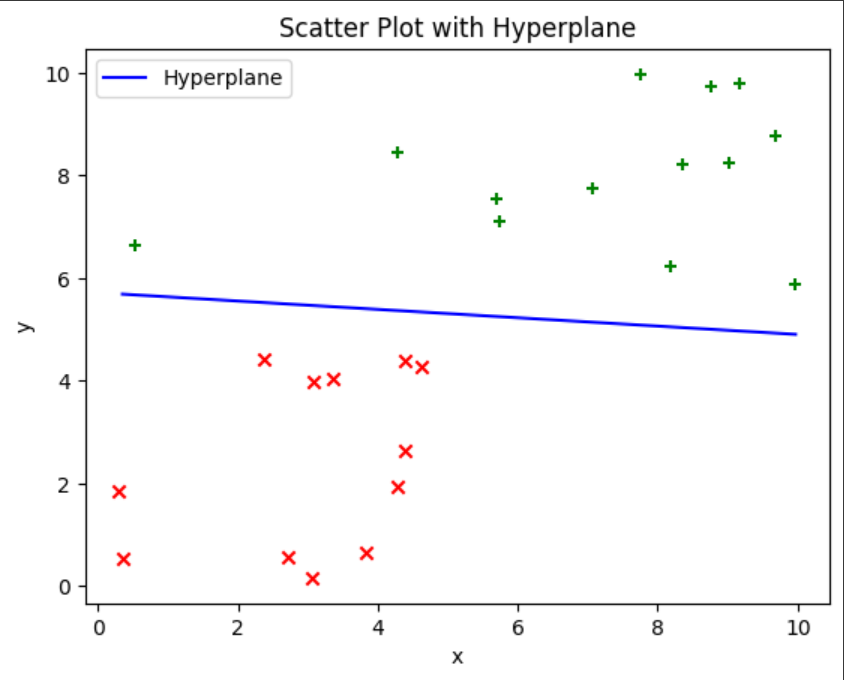
**Assignment 3**

**Question 1 a)**

[**https://colab.research.google.com/drive/1niZ-Iwb1IrNvoVkUOGyETxJQIHi4RKnJ?usp=sharing**](https://colab.research.google.com/drive/1niZ-Iwb1IrNvoVkUOGyETxJQIHi4RKnJ?usp=sharing)

**From the scatter plot where green ‘+’ signs denoting the positive instances of data, and red ‘x’ showing the negative instances of data, keeping in mind 1 is classified as positive, and -1 is classified as negative.**

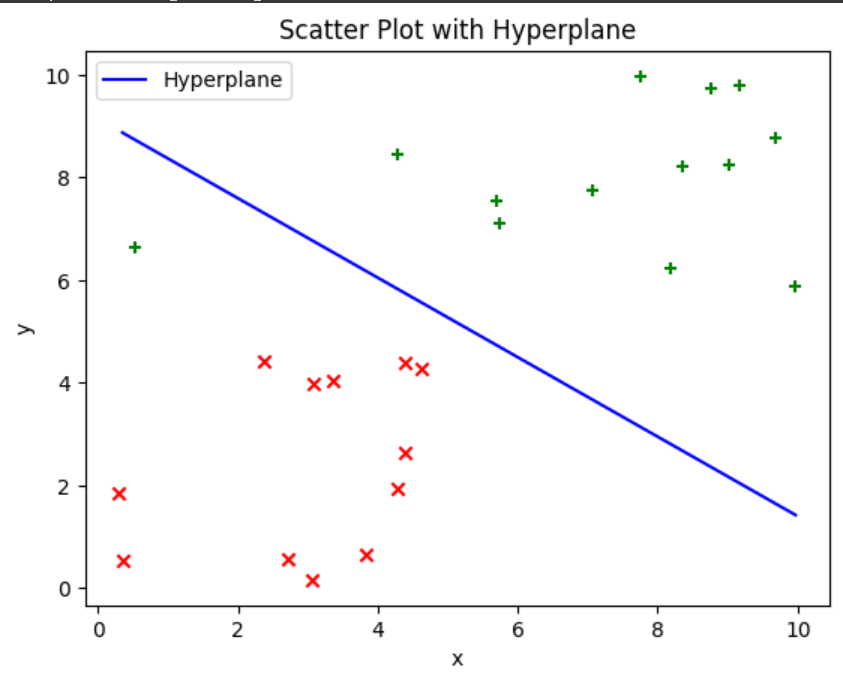
**We can see that the ‘+’ data points are completely separate from the ‘x’ data points shown in red.**

****

**b) From the scatter plot, it is showing seperately the positively classified elements and negatively classified elements. As we are familiar with svm it ignores the misclassified, or the outliers in data, so in our case, if there was an outlier, using the basic svm, there would be a data point on the wrong side, as the linear svm divides data in two parts, one positive classification and the other negative classification. If there was an outlier, so in the positive side, there would be a data point of negative class or vice versa. So using the default svm, it can be seen there is not outlier.**

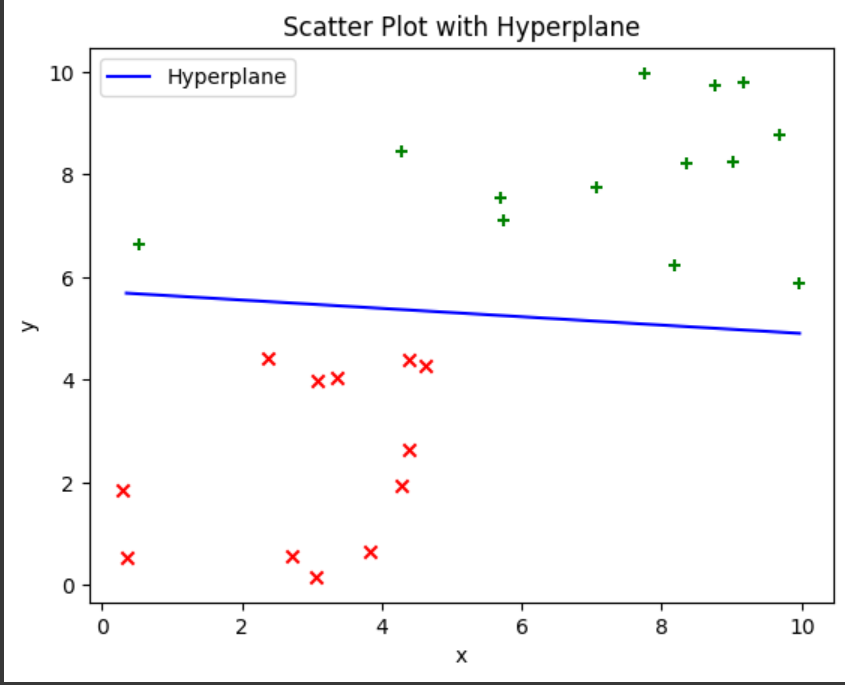
**Question 2**

1. [**https://colab.research.google.com/drive/1WNUUP-K7jeB5aAvG9VyQVM3d6Lj0kR4a?usp=sharing**](https://colab.research.google.com/drive/1WNUUP-K7jeB5aAvG9VyQVM3d6Lj0kR4a?usp=sharing)

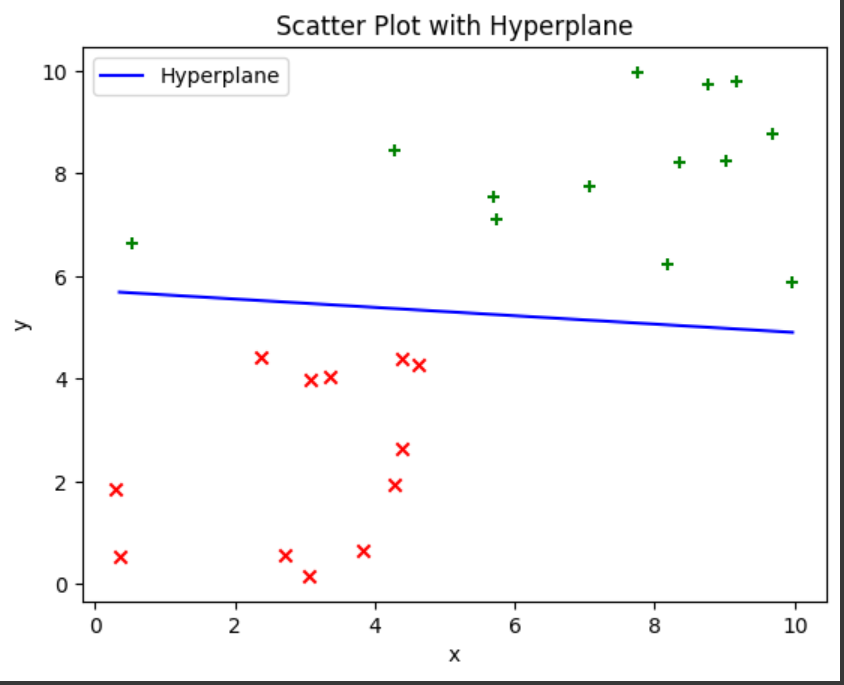
****

**Yes in this case, taking the value of c=0.01, it can be seen that there is an outlier. As we know that svm ignores the outliers and linear svm divides the data in two parts, so in our case, below the line we have all of the negative response or negatively classified data, but there is a positive point also, in this negative classified area. So this means that point is an outlier**

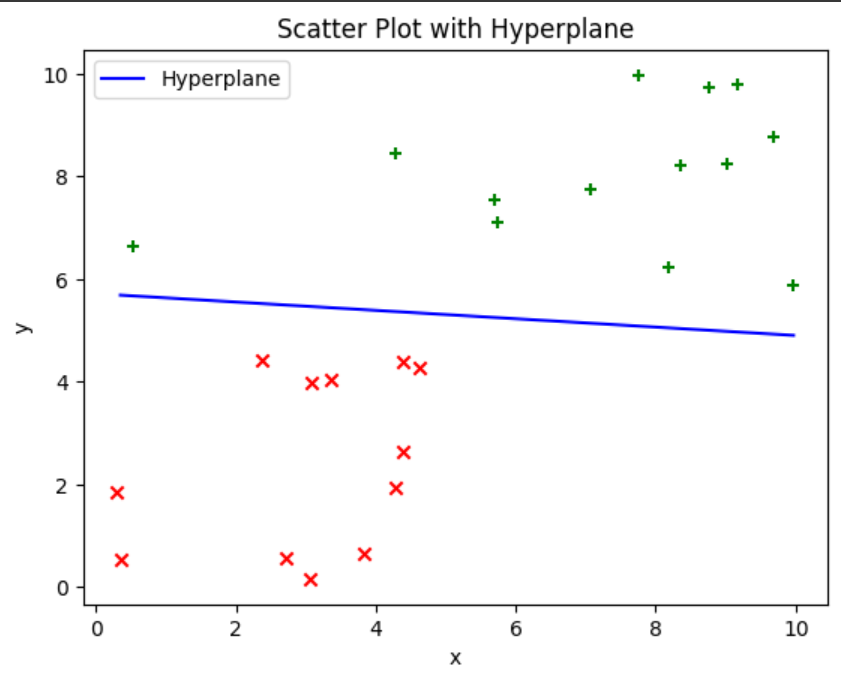
1. **Taking c=100**

****

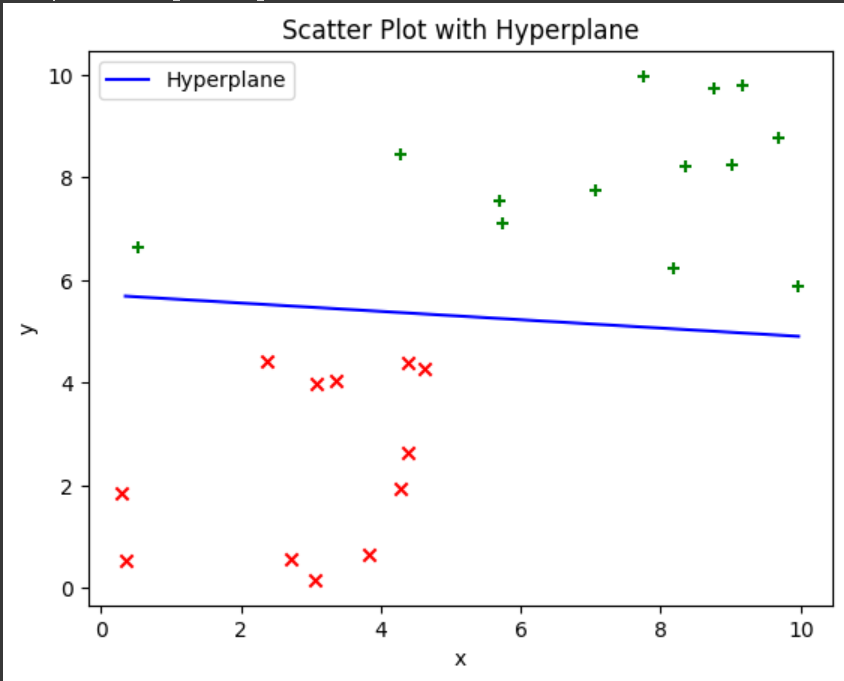
**Taking C=300**

****

**Taking C=700**

****

**Taking c=1000**

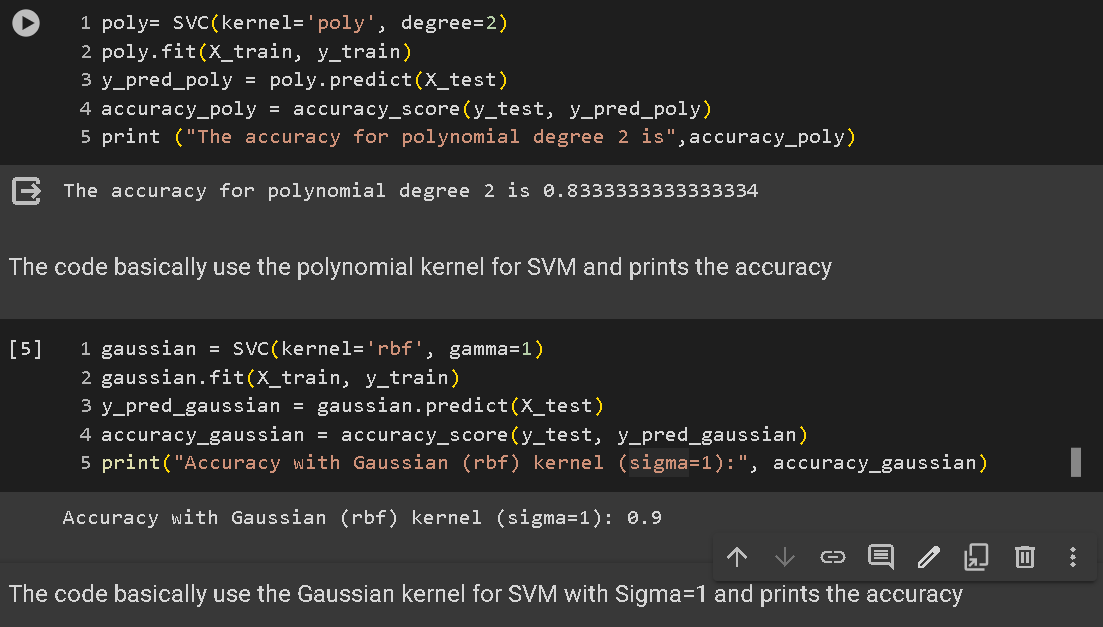
****

**The code for all these is written in Colab, and because the data is completely separate able, varying C doesn’t create any significant change in the hyperplane equation.**

**Question 3**

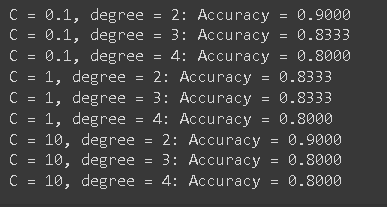
[**https://colab.research.google.com/drive/1anJ-Oe\_v\_oSSh8WoKKyWZKJvG8NnhT4V?usp=sharing**](https://colab.research.google.com/drive/1anJ-Oe_v_oSSh8WoKKyWZKJvG8NnhT4V?usp=sharing)

1. **The accuracy of the gamma with sigma=1 is a little bit more than that of polynomial with degree 2.**

****

1. **I took 3 different values of c which were 0.1, 1, 10**

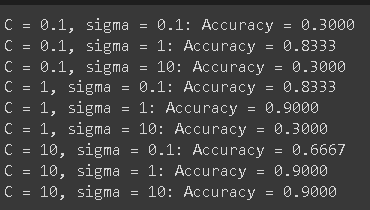
**And same goes for degree with values 2,3 and 4 so got the following outcomes**

****

**The experiment involved training Support Vector Machine (SVM) models with polynomial kernels on the Iris dataset, varying both the regularization parameter (( C )) and the degree of the polynomial. Three different values of ( C ) (0.1, 1, and 10) and three different degrees (2, 3, and 4) were considered, resulting in nine combinations. The models were evaluated using an 80/20 split, with accuracy scores recorded for each combination. The results showed that the choice of ( C ) and polynomial degree significantly influenced the model's performance. Lower values of ( C ) (0.1) tended to produce higher accuracy when paired with lower polynomial degrees (2 or 3), indicating that these models were less prone to overfitting. Higher values of ( C ) (10) performed well with higher polynomial degrees, suggesting that stronger regularization allowed the model to handle more complex decision boundaries effectively. In summary, the optimal combinations for this dataset were found to be ( C = 0.1 ) with degree 2, or ( C = 10 ) with degree 2, both achieving an accuracy of 0.9. These findings demonstrate the importance of parameter tuning in SVM models and highlight the trade-off between model complexity and generalization performance.**

**C\_values = [0.1, 1, 10]**

**sigma\_values = [0.1, 1, 10]**

****

**The experiment involved training Support Vector Machine (SVM) models with Gaussian kernels on the Iris dataset, varying both the regularization parameter ( C ) and the kernel parameter ( sigma ). Three different values of ( C ) (0.1, 1, and 10) and three different values of ( sigma ) (0.1, 1, and 10) were considered, resulting in nine combinations. The models were evaluated using an 80/20 split, with accuracy scores recorded for each combination. The results revealed that the choice of ( C ) and ( sigma ) significantly influenced the model's performance. Lower values of ( C ) (0.1) generally resulted in lower accuracy, while higher values of ( C ) (10) produced more consistent accuracy across different values of ( sigma ). However, the choice of ( sigma ) had a more pronounced effect on the accuracy, with the optimal performance consistently achieved with ( sigma = 1 ). This indicates that the width of the Gaussian kernel has a critical impact on the model's ability to generalize to unseen data. In summary, the optimal combination for this dataset was found to be ( C = 1 ) and ( sigma = 1 ), achieving the highest accuracy of 0.9. These findings underscore the importance of parameter tuning in SVM models and highlight the trade-off between model complexity and generalization performance, particularly in the context of Gaussian kernels.**

**Question 4:**