**EXERCISE 1**

3**. Why no single classifier is the best?**

**Answer:** The No Free Lunch Theorem proves that algorithm works best on all problems. This implies if a model is good on a certain type of problems, it will be bad on another type of problem. Performance depends on dataset, separability, noise levels and data distribution. Small datasets favour simpler model and this avoids overfitting. However, the model may not see the true underlying relationship when your using small dataset. Noisy datasets require simple algorithms as they have fewer parameters. This helps making less assumptions and the noise will less likely be fitted. In separability, linear class boundary favour different approaches from non-linear. Gaussian data suits Naive Bayes, arbitrary distributions need non-parametric methods.

**Exercise 2**

-

**0.01**: Slower convergence but more stable learning

**0.1**: Good speed and stability

**1.0**: Fast convergence but may overshoot optimal solution

- moons datasets fail because perceptions only create straight line while moons require non-linear boundary

**Exercise 3**

C = 0.01 (Underfitting): poor performance on both training and test data. It does not allow model to learn complicated pattern

C = 1 (Balanced): The model learns good pattern without fitting the Noice

C = 100 (Overfitting): poor performance on unseen data due to the model fitting the data too closely

**Exercise 4**

* The model is penalized(punished) severely for any wrong prediction so it prioritizes getting every training data Correct
* Support vectors are stored in svm.support\_vectors\_ array once the model is fitted. Their values are dependent on X\_train\_std and y\_train data.

**Exercise 5**

* Low gamma = low performance
* Medium gamma = balance between over fitting and performance
* High gamma = overfitting
* **Iris dataset**: Linear kernel performs because classes are linear separable
* **Moons dataset**: essential due to non-linear nature

**Exercise 6**

* The decision tree has 3 levels of depth. It was trained on105 samples with balanced classes. It uses the Gini impurity measure to make the splits

**Exercise 8**

* k=1 (Overfitting)

Training Accuracy: 1.000 (perfect)

Test Accuracy: ~0.90-0.95

The model memorises the data instead with complex decision boundaries.

The model will perform bad on untested data

* 2. k=5 (Good Balance)

Training Accuracy: ~0.95-0.98

Test Accuracy: ~0.95-0.98

There is a good balance between generalization and complexity of the model

* k=10 (Underfitting)

Training Accuracy: ~0.90-0.95

Test Accuracy: ~0.90-0.95

The model over simplifies the decision boundaries leading to poor performance both on testing and training data.

**Exercise 9**

* Logistic Regression performs well on linear because it assumes linear decision boundaries.

KNN usually outperforms Logistic Regression significantly on non- linear

**Conclusion**

* Regularization prevents overfitting by penalizing the model every time the buas gets high.
* Use simple model on straight forward problems, when you have less data or you have clear patterns.
* Use ensembles when you have complex problems, big data and more computing resources