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

SVM or Support Vector Machine is **a linear model for classification and regression problems**. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are based on the idea of **finding a hyperplane that best divides a dataset into two classes**, as shown in the image below.

SVM needs to find the optimal line with the **constraint of correctly classifying** either class: Follow the constraint: only look into the separate hyperplanes (e.g. separate lines), hyperplanes that classify classes correctly. Conduct optimization: pick up the one that maximizes the margin.

2. What is the concept of a support vector?

Support vectors are **data points that are closer to the hyperplane and influence the position and orientation of the hyperplane**. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: **The algorithm creates a line or a hyperplane which separates the data into classes**.

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

the main advantage of scaling is **to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges**. Another advantage is to avoid numerical difficulties during the kernel calculation.

Because Support Vector Machine (SVM) optimization occurs by minimizing the decision vector w, the **optimal hyperplane is influenced by the scale of the input features** and it's therefore recommended that data be standardized (mean 0, var 1) prior to SVM model training.

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

An SVM classifier **can output the distance between the test instance and the decision boundary**, and you can use this as a confidence score. However, this score cannot be directly converted into an estimation of the class probability.

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?

Yes

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?

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?

8. On a linearly separable dataset, train a LinearSVC. Then, using the same dataset, train an SVC and an SGDClassifier. See if you can get them to make a model that is similar to yours.

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?

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