [41]: mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

~~8 / 10~~ Few 3/11/93

3, 1:27 PM

linearRegression - Jupyter Notebook

```
In [42]: print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared score: {r2}")
```

```
Mean Squared Error (MSE): 0.1191452608759399
R-squared score: 0.996529637373755
```

In []:

Few

House Data

1 1 1

2 2 2

3 3 3

4 4 4

5 5 5

6 6 6

7 7 7

8 8 8

9 9 9

10 10 10

11 11 11

12 12 12

13 13 13

14 14 14

15 15 15

16 16 16

17 17 17

18 18 18

19 19 19

20 20 20

21 21 21

22 22 22

23 23 23

24 24 24

25 25 25

26 26 26

27 27 27

28 28 28

29 29 29

30 30 30

31 31 31

32 32 32

33 33 33

34 34 34

35 35 35

36 36 36

37 37 37

38 38 38

39 39 39

40 40 40

41 41 41

42 42 42

43 43 43

44 44 44

45 45 45

46 46 46

47 47 47

48 48 48

49 49 49

50 50 50

51 51 51

52 52 52

53 53 53

54 54 54

55 55 55

56 56 56

57 57 57

58 58 58

59 59 59

60 60 60

61 61 61

62 62 62

63 63 63

64 64 64

65 65 65

66 66 66

67 67 67

68 68 68

69 69 69

70 70 70

71 71 71

72 72 72

73 73 73

74 74 74

75 75 75

76 76 76

77 77 77

78 78 78

79 79 79

80 80 80

81 81 81

82 82 82

83 83 83

84 84 84

85 85 85

86 86 86

87 87 87

88 88 88

89 89 89

90 90 90

91 91 91

92 92 92

93 93 93

94 94 94

95 95 95

96 96 96

97 97 97

98 98 98

99 99 99

100 100 100

101 101 101

102 102 102

103 103 103

104 104 104

105 105 105

106 106 106

107 107 107

108 108 108

109 109 109

110 110 110

111 111 111

112 112 112

113 113 113

114 114 114

115 115 115

116 116 116

117 117 117

118 118 118

119 119 119

120 120 120

121 121 121

122 122 122

123 123 123

124 124 124

125 125 125

126 126 126

127 127 127

128 128 128

129 129 129

130 130 130

131 131 131

132 132 132

133 133 133

134 134 134

135 135 135

136 136 136

137 137 137

138 138 138

139 139 139

140 140 140

141 141 141

142 142 142

143 143 143

144 144 144

145 145 145

146 146 146

147 147 147

148 148 148

149 149 149

150 150 150

151 151 151

152 152 152

153 153 153

154 154 154

155 155 155

156 156 156

157 157 157

158 158 158

159 159 159

160 160 160

161 161 161

162 162 162

163 163 163

164 164 164

165 165 165

166 166 166

167 167 167

168 168 168

169 169 169

170 170 170

171 171 171

172 172 172

173 173 173