A20474413
Jorge Gonzalez Lopez

# Assignment 4

S484 Spring 2021
March 27, 2021

# Question 1

a)

- **Answer**

| Insurance | 0 | 1 | 2 |
|---|---|---|---|
| Frequency Count | 143691 | 426067 | 95491 |
| Class Probability | 0.215996 | 0.640462 | 0.143542 |

- **Explanation**

To do so, the pandas' function df['insurance'].value_counts() (with and without the option normalize) gives back the frequency counts and the Class Probabilities of the target variable (insurance).

b)

- **Answer**

| Group_size | Insurance | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| 1 | 115460 | 329552 | 74293 |
| 2 | 25728 | 91065 | 19600 |
| 3 | 2282 | 5069 | 1505 |
| 4 | 221 | 381 | 93 |

- **Explanation**

To do so, ther are two easy ways:

1. df.groupby(['group_size','insurance']).size())
2. pd.crosstab(df['group_size'], df['insurance']) (this way outputs the result in the same format as the table shown above).

c)

- **Answer**

| Homeowner | Insurance | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| 0 | 78659 | 183130 | 46734 |
| 1 | 65032 | 242937 | 48757 |

- **Explanation**

  To do so, the pandas' function pd.crosstab(df['homeowner'], df['insurance']) is used.

d)

- **Answer**

| Married_couple | insurance | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| 0 | 117110 | 333272 | 75310 |
| 1 | 26581 | 92795 | 20181 |

- **Explanation**

  To do so, the pandas' function pd.crosstab(df['married_couple'], df['insurance']) is used.

e)

- **Answer**

| Feature | Cramer's V |
|---|---|
| group_size | 0.0271020 |
| homeowner | 0.0970864 |
| married_couple | 0.0324216 |

- **Explanation**

  The feature with the strongest association with the target value is the one with the highest Cramer's V statistic. In this case, this feature is **homeowner.**

f)

- **Answer**

| group_size | homeowner | married_couple | P_ins0 | P_ins1 | P_ins2 |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0.269722 | 0.580133 | 0.150145 |
| 1 | 0 | 1 | 0.232789 | 0.614219 | 0.152992 |
| 1 | 1 | 0 | 0.194038 | 0.669659 | 0.136303 |
| 1 | 1 | 1 | 0.164935 | 0.698278 | 0.136787 |
| 2 | 0 | 0 | 0.231143 | 0.616518 | 0.152338 |
| 2 | 0 | 1 | 0.198016 | 0.647907 | 0.154078 |
| 2 | 1 | 0 | 0.163628 | 0.700288 | 0.136085 |
| 2 | 1 | 1 | 0.138274 | 0.725955 | 0.135771 |
| 3 | 0 | 0 | 0.308219 | 0.515924 | 0.175856 |
| 3 | 0 | 1 | 0.268311 | 0.550951 | 0.180738 |
| 3 | 1 | 0 | 0.226972 | 0.609612 | 0.163416 |
| 3 | 1 | 1 | 0.194370 | 0.640410 | 0.165221 |
| 4 | 0 | 0 | 0.375490 | 0.487810 | 0.136700 |
| 4 | 0 | 1 | 0.330743 | 0.527098 | 0.142158 |
| 4 | 1 | 0 | 0.282173 | 0.588196 | 0.129631 |
| 4 | 1 | 1 | 0.243930 | 0.623766 | 0.132304 |

- **Explanation**

A Naïve Bayes model without any smoothing has been trained using all the observations. The Laplace/Lidstone alpha has been set to 1e-10 instead of zero as an alpha too small will result, in the sklearn library, in numeric errors.

g)

- **Answer**

The maximum odds value of *Prob(insurance=1)/Prob(insurance=2)* = 5.3469126

The value combination that yields it is:

group_size = 2, homeowner = 1, married_couple = 1

- **Explanation**

It was calculated by dividing the column P_ins1 by P_ins2 of the dataframe computed in question f) and getting the line that has the maximum value of them all.

# Question 2

a)

- **Answer**

```
The equation is:  0.0033450  +  0.0533351x +  0.3286838y = 0
```

- **Explanation**

The coefficients are given by coef_ and intercept_ of the SVM.SVC model.

b)

- **Answer**

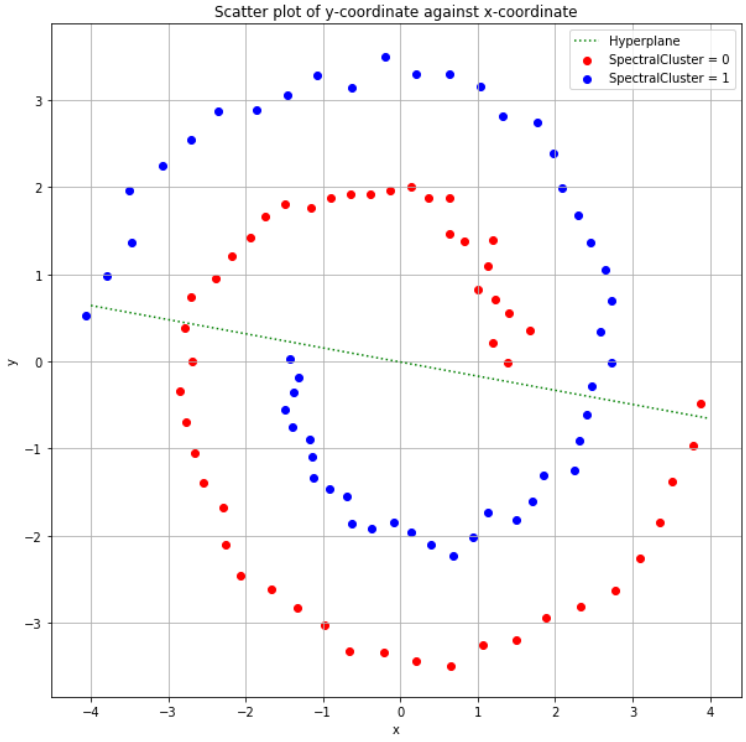
The misclassification rate is 0.5

- **Explanation**

The model does not have a good accuracy as the values cannot be split with a linear hyperplane.

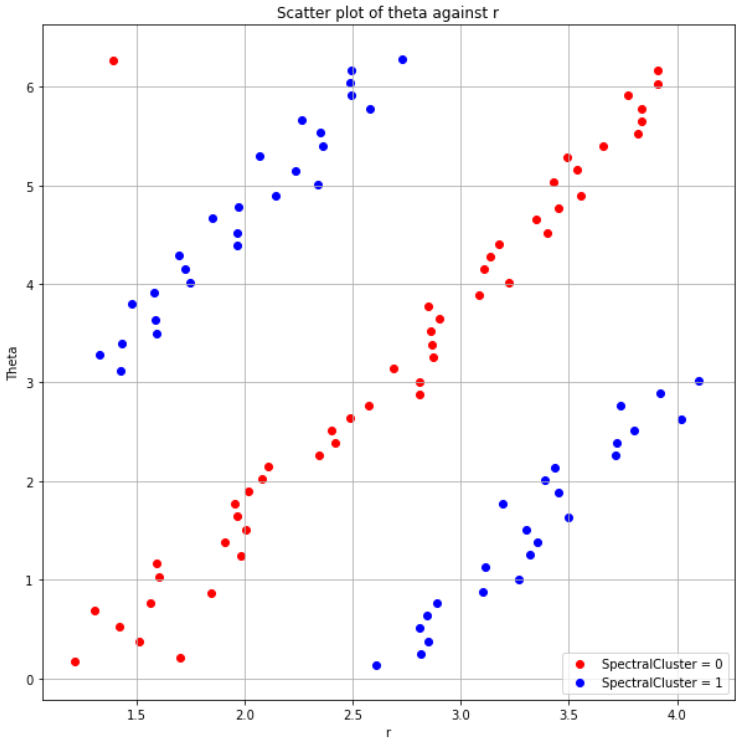
c)

- **Figure**



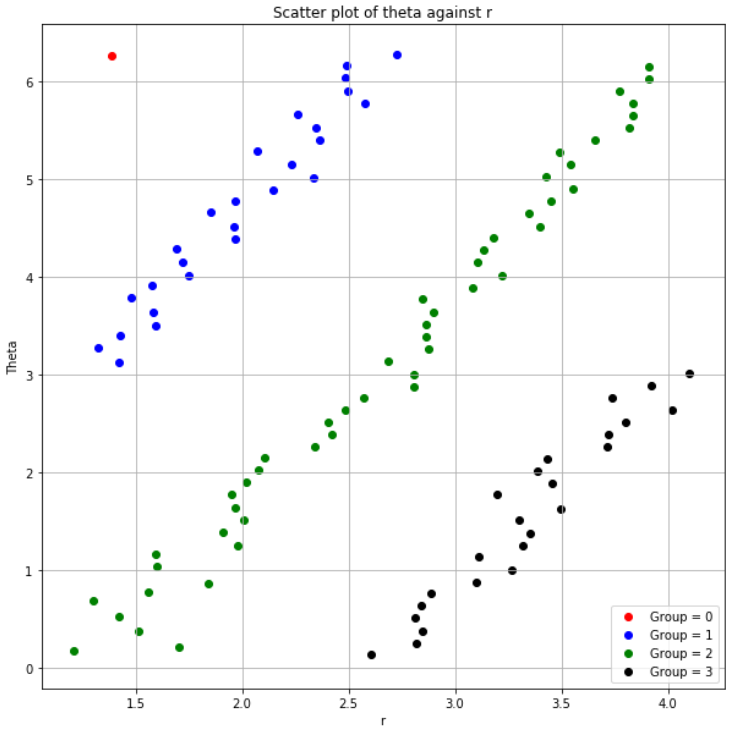

d)

- **Figure**



e)

- **Figure**




f)

- **Answer**

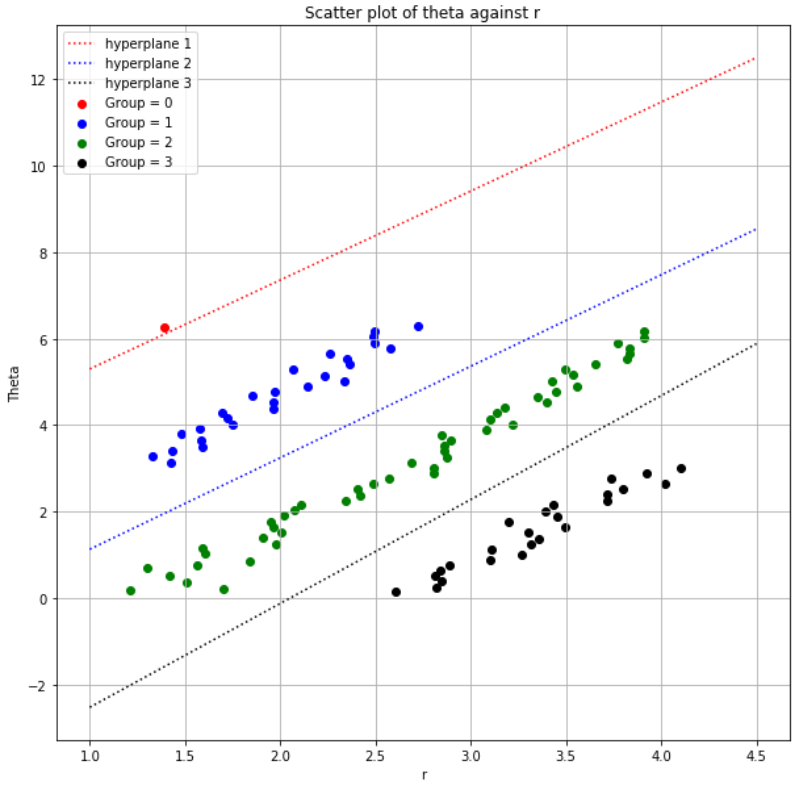
```
SVM 0
The intercept w0 is:  [1.46912508]
The coeficients are:  [[ 0.93378415 -0.45380249]]
The equation is:  1.4691251  +  0.9337841 r +  -0.4538025 theta = 0
SVM 1
The intercept w0 is:  [0.88406321]
The coeficients are:  [[-1.88674959  0.8914745 ]]
The equation is:  0.8840632  +  -1.8867496 r +  0.8914745 theta = 0
SVM 2
The intercept w0 is:  [-4.13284488]
The coeficients are:  [[ 2.01258355 -0.83756164]]
The equation is:  -4.1328449  +  2.0125835 r +  -0.8375616 theta = 0
```

- **Explanation:**

  Split the data depending on the value of ‘Group’ and apply SVM three times as in (a).

g)

- **Figure**

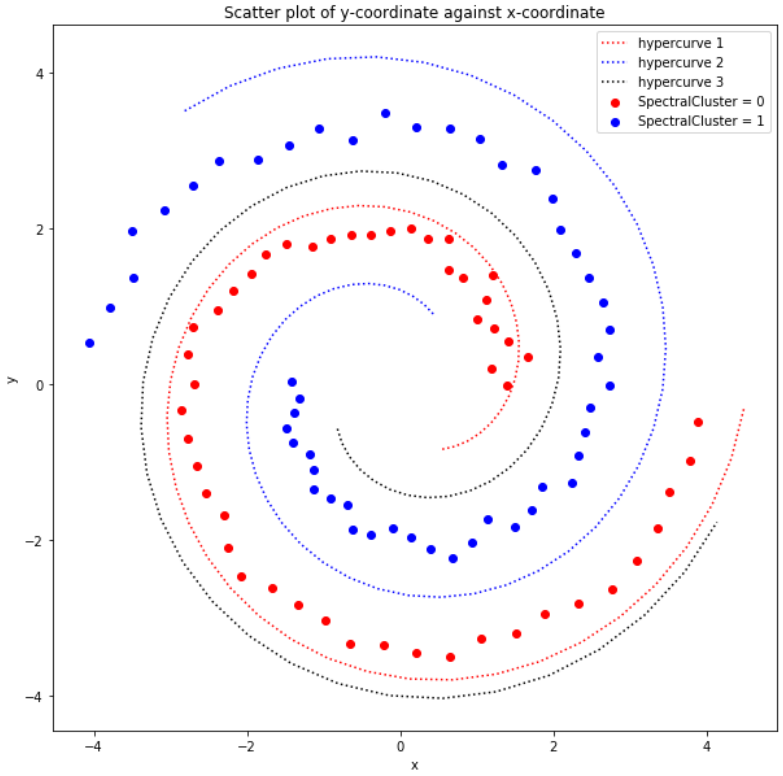



- **Explanation:**

  The hyperplane 1 is the one obtained from applying SVM to Group 0 and 1, hyperplane 2 to Group 1 and 2 and hyperplane 3 to Group 2 and 3.

h)

- **Figure**




- **Answer**

  The hyper curve that is not needed is hyper curve 1 because hyper curve 3 performs the same classification but avoiding some misclassifications.