DataSet1

1 **Random Forest**

2 **Neural Network**

3 **Nearest Neighbors**

The CIFAR-100 dataset contains 50,000 training and 10,000 test images of 20 object classes, along with 100 object subclasses.

In[2]:= `ResourceData["CIFAR-100"]`

Out[2]=

Pre-Processing

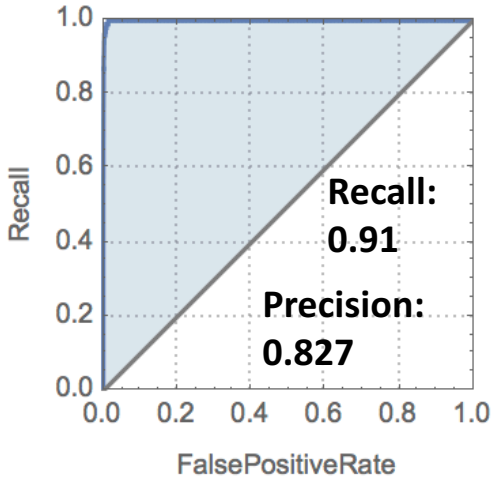
The result of taking 20% of the original data testing 5 different preprocessing method

feD = FeatureExtraction[keys, "DimensionReducedVector"]
Takes least time and yields the highest accuracy. So I choose to use this method to preprocess my data.

Method	Training Time	Accuracy
fe = FeatureExtraction[keys]	229.907s	0.64
feR = FeatureExtraction[keys, "NumericVector"]	231.224s	0.71
feD = FeatureExtraction[keys, "DimensionReducedVector"]	223.637s	0.7175
feI = FeatureExtraction[keys, "ImageFeatures"]	224.111s	0.69
No Preprocessing	225.363s	0.6725

Random Forest

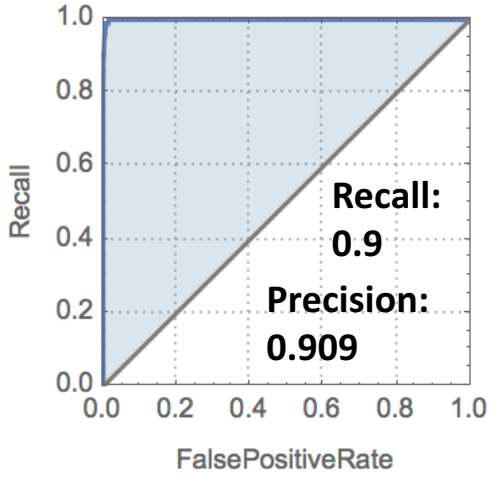
Orange **AUROC: 0.999(Highest)**



Accuracy
0.5368

Neural Network

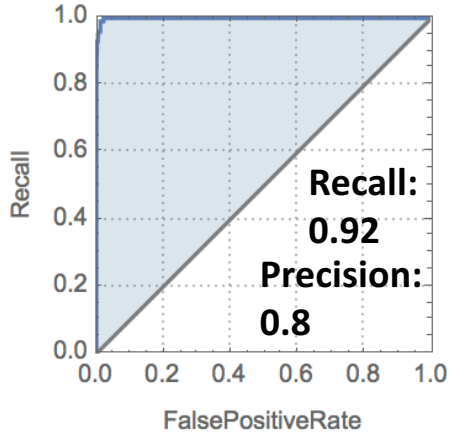
Sunflower **AUROC: 0.999(Highest)**



Accuracy
0.6163

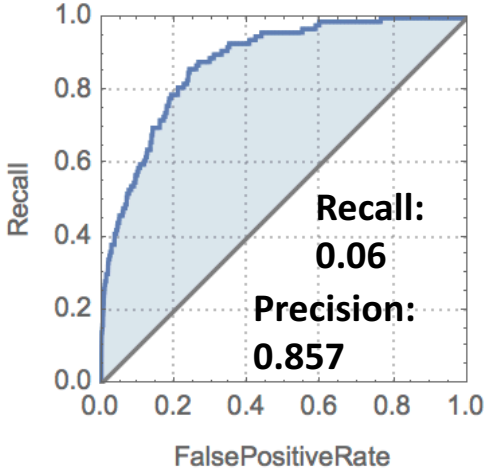
Nearest Neighbors

Apple **AUROC: 0.999(Highest)**



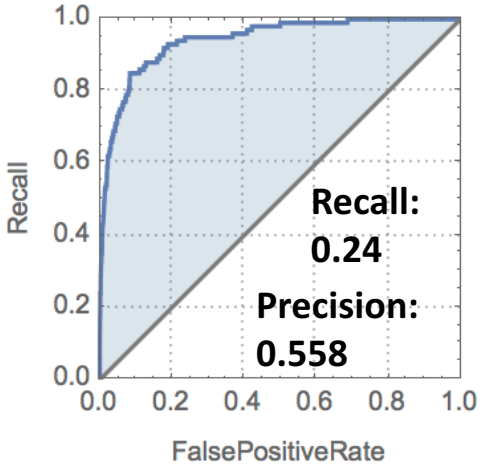
Accuracy
0.6011

Rabbit **AUROC: 0.878(Lowest)**



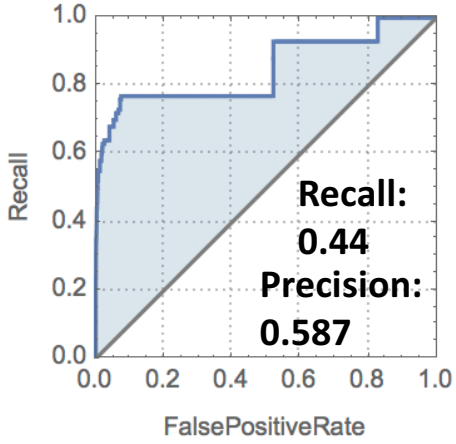
Training time
7037.34s

Beaver **AUROC: 0.943(Lowest)**



Training time
9295.82s

Lizard **AUROC: 0.901(Lowest)**



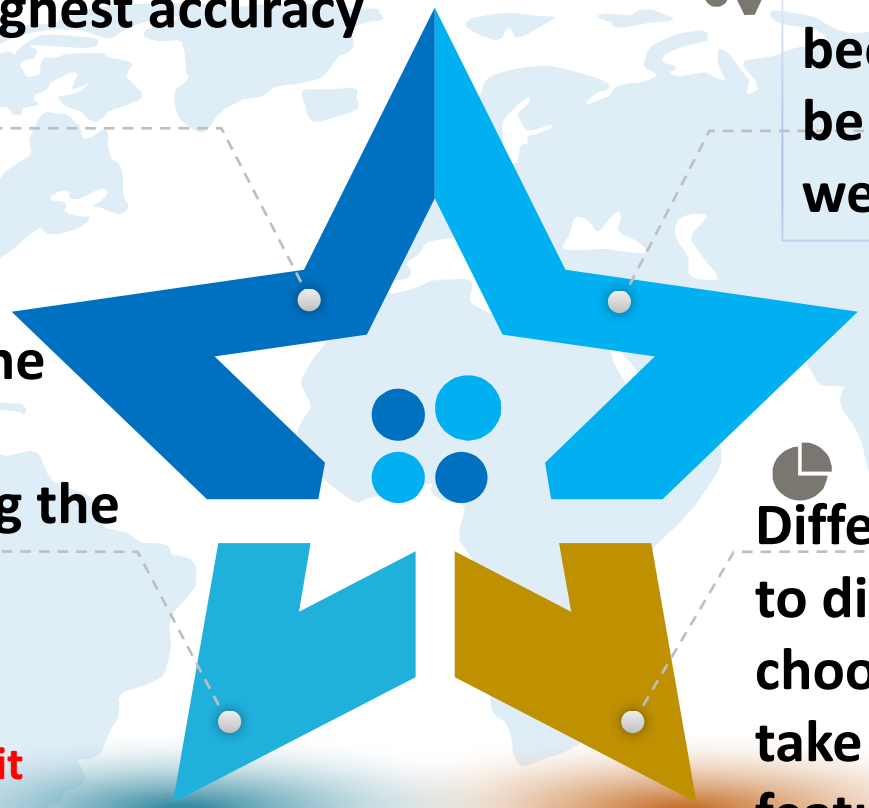
Training time
5387.19s

Conclusions and What I have learned

Neural Network yields highest accuracy but takes most time.

I choose Nearest Neighbors because there are 100 classes to be classified, and k-NN performs well when there are multi-classes.

I don't use cross-validation because we are comparing the performance of different algorithm instead of choosing the best one.



Different algorithm should be applied to different situations, when we are choosing an algorithm, we should take a lot into consideration: the feature of the problem, the pros and cons of the algorithm and etc.

Error source: lack of data, an unfit model for that data, incorrect settings of parameters and so on.