HW2 Report

Part 1

Q1:

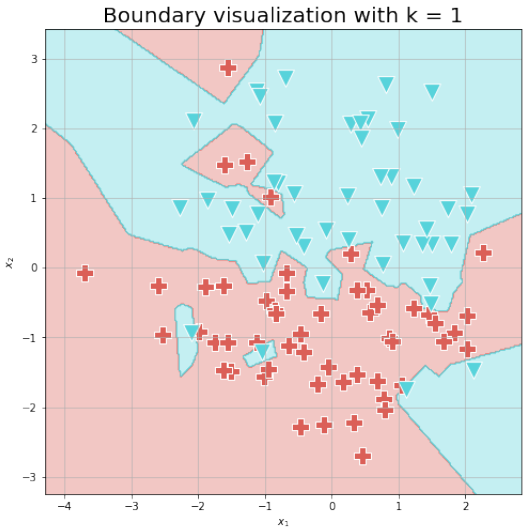

Figure 1: Boundary Visualization for KNN with K = 1

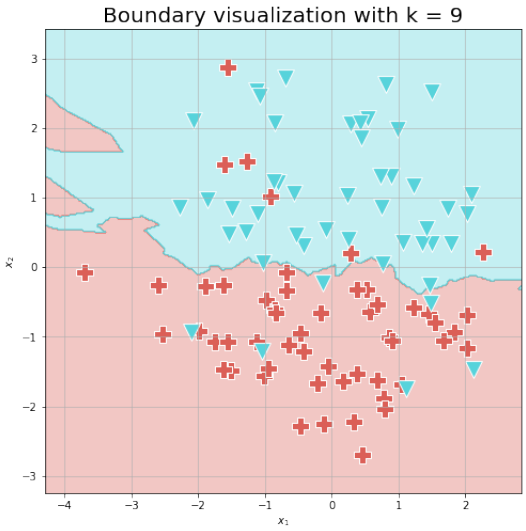

Figure 2: Boundary Visualization for KNN with K = 9

As can be seen in figure 1, the model with k equal to 1 caused overfitting to occur, as for every new sample point, its prediction took into account only the closest point to it, thereby increasing the complexity of the model since every new point is predicted locally. On the other hand, as can be seen in figure 2, the model with k equal to nine takes into account the nine closest points to the sample being predicted, and therefore takes a broader outlook on the training data in-order to tdo the prediction, thereby lowering the complexity. But at the same time, we can see that a number of points were not classified correctly, and therefore this could indicate that this model is performing underfitting.

Q2:

	spread
spread	1.000000
PCR_10	0.212723
PCR_07	0.042114
blood_A_AB	0.037510
PCR_01	0.022668
PCR_06	0.020925
covid	0.014039
PCR_02	0.013176
num_of_siblings	0.010719
household_income	0.008204
PCR_03	0.003534

Figure 3: The 10 most correlated features to `spread`

Q3:

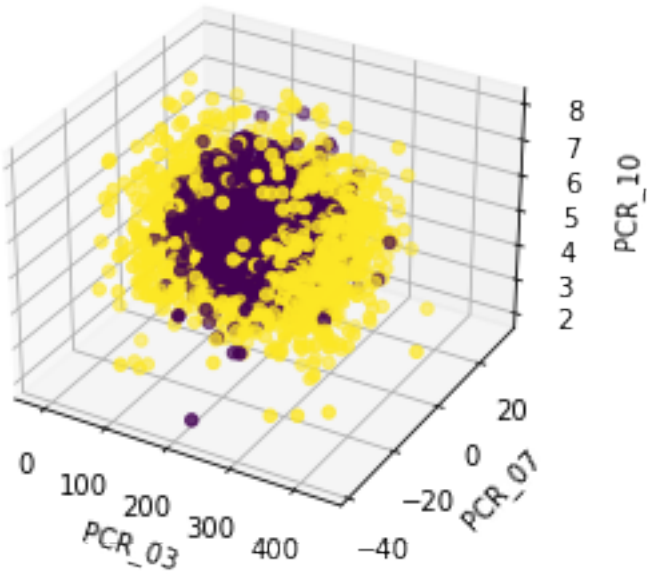

Figure 4: 3D Scatter plot of `PCR_03`, `PCR_07`, and `PCR_10` according to `spread`

Q4:

The points with label -1 for `spread` are contained within an ellipsoid described by the following equation:

$$\frac{PCR\_03}{a^2} + \frac{PCR\_07}{b^2} + \frac{PCR\_10}{c^2} = 1$$

where a, b, c are positive real numbers.

Q5:

The training accuracy for $k = 11$ is 81.333%.

Q6:

Z-score scaling scales the data of a feature by ensuring that they have zero mean and unit standard deviation, thereby causing the data to adhere to a normal distribution. Features scaled according to this technique have their outliers handled correctly, but no guarantee on the resulting range of the data is made, and the ranges of different features scaled according to Z-score may differ from each other. This technique is preferable in cases that have outliers and when the learning model assumes that the data adheres to a normal distribution . On-the-other-hand, the min-max technique involves scaling the data of a feature to a specific range (generally between 0 and 1). Contrary to the Z-score method, this technique guarantees a uniform range across features and maintains the original distribution of the data, but does not handle outliers well. Therefore, it would be preferable to use this technique only when the feature in question has no significant outliers and_or the learning model to be used requires the feature data to fall within a certain range.

Q7:

The updated accuracy after normalizing the data is 88.5416%. This is an increase of approximately 7% over the accuracy from the non-normalized data. This can be explained by the fact that KNN is an algorithm that uses euclidean distances between data points for its predictions. Therefore, it is very sensitive to differences in scales between different features, and thus features that have undergone normalization, which equalizes or nearly equalizes the scales and ranges of different features, will do better in KNN because no single feature will dominate and have more influence over other features by way of having a larger scale.

Q8:

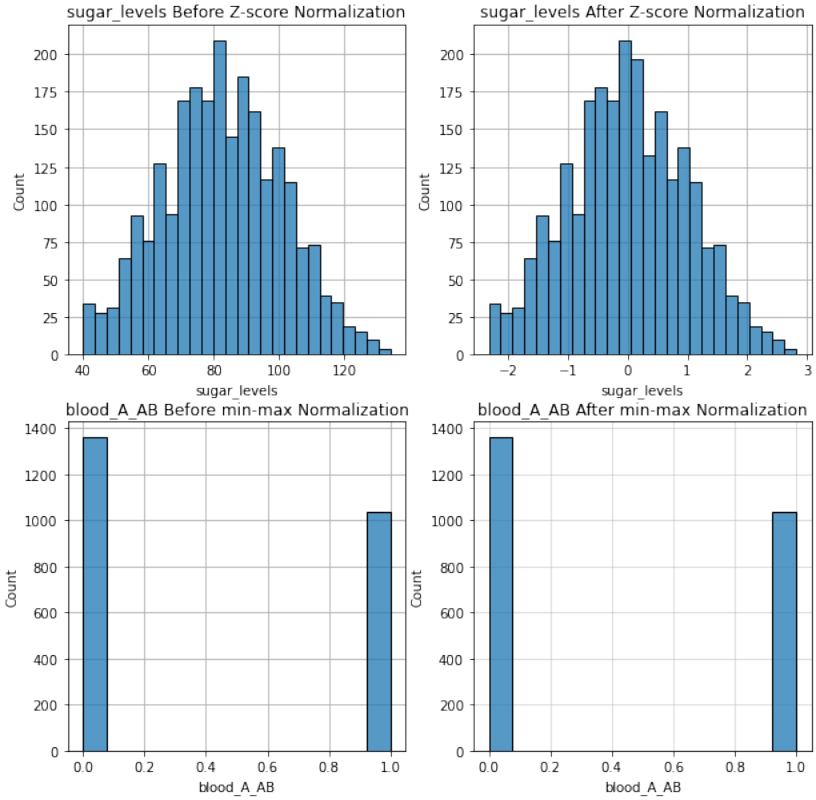

Figure 5: Histogram Plots of `sugar_levels` and `blood_A_AB` Before and After Normalization

`sugar_levels` We chose to do Z-score Normalization on `sugar_levels` since it already obeys an approximate normal distribution, as can be seen in figure 5, and therefore we would not be forcibly changing its distribution through the normalization process. Furthermore, we will not be using it in the KNN model, which works better with min-max normalized features, but rather with SGD SVM, which works better with Z-score normalized features, and a decision tree, which is mostly agnostic to Normalization.

`blood_A_AB` In contrast, we chose to do min-max normalization on `blood_A_AB`, since it does not obey a normal distribution, but rather already obeyed a binary distribution of 0 and 1, which we did not want to modify. Furthermore, it does not have outliers which could negatively affect the min-max normalization.

Q9:

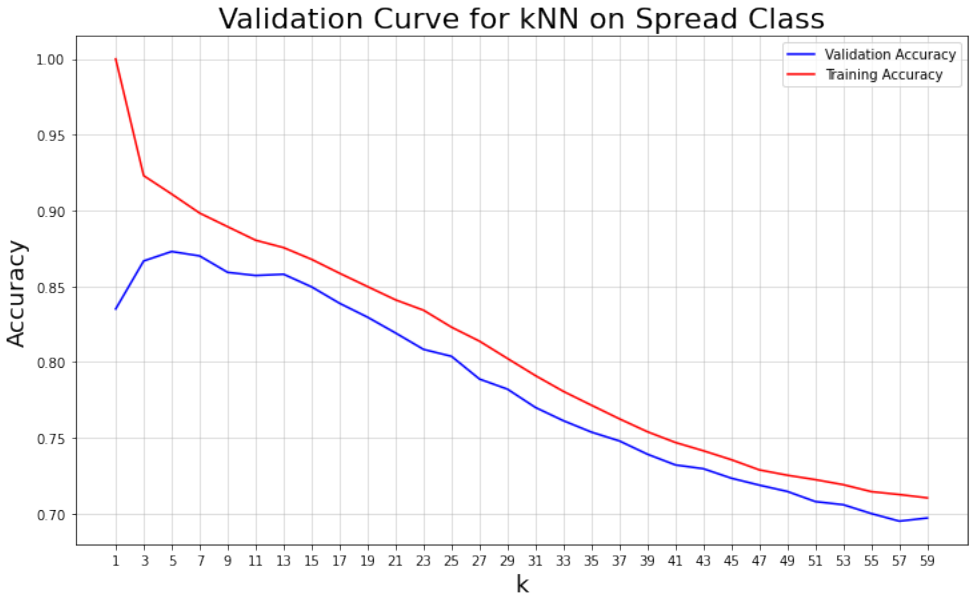

Figure 6: Validation Curve for kNN on spread Class

The optimal k value is 5, as can be seen by virtue of the fact that in figure 6, the highest point (peak) of the validation curve is at 5. The mean validation and training accuracies for $k = 5$ are 87.2916% and 91.083%, respectively.

Part 2

Q11:

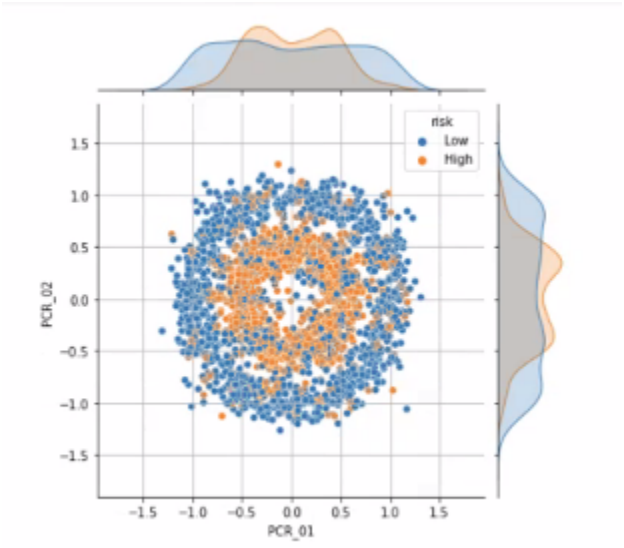

Figure 7: Joint plot of PCR_01 and PCR_02 according to risk

As can be seen from the scatter portion of the jointplot in figure 7, the plot is mostly separable into radiuses, and therefore it seems likely that `PCR_01` and `PCR_02` will be important in predicting the `risk` class.

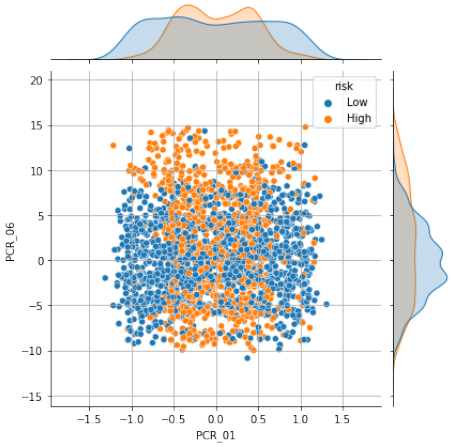

Figure 8: Joint plot of PCR_06 versus PCR_01 with respect to risk

Furthermore, from the scatter portion of the jointplot in figure 8 we noticed a mostly separable form similar to the letter "H" where the "H" itself is made up of a high proportion of points with low risk surrounded by clusters of points of high risk. Therefore, we can conclude that in addition, `PCR_06` will be important in predicting the `risk` class.

Q12:

	risk
risk	1.000000
sugar_levels	0.244313
PCR_06	0.175028
blood_A_AB	0.051560
PCR_07	0.047994
PCR_01	0.023978
PCR_02	0.021614
PCR_03	0.017089
PCR_10	0.013302
covid	0.012597
household_income	0.005975

Figure 9: The 10 most correlated features to `risk`

Q13:

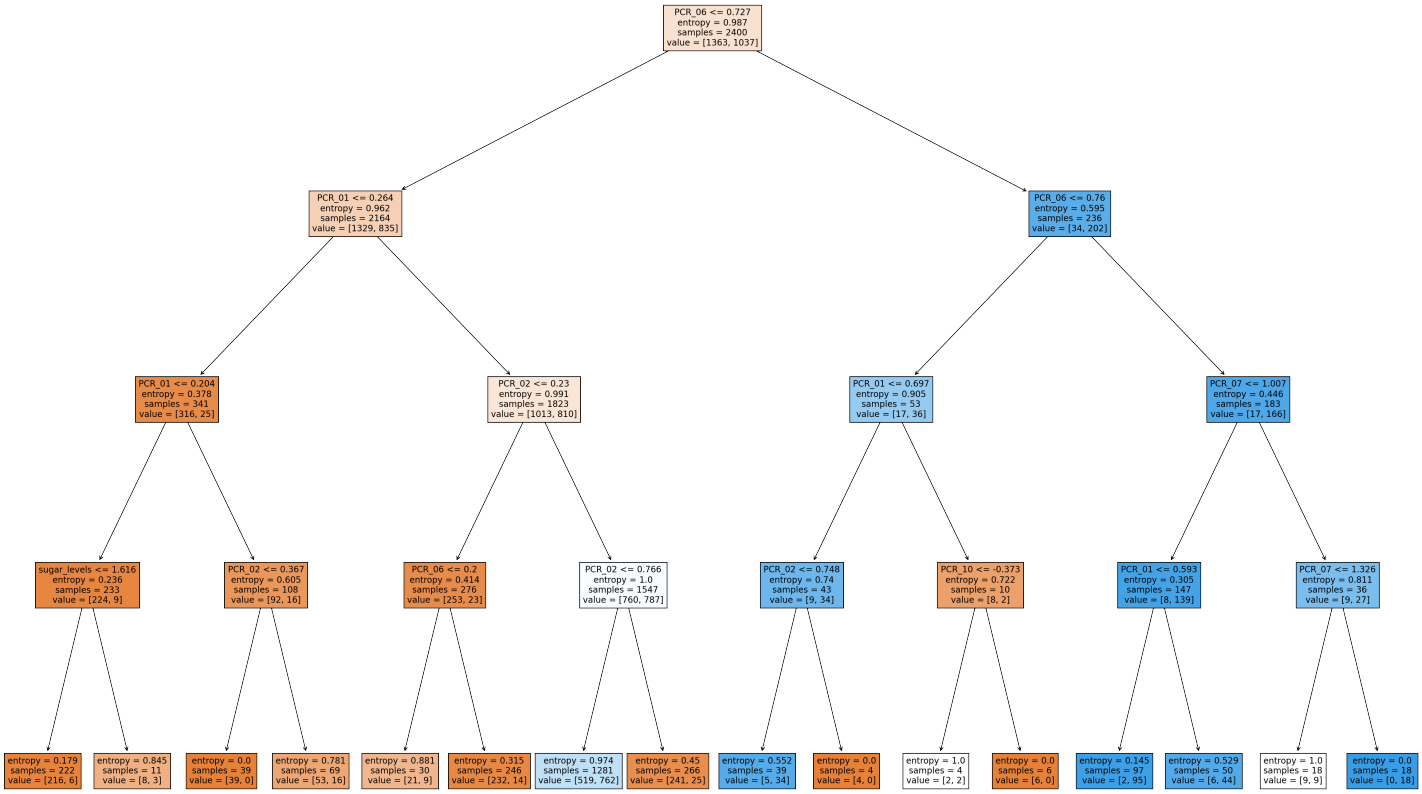

Figure 10: ID3 Decision Tree for Predicting `risk` CLass

The training accuracy of the model is 74.333%.

Q14:

`PCR_01` and `PCR_02`, which we concluded would be important features, are the features with the fifth and sixth highest correlations with **`risk`**, as can be seen in figure 9, and their actual correlation values (0.023 and 0.021 respectively) are very small. It is therefore easy to see that that these features are very much not discernable as important features by merely analyzing their correlation to **`risk`**. Even `PCR_06`, which has the second highest correlation to **`risk`**, has a low correlation value of only 0.175, and is therefore also not easily discernable as an important feature by way of correlation alone.

All three of these features which we deemed important were used by ID3 in the decision tree. And in-fact, they were used in 11 out of 15 of the internal nodes of the tree, further proving their importance.

Q15:

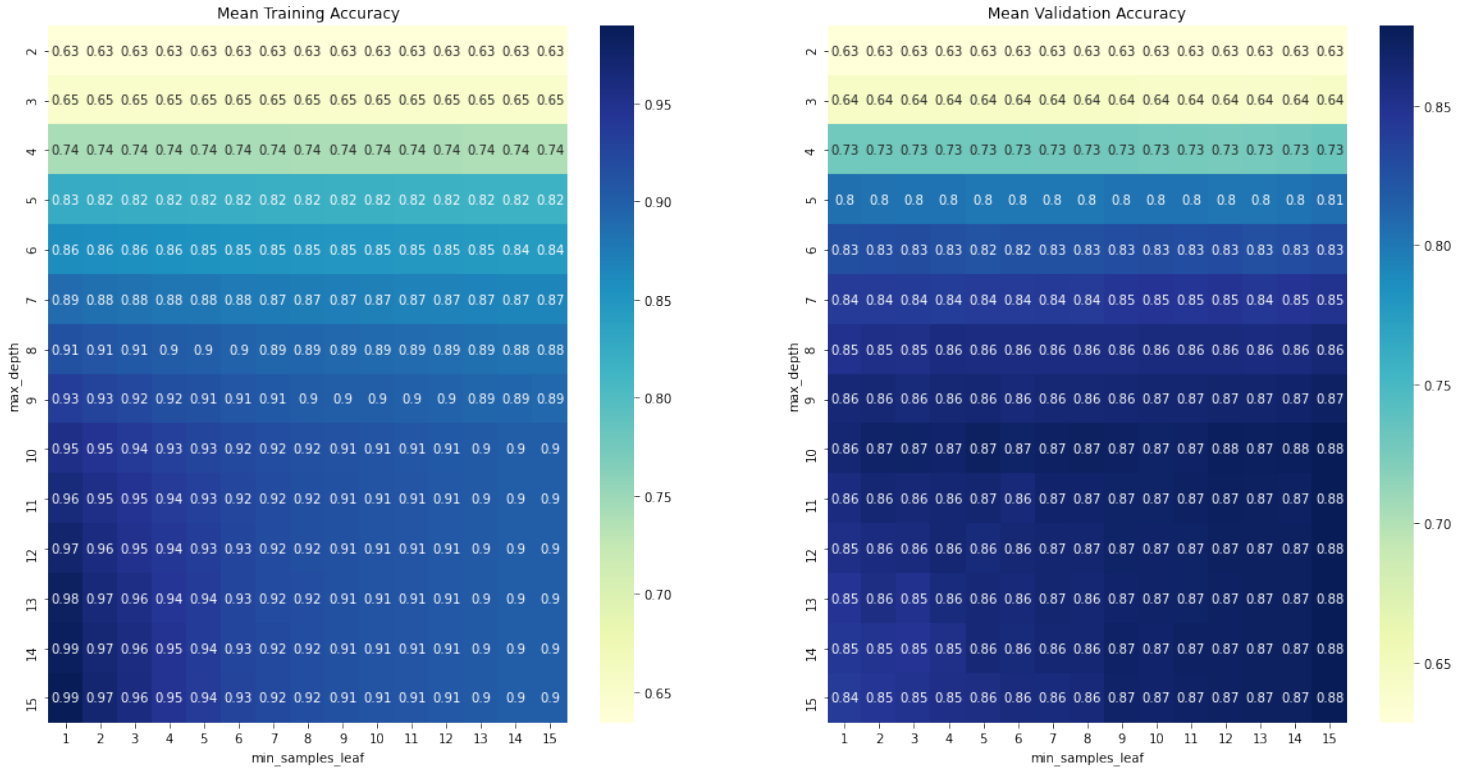

Figure 11: Heat Maps of Training and Validation Accuracies of ID3 on `risk` Class for Different Values of `max_depth` and `min_samples_leaf` Hyper-Parameters

The optimal hyperparameter combination is a `max_depth`=10 and `min_samples_leaf`=15, since they produce the highest validation accuracy of 88% with the minimal required depth and highest minimum sample leaf requirement, as can be seen in the mean validation accuracy heat map in figure 11.

The combination of `max_depth`=2 and `min_samples_leaf`=1 causes underfitting, as this leads to a low training accuracy of 63%.

The combination of `max_depth`=15 and `min_samples_leaf`=1 causes overfitting, as this leads to a high training accuracy of 99% versus a relatively low validation accuracy of 84%.

Part 3

Q20:

In Q19, we used a 2 dimensional polynomial transform on the features within the primal objective of the SVM model. This approach is slower than the two dimensional kernel objective because it involves computing inner products between data points and the w vector in high dimensions, whereas in the kernel case, the dual objective is used, which only involves computing the kernel function for 2 dimensional polynomials which is of much lower dimension (and finding and storing the αs, which is not too computationally burdensome on the assumption that the support vectors are sparse).