Introduction to Machine Learning - Major 2, Report

Omer Simhi - 316572593, Chen Peretz - 204219638

June 6, 2021

Part 1 - Data preparation

Q1. We didn't use the validation and test set for the data preparation process. The validation set is mainly for tuning and fixing the data here and there (after a training phase), and thus preparing based on this data isn't correct. The test set must be completly clean (and in general, in real life, is even unknown during the phase of data preparation) so we didn't prepar it's data too (actually, the test set we created in the previous hw assignment is no longer relevant for this assignment).

Part 3 - Classification

k-NN Baseline

Q2. The Graphs for the train and validation sets as function of the k for KNN model for each task:

Figure 1: Virus - Train and Validation accuracies

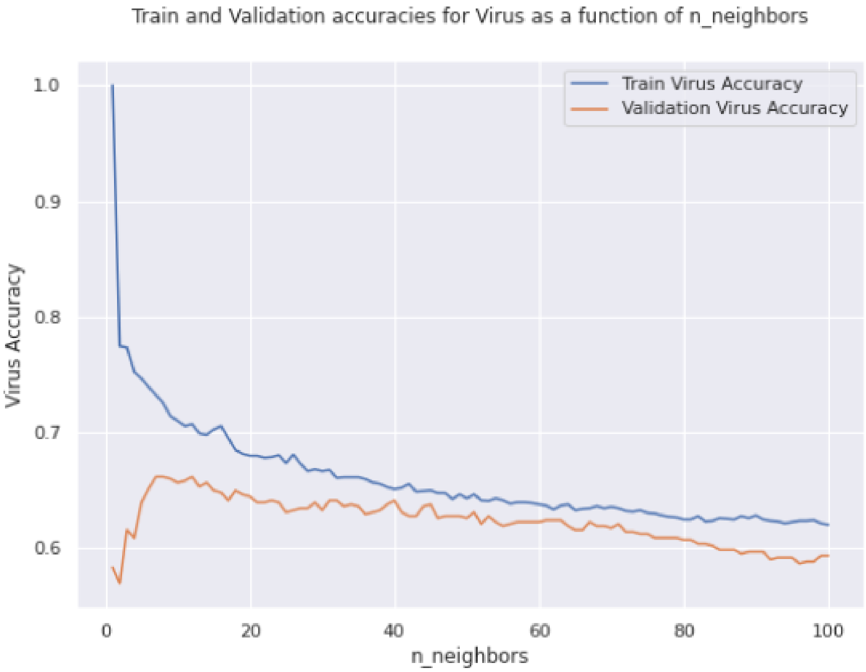


Figure 2: Risk - Train and Validation accuracies

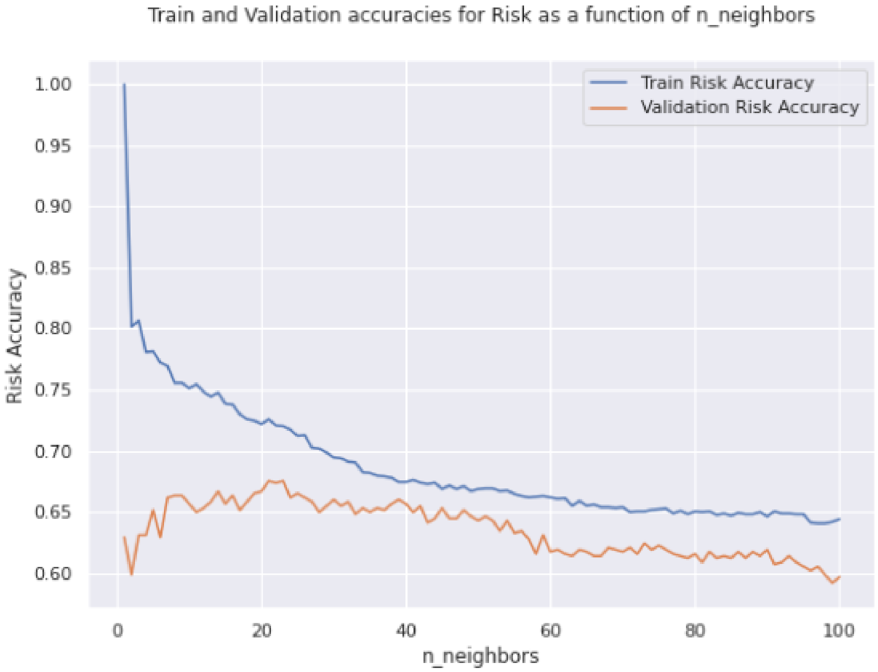
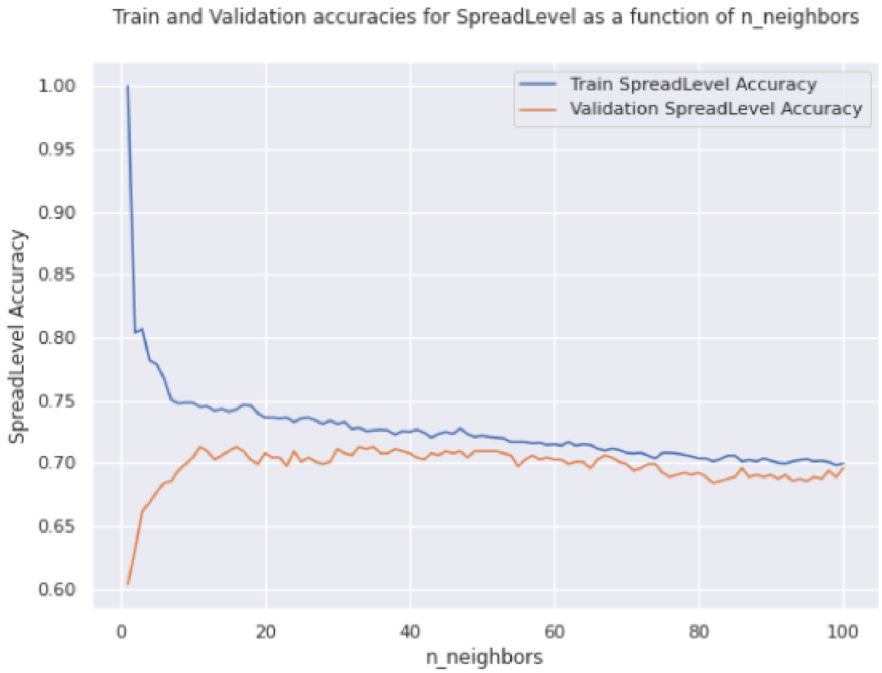
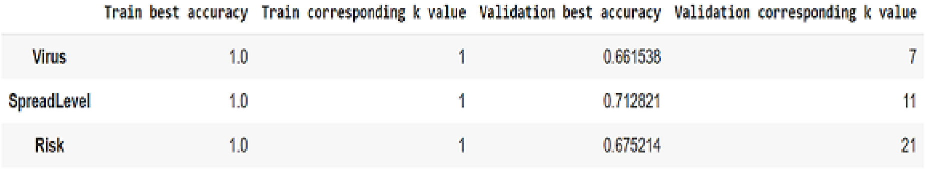


Figure 3: SpreadLevel - Train and Validation accuracies



Q3. Best accuracies for train and validation sets for each of the three tasks (Virus, Risk and SpreadLevel) in the KNN model:

Figure 4: Best Train and validation Accuracies - KNN



	Train best accuracy	Train corresponding k value	Validation best accuracy	Validation corresponding k value
Virus	1.0	1	0.661538	7
SpreadLevel	1.0	1	0.712821	11
Risk	1.0	1	0.675214	21

Q4. We will explain the accuracies graphs behavior for both train and validation sets and for each task, separately:

(a) Train set -

i. Virus - The tend is clear - the accuracy decreases as k increases - the best accuracy achieved for $k = 1$ (100%) while from $k \approx 60$ and afterwards we get the lowest accuracy ($\approx 63\%$). This behavior can be explained due to various reasons that discussed in the lecture - Large neighborhoods (large k's) can cause inaccurate estimation and also large bias while small neighborhoods (small k's) unreliable or uninformative and thus we see a very steep decline for small k's.

   ii. Risk - The tend is similar to the Virus's tend, except for the fact that the descent is slightly less steep, and that for large k's, there is a stabilization on less poor accuracy (around 65% accuracy for the lowest accuracy value). Other than that, similar explanations.

   iii. SpreadLevel - The tend is similar to the Virus's and the Risk's tend, except for the fact that the descent is less steep, and that for large k's, there is a stabilization on less poor accuracy (around 70% accuracy for the lowest accuracy value). Other than that, similar explanations.

(b) Validation set -

   i. Virus - The trend of accuracy is relatively stable - There is one significant increase for k increases from 1 to 10 (until we achieve $\approx$ 66% accuracy) and then slight decrease and stabilization for bigger k's. This behavior is caused because the trained model is overfit for large k's ($k > 11$).

   ii. Risk - The tend is similar to the Virus's tend, except the Risk tend is less "smooth" and then the "sweet spot" achieved for k ($k = 21$) and that the decrease lasts longer and stabilizes for bigger k ($k \approx 80$). Other than that, similar explanations.

   iii. SpreadLevel - The tend is similar to the Virus's and Risk tend, except the spreadLevel achieve higher accuracy at it's pick ($\approx$ 71%) and stabilizes faster (from $k > 20$ and onwards) and at a better accuracy. Other than that, similar explanations.

Q5. The best k values are the validation's best values. The reason is that the validation set purpose is to validate our training results. The accuracies we get for the validation set after training with the train set demonstrates the the relation between k and the accuracy we achieve in the validation process. For finding these values we run iteratively for all k's, train the model with this k and then predict using the validation set and save all the accuracies obtained. After that, just extract the values that correspond to the best accuracies for each task - results in Q3 in the two validation columns.

Decision Trees:

Q6. The Graphs for the train and validation sets as function of the tree depth for decision tree model for each task:

Figure 5: Virus - Train and Validation accuracies

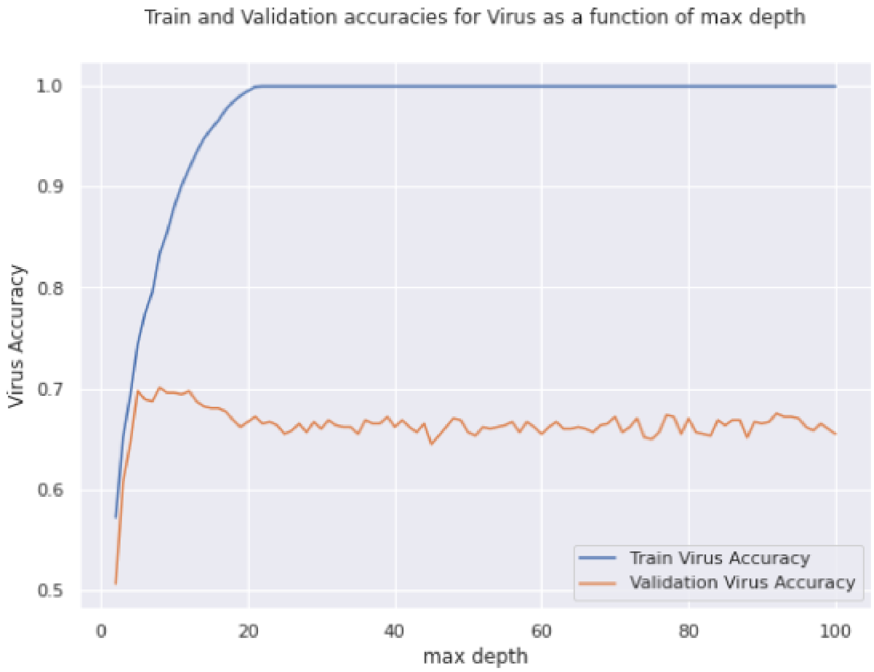


Figure 6: Risk - Train and Validation accuracies

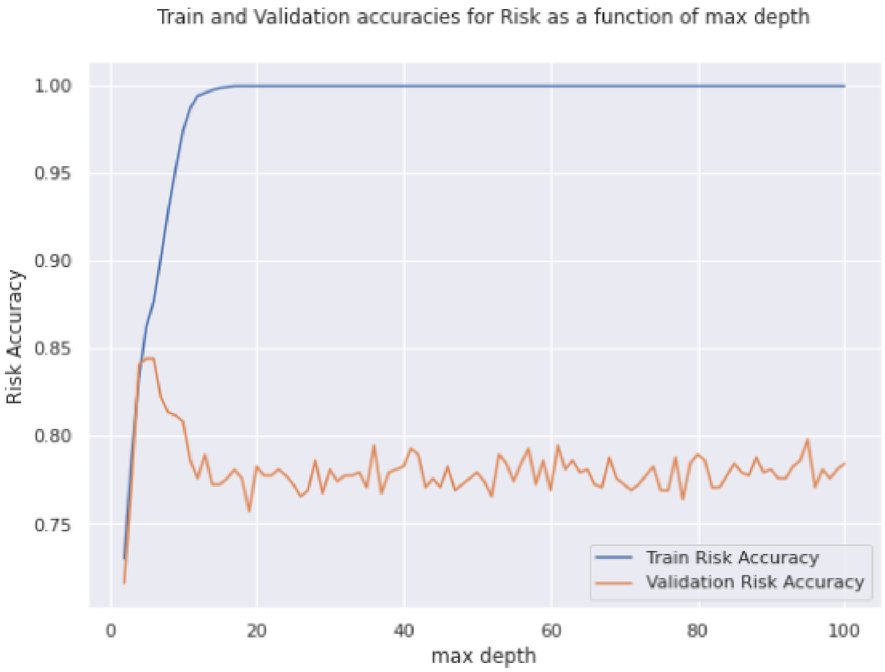
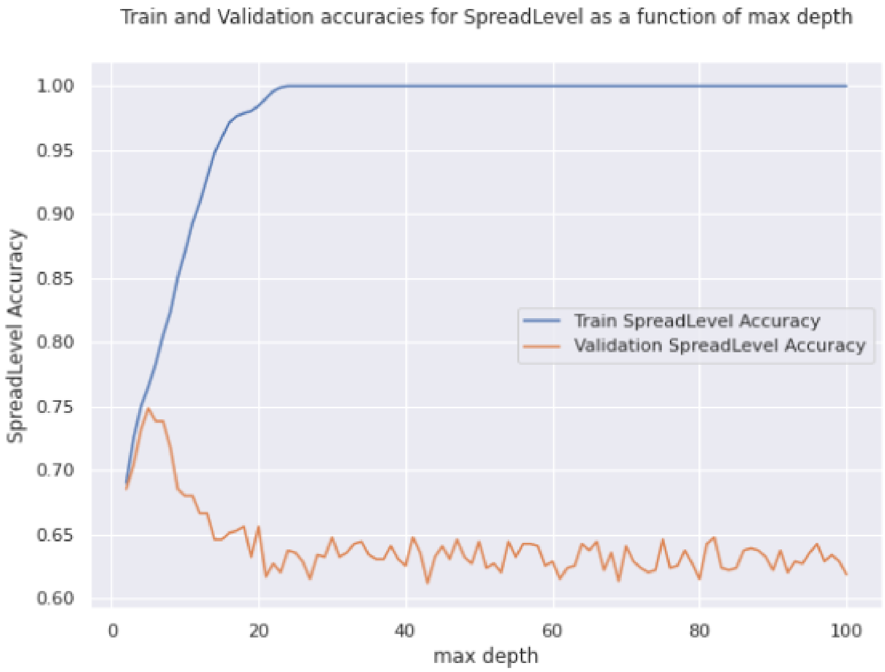
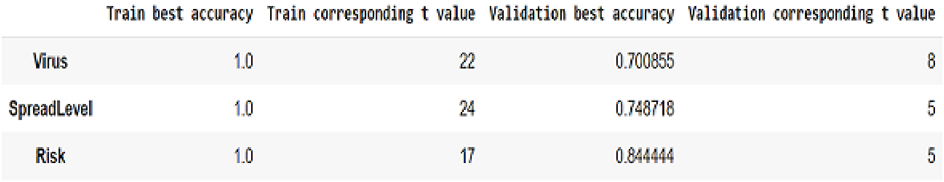


Figure 7: SpreadLevel - Train and Validation accuracies



Q7. Best accuracies for train and validation sets for each of the three tasks (Virus, Risk and SpreadLevel) in the decision tree model:

Figure 8: Best Train and validation Accuracies - Decision Tree

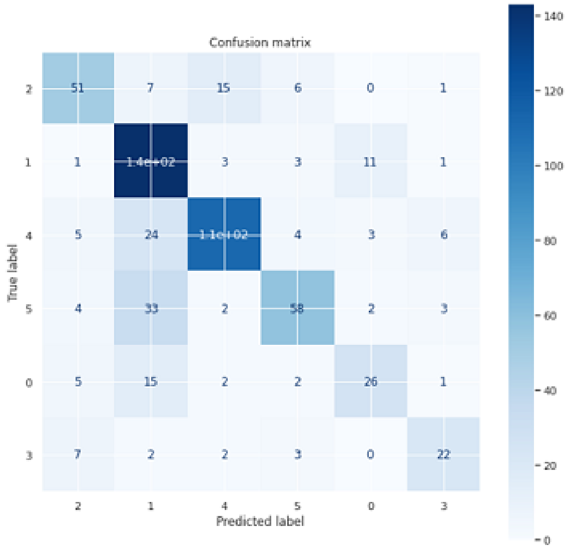


	Train best accuracy	Train corresponding t value	Validation best accuracy	Validation corresponding t value
Virus	1.0	22	0.700855	8
SpreadLevel	1.0	24	0.748718	5
Risk	1.0	17	0.844444	5

Q8. Plot of the confusion matrix for the Virus classification task on the validation set using the best performing model:

Figure 9: Confusion matrix for the Virus label

```
The confusion matrix for Virus classification task on the validation set for the best hyperparameter we achieved (maximal_depth = 8 ):
```



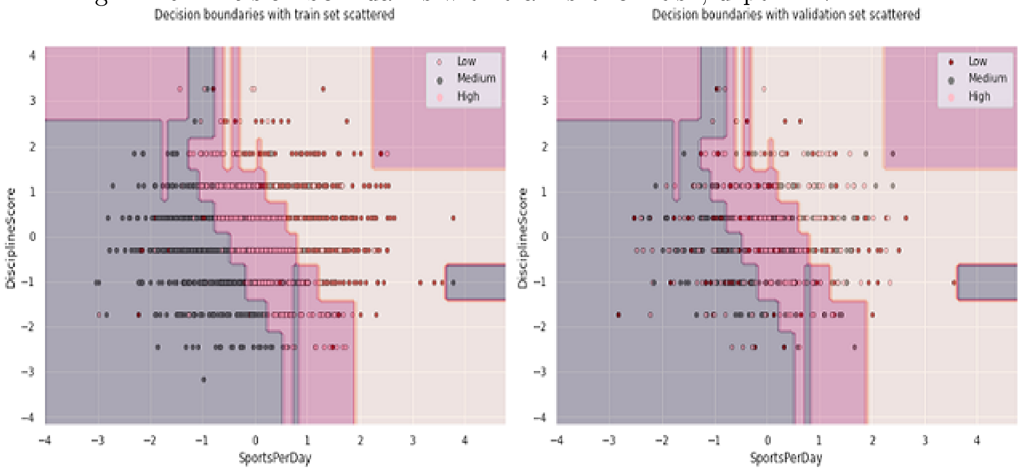
Confusion matrix

True label \ Predicted label	2	1	4	5	0	3
2	51	7	15	6	0	1
1	1	1.4e+02	3	3	11	1
4	5	24	1.1e+02	4	3	6
5	4	33	2	58	2	3
0	5	15	2	2	26	1
3	7	2	2	3	0	22

Q9. Tree graph for the Risk classification with maximal depth of $t = 7$ - **SEE ATTACHMENT IN ZIP.** We see from the result tree the the two most important features in predicting Risk level are **SportsPerDay** and **DisciplineScore.** To see that from the graph, we notice that these two features are the highest decision nodes in the graph.

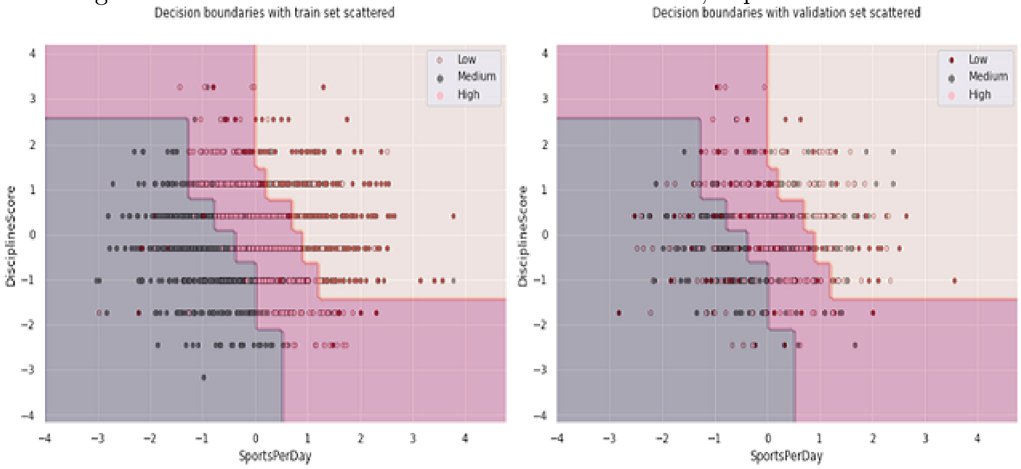
Q10. Plot of the decision boundaries for the validation and train sets with the two most important features from the previous question:

Figure 10: Decision boundaries with train set for Risk, depth = 7



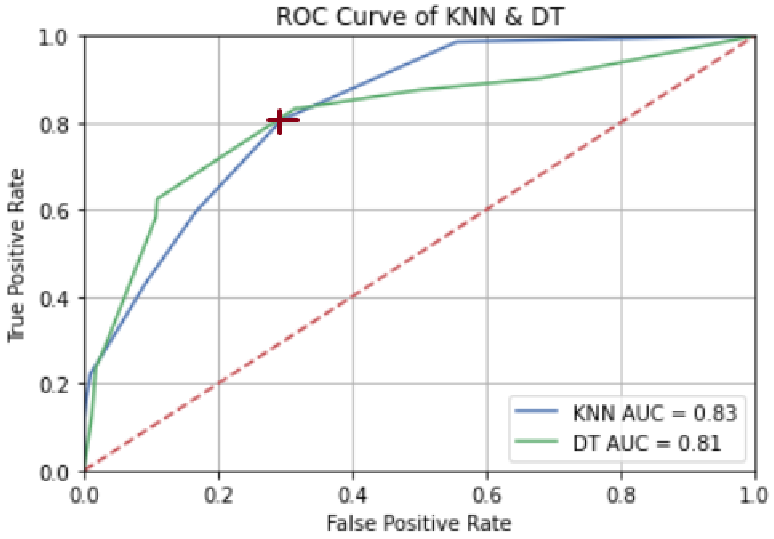
We see that for depth $t = 7$, in order to separate correctly the different classified points, more then 3 (the number of risk levels) areas are required and thus the model is **overfit** for this parameter. To see that its overfit graphically, here is the plots for depth $t = 5$:

Figure 11: Decision boundaries with train set for Risk, depth = 5



Q11. Plot of the ROC curve of the kNN and decision tree models, using the validation set, for SpreadLevel prediction (high spread = positive prediction, medium and low spread = negative prediction):

Figure 12: ROC curve for kNN and Decision Tree models



Q12. We will choose the kNN model for the spread level classification for the specific offset of the given question (where high spread = positive prediction, and medium and low spread = negative prediction). As we can see fron the above graph, the kNN model achieves larger area then the Decision Tree model. We believe that the optimal TPR and FPR tradeoff achieved at the point indicated in the diagram above (with ” + ” sign): $TruePositiveRate \approx 81\%$ and $FalsePositiveRate \approx 28\%$. The "risk" in the scenario of the false positive, is that we report a high spread level even though this is not the case in reality. This risk is insignificant as in total it will cause an unnecessary precaution.

Support Vector Machine

Q13. The Graphs for the train and validation sets as a function of the hyperparameter C:

Figure 13: Virus - Train and Validation accuracies

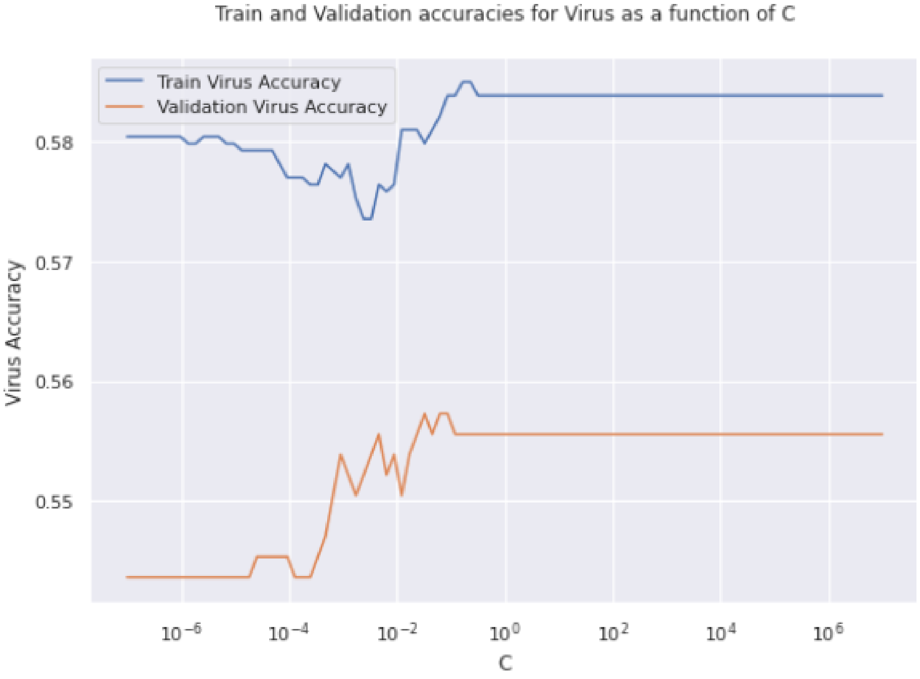


Figure 14: Risk - Train and Validation accuracies

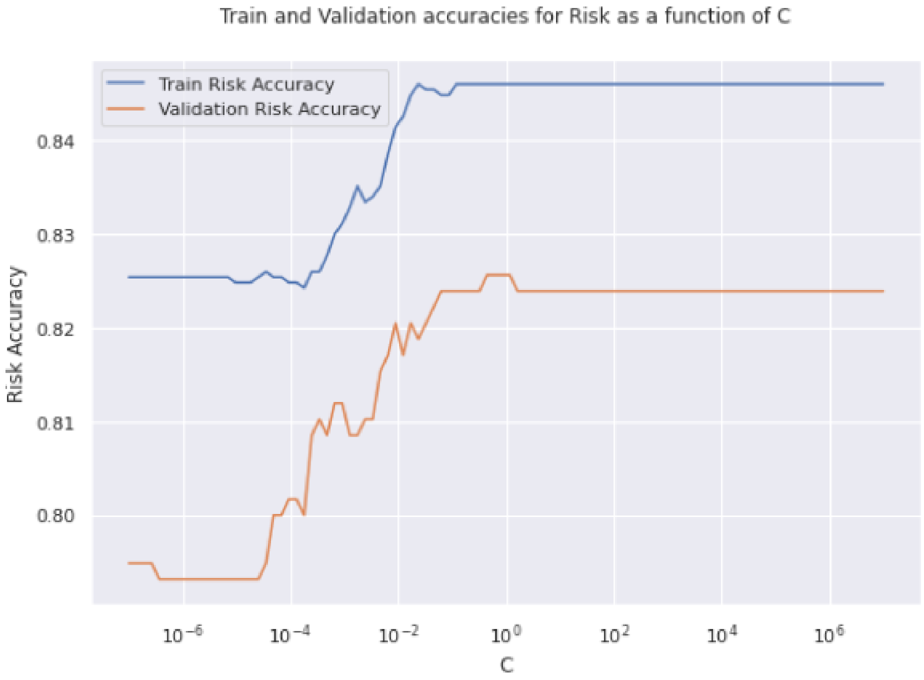
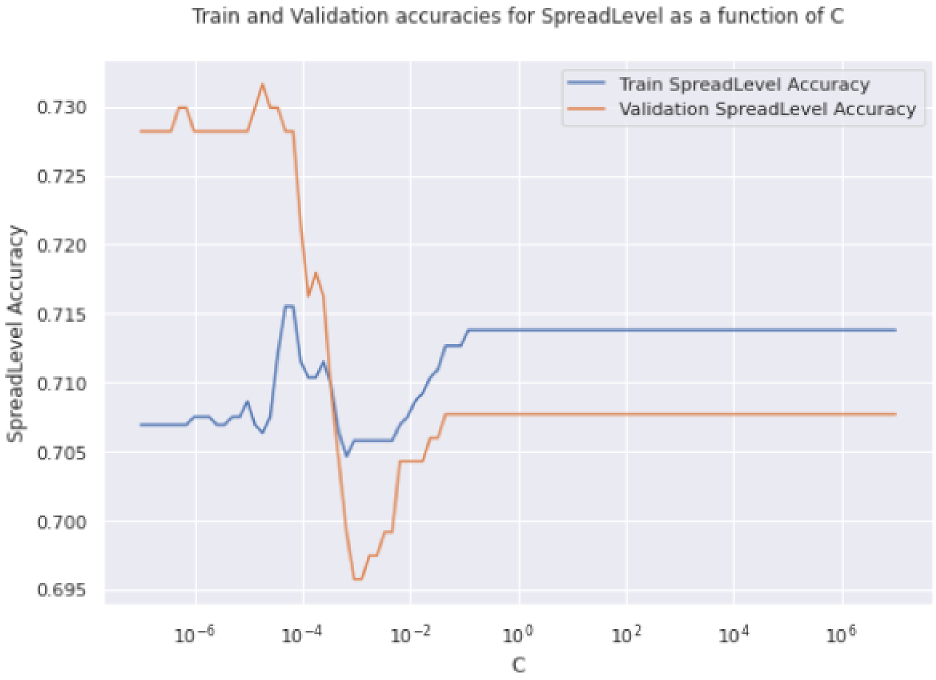


Figure 15: SpreadLevel - Train and Validation accuracies



Same grpahs, but now “zoom in” to the region where the best predictions occur:

Figure 16: Virus - Train and Validation accuracies

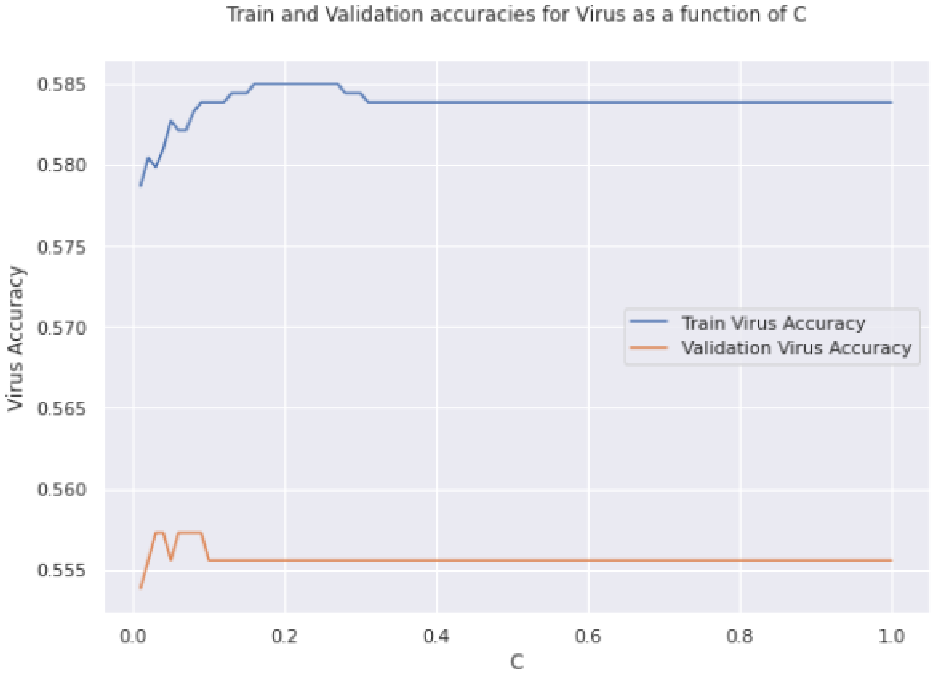


Figure 17: Risk - Train and Validation accuracies

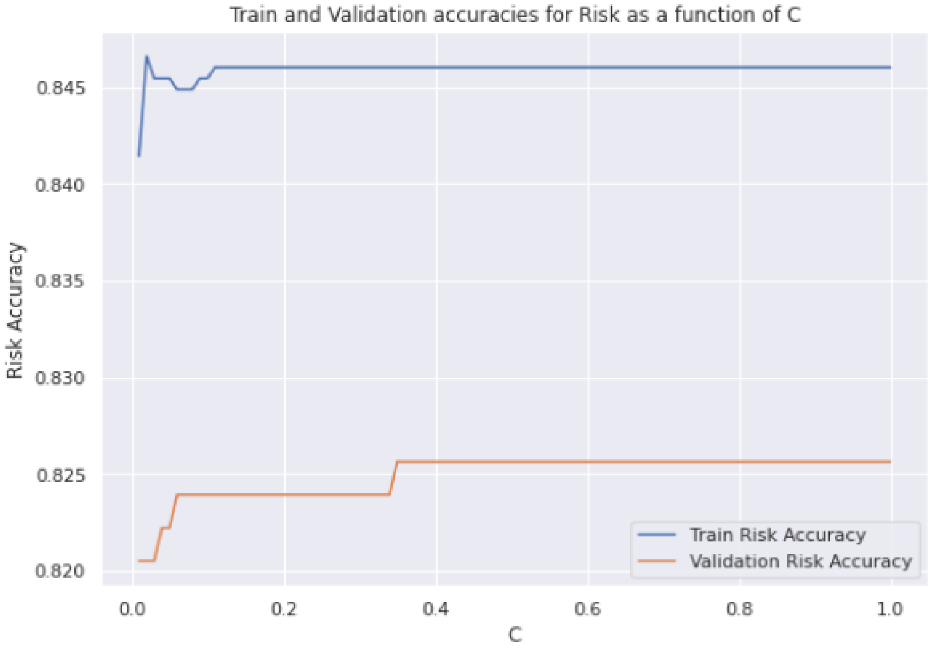
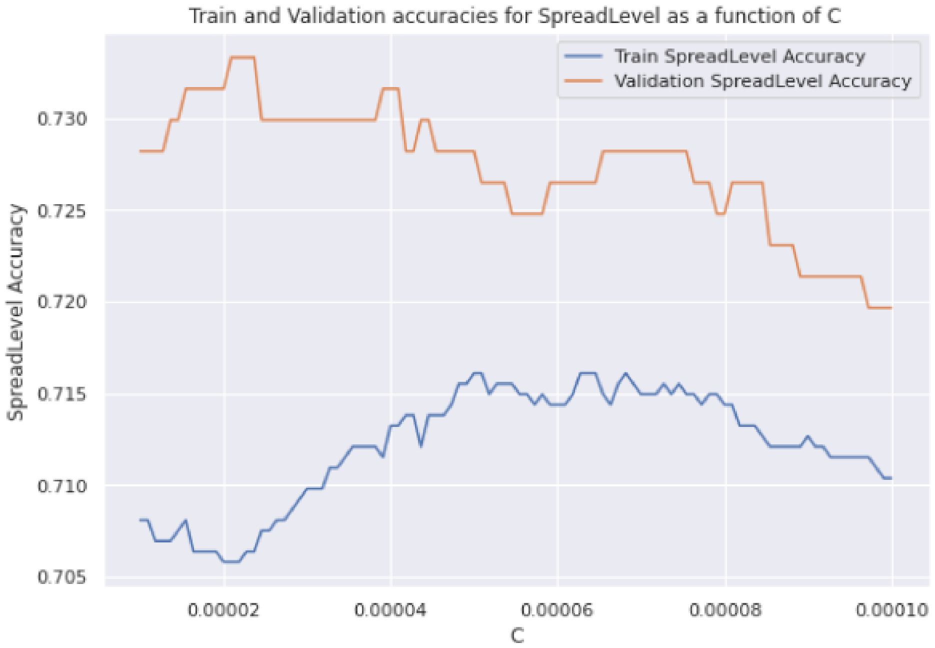
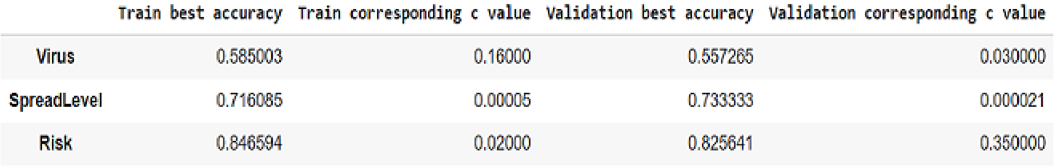


Figure 18: SpreadLevel - Train and Validation accuracies



Q14. The train and validation accuracies of your best SVM model for each task:

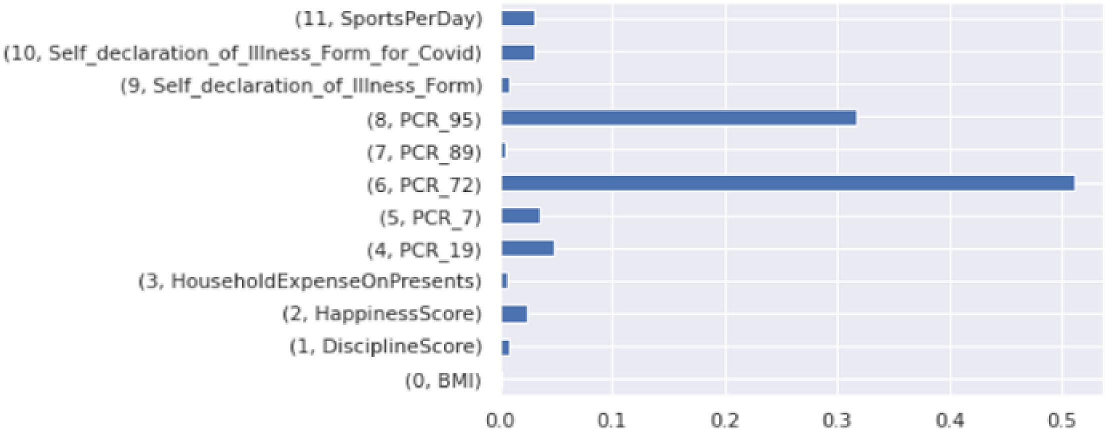
Figure 19: Best Train and validation Accuracies - SVM



	Train best accuracy	Train corresponding c value	Validation best accuracy	Validation corresponding c value
Virus	0.585003	0.16000	0.557265	0.030000
SpreadLevel	0.716085	0.00005	0.733333	0.000021
Risk	0.846594	0.02000	0.825641	0.350000

Q15. The features that are most dominant for the “flue” (flu) class in the current feature space are shown in the diagram below:

Figure 20: Most dominant features



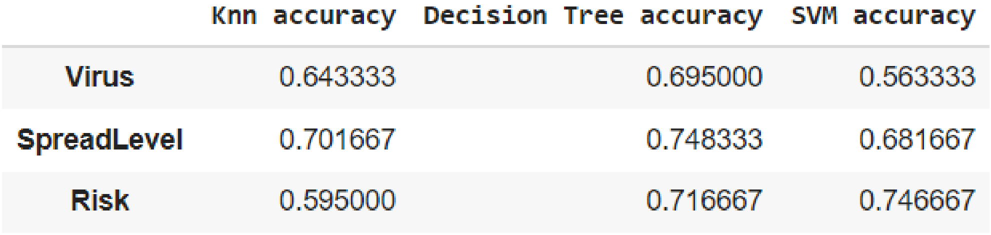
As we can see, the two most dominant features are PCR_95 and PCR_72.

Q16. Are the different spread levels are **not** linearly separable. As we saw in Q14, this SVM achieves ≈ 73% and not a value close enough to 100% and thus it cannot be linearly separated.

Part 4 –Testing your models

Q17. Test accuracies for each of the models and for each classification task. Model types to test: kNN, decision tree, linear SVM:

Figure 21: Test accuracies for each of the models



	Knn accuracy	Decision Tree accuracy	SVM accuracy
Virus	0.643333	0.695000	0.563333
SpreadLevel	0.701667	0.748333	0.681667
Risk	0.595000	0.716667	0.746667

Q18. Best performed models for each of the 3 tasks:

(a) Virus - The model that preforme best in the **test phase** is the **Decision Tree** model (≈ 69.5% accuracy). Similarly, the model that preforme best in the **validation phase** is also the **Decision Tree** model (≈ 70% accuracy).

(b) Risk - The model that preforme best in the **test phase** is the **SVM** model (≈ 74.6% accuracy). The model that preforme best in the **validation phase** is also the **Decision Tree** model (≈ 84% accuracy).

(c) SpreadLevel - The model that preforme best in the **test phase** is the **Decision Tree** model (≈ 74.8% accuracy). Similarly, the model that preforme best in the **validation phase** is also the **Decision Tree** model (≈ 74.6% accuracy).

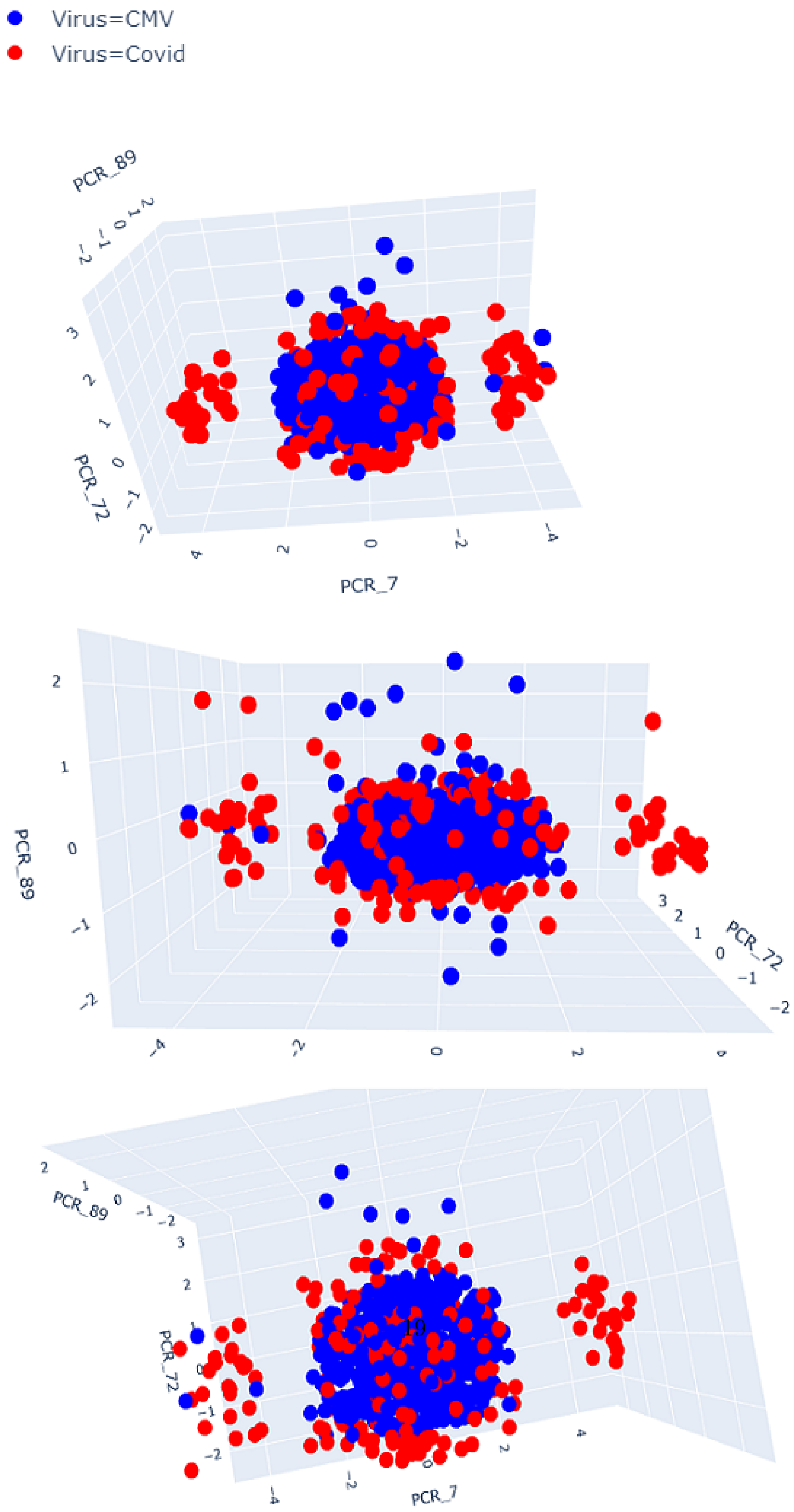
If model preforms well on the validation set but not on the test set, it means that the hyperparameters that were chosen for the validation phase, are fitting to the validation set but not to a general data set. This is indeed overfitting (when we are making decisions based on existing data set in order to get a good score in the learning phase instead of trying to generalize the process).

Q19. It is important that we do not tune our model using the test set because in real life, this set is unknown during the phase of data preparation and training and its purpose is to objectively examine the model we have trained and validate.

Part 5 –Non-Linear SVM

Q20. Plots of the relationship between the **Covid** and **CMV** viruses:

Figure 22: **covid** and **cmv** viruses plots



We see that the shape obtained is a concentration of the points in the form of an ellipse representing the CMV virus. Around the above concentration, the dots representing the covid virus appear in a way that is relatively separate from the CMV virus.

Q21. The kernels we tried are **poly**, **rbf** and and the **sigmoid** kernels. For each of the kernels, we tuned the hyperparameter's range and then checked the accuracies obtained. Out of the three kernels, **rbf** achieved the best results - while the **poly** and **sigmoid** kernels achived around 65% accuracy **rbf** achived almost 74% accuracy. Despite this, we were unable to make the data linearly separable, as accuracy does not yet strive for 100% accuracy.

Q22. Train and validation accuracies as a function of the hyperparameters using heatmaps:

Figure 23: Train accuracies

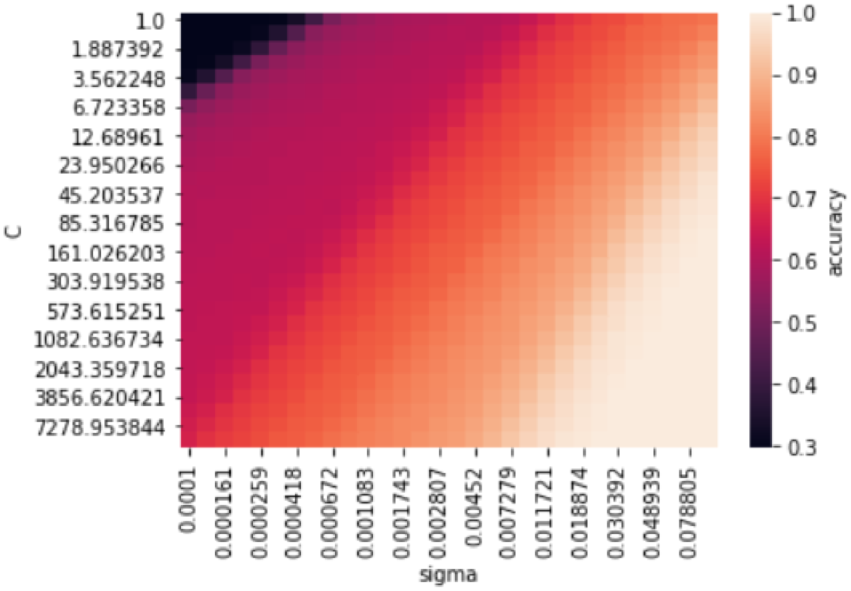
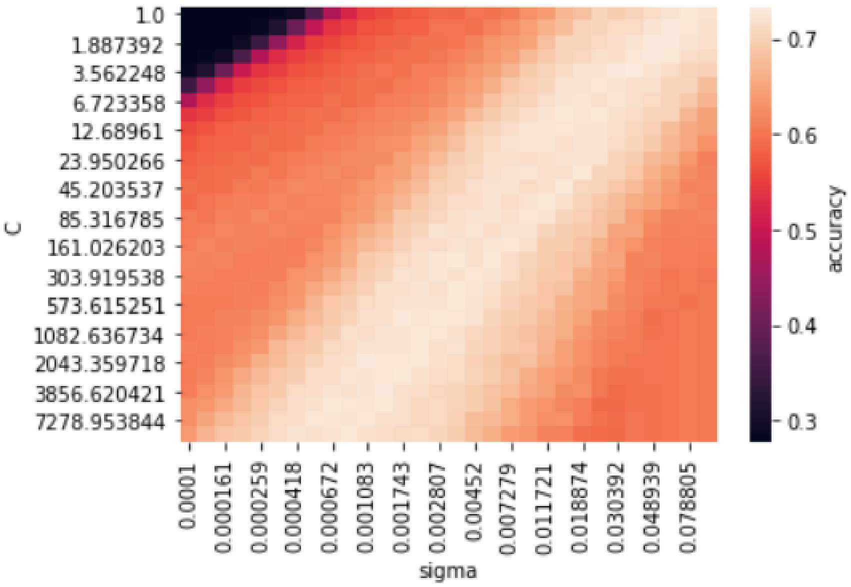


Figure 24: Validation accuracies



The ranges we taken from Q13 as the ranges that provide the best accuracies according to the results in that section.

Q23. Training and validation accuracies for the best non-linear SVM model - **rbf** with the best hyperparameters from the previous question:

Figure 25: Table of best Virus accuracies - **rbf model**

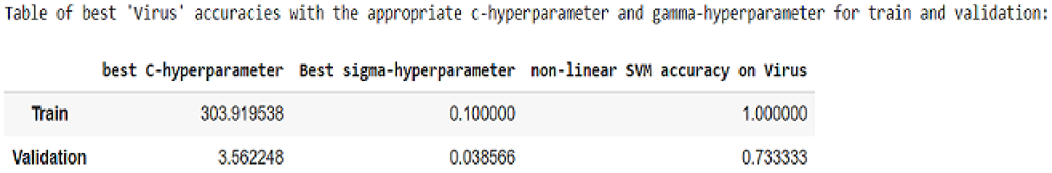


Table of best 'Virus' accuracies with the appropriate c-hyperparameter and gamma-hyperparameter for train and validation:

	best C-hyperparameter	Best sigma-hyperparameter	non-linear SVM accuracy on Virus
Train	303.919538	0.100000	1.000000
Validation	3.562248	0.038566	0.733333