**Machine Learning Assignment 20**

1. What is the underlying concept of Support Vector Machines?

Ans-)The underlying concept of Support Vector Machines (SVMs) is to find the best decision boundary (hyperplane) that separates two classes in a dataset with the largest possible margin between them. SVMs achieve this by mapping the input data to a high-dimensional feature space where the classes can be separated by a hyperplane and then finding the hyperplane that maximizes the margin between the classes.

2. What is the concept of a support vector?

Ans-)In SVMs, support vectors are the instances (data points) that lie closest to the decision boundary (hyperplane) and directly determine the location of the hyperplane. These instances have a non-zero value for the decision function and are crucial for determining the optimal hyperplane and maximum margin.

3. When using SVMs, why is it necessary to scale the inputs?

Ans-)It is necessary to scale the inputs when using SVMs because SVMs are sensitive to the scale of the features. If the features are not scaled, features with larger values will dominate the optimization process and lead to a suboptimal solution. Scaling the features to a similar range (e.g., using StandardScaler or MinMaxScaler) allows the SVM to find a better decision boundary.

4. When an SVM classifier classifies a case, can it output a confidence score? What about a

percentage chance?

Ans-)Yes, an SVM classifier can output a confidence score, which represents the distance between the test instance and the decision boundary. However, SVMs do not output a percentage chance. The confidence score is not a probability but rather a measure of how confidently the classifier assigns a class label to an instance.

5. Should you train a model on a training set with millions of instances and hundreds of features

using the primal or dual form of the SVM problem?

Ans-)When training a model on a large dataset with millions of instances and hundreds of features, it is generally better to use the dual form of the SVM problem, as the primal form may be computationally expensive and memory-intensive. The dual form can efficiently solve the problem even with a large number of features.

6. Let’s say you’ve used an RBF kernel to train an SVM classifier, but it appears to underfit the training collection. Is it better to raise or lower (gamma)? What about the letter C?

Ans-)If an SVM classifier with an RBF kernel underfits the training collection, it is better to raise gamma, which controls the width of the kernel and increases the influence of each training instance. Raising C, which controls the penalty for misclassifications, will also make the model less tolerant to misclassifications and may lead to overfitting.

7. To solve the soft margin linear SVM classifier problem with an off-the-shelf QP solver, how should

the QP parameters (H, f, A, and b) be set?

Ans-)To solve the soft margin linear SVM classifier problem with an off-the-shelf QP solver, the QP parameters should be set as follows: H is a matrix that depends on the kernel and the training data, f is a vector of -1s and 1s representing the class labels, A is a matrix containing the constraints (one per training instance), and b is a vector of zeros.

8. On a linearly separable dataset, train a LinearSVC. Then, using the same dataset, train an SVC and

an SGD Classifier. See if you can get them to make a model that is similar to yours.

Ans-)On a linearly separable dataset, a LinearSVC, SVC, and SGDClassifier are expected to make similar models. However, the specific hyperparameters used for each model may lead to slight differences in the resulting decision boundary and accuracy.

9. On the MNIST dataset, train an SVM classifier. You’ll need to use one-versus-the-rest to assign all 10 digits because SVM classifiers are binary classifiers. To accelerate up the process, you might want to tune the hyperparameters using small validation sets. What level of precision can you achieve?

Ans-) On the MNIST dataset, an SVM classifier can achieve an accuracy of around 95% using one-versus-the-rest classification and hyperparameter tuning. The hyperparameters to tune include C (the penalty for misclassifications), gamma (for RBF kernel), and degree (for polynomial kernel).

10. On the California housing dataset, train an SVM regressor.

Ans-) On the California housing dataset, an SVM regressor can be trained to predict housing prices. The hyperparameters to tune include C (the penalty for errors), epsilon (for the margin around the predicted values), and gamma (for RBF kernel). The performance of the SVM regressor can be compared to other regression models such as Linear Regression or Random Forest Regression.