0. Preparation

0.1.1 (4 points) For each of the five algorithms list key strength and key weakness. Use no more than 250 words in total (+- 50 per algorithm).

- GaussianNB:

✓ it can perform much better than other four classifiers when the samples are independent predictors	× It highly relies on the assumption of independent predictors
✓ The training speed is faster even with a large dataset.	× It has Zero Frequency problem
✓ It requires a small amount of training data	

- DecisionTreeClassifier:

✓ It requires less preprocessing work	× Insignificant variations in the data lead to the global changes in the tree structure
✓ It can deal with the null values	× The training and optimization are expensive
✓ It is easy to interpret after being built	× The ability to predict continuous values is lower

- KNeighborsClassifier:

✓ It is instance-based learning, which can be faster	× It performs worse in large datasets
✓ New data can be added seamlessly	× It performs worse in high dimensions
	× It is sensitive to varying scales, noises, missing values, and outliers

- SVC:

✓ It performs well if there is a clear margin of separation	× It is difficult to be interpreted after being built

- ✓ It is more effective in high dimensions
- ✓ The kernel function is strong to solve a complex problem
- × The tuning process is tough

- LogisticRegression:

- ✓ It is highly interpretable
- ✓ It requires less preprocessing work
- ✓ It performs well on a linearly separable dataset
- × Linear boundaries cannot solve non-linear problems
- × It may be over-fit on high dimensional datasets

0.1.2 (3 points) Carefully read the Scikit-learn hyper-parameter documentation for each of the five algorithms. Based on this documentation explain how the previously mentioned hyper-parameters affect the algorithms and their performance. Express yourself clearly and provide your reasoning. Use no more than 300 words in total (+- 75 per algorithm). Note: You don't have to write anything about the Naive Bayes since it has not hyperparameters of interest.

- DecisionTreeClassifier:

max_depth and min_samples_leaf are the hyper-parameters relevant to tree pruning. In terms of max_depth, its default value is none which means it will expand the tree until all leaves are pure. Turning to min_samples_leaf, the minimum number of samples required to be at a leaf node, its default value is 1. For a small dataset or a dataset with the relatively small number of features, people can ignore to set these values. Otherwise, people can restrict tree structure to avoid overfitting.

- KNeighborsClassifier:

n_neighbors is the number of neighbors used for voting, and its default value is 5. When deciding the class of a sample, it will look k other nearest samples around it, and assign the sample to the class which contains most samples in this k-sample group. weights decides the weights of voting. 'uniform' represents all the k samples voting are equal weights; whereas 'distance' represents the nearer samples have prior rights.

- SVC:

C is the regularization parameter, and the strength of the regularization is inversely

proportional to C. C is used for avoiding overfitting problem. kernel decides the type of the kernel function. Since it has various kernels, it can process both linear and non-linear problems well through changing the kernel. ‘rbf’ kernel is widely used which is Gaussian kernel, mapping the lower dimensions to the higher ones.

- LogisticRegression:

penalty is used to specify the norm in the penalization. ‘l2’ penalty has four optimization algorithms, ‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’; whereas ‘l1’ penalty has only one, ‘liblinear’. C is the regularization parameter, and the strength of the regularization is inversely proportional to C. C is used for avoiding overfitting problem.

random_state is a pseudo-random generated number. If people want to have the same shuffle each time, it should be fixed.