Question 1.1

For this problem two different algorithms have been implemented:

1. **Support Vector Machine (SVM):**

Working of SVM: In SVM an optimal separation is found between different classes. A street formed by the support vectors which are known as the gutters at the edge is used to perfectly separate the classes. If a new datapoint comes then depending on the position of the datapoint on left or right of the street that datapoint is classified accordingly. To consider the optimal separation, it is an optimization problem given as:

$1/2||w||$ where the ||w||- width of the street is replaced by $||w||^2$ subject to the condition that $y_i(w^T x_i + b) \geq 1 \; \forall i$ when y_i =+1 or -1

Then min max optimization is applied to obtain

$$\min_{w,b} \max_{\alpha \geq 0} \{\frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1]\}$$

but this has some major issues that are it does not solve the case when classes are not linearly separable, when there is noise and when some error is needed to be allowed in training data which is called Soft Margin.

In that case a Soft margin parameter C is used which means higher the C is more is the cost of using slack variables. That reduces the optimization problem as follows:

Minimize $v(w,b,\xi) = 1/2||w||^2 + c \sum_{i=1}^{n} \xi_i$

subject to $y_i(w^T x_i + b) \geq 1 \; \forall i \;\; and \; \xi_i > 0, \forall i$

When SVM has to be extended as non-linear learner then: the vectors x_i are transformed to a high dimension space using non-linear mapping $\phi(x_i)$ so that problem becomes linearly separable in a feature space. Computing the dot product $< \phi(x_i) \; \phi(x_j) >$ is computationally expensive so it is computed efficiently using $k(x_i, x_j)$. many common kernel functions like rbf, poly and sigmoid are used to compute that. The margin is maximized by minimizing the support vector. The decision function of classifier is then written as :

$$D(x) = \sum_{\forall x_{i \in S}} \alpha_i \lambda_i k(x_i, x) + \alpha_o$$

$\lambda_i = +1 \; or \; -1 \; which \; are \; the \; labels \; and \; \alpha_i$
$\geq 0 \; which \; is \; the \; parameter \; that \; is \; optimized \; using \; the \; optimization \; problem \; usually$

$$Quadratic \; Optimization$$

S is set of support vectors.

Why SVM was chosen:

SVM separates different classes by a hyperplane and produces a classifier that will work on unseen examples hence it generalizes well. The hyper parameters can be tuned in order to prevent overfitting.

The factor gamma in radial basis function(rbf) kernel for example is tuned such that if its too low value then it might cause overfitting and include all data points without capturing the shape.

2. **Random Forest:**

Working of RF: Random Forest is a meta estimator that trains on different number of decision tree classifiers on different samples of dataset. It is an Ensemble approach for classification. It uses averaging to improve the prediction accuracy and controls the Overfitting of the data to the classifier. In Random Forest, the trees are fully grown without pruning. Then the decision of the trees is fused which results in Random Forest.

Why is it chosen: Random Forest is mainly chosen here because it is an Ensemble Classifier and it avoids overfitting by improving the accuracy score while training on the dataset. In addition, it can handle high dimensional feature space and the performance is more robust than other classifiers.