1. **Introduction**
2. **Description and motivation**

From features like Clump Thickness, Uniformity of Cell Size/Shape, Marginal Adhesion, etc I want to find the best model that helps classify whether the cancer is benign (noncancerous) or malignant (cancerous). The prediction is important because determining where the cancer is dangerous or not is critical for the patient. False prediction can lead to serious consequence (for example, the patient may suffer from huge anxiety if we output the result that the patient has malignant while she does not have it). With task requiring huge accuracy like this, finding the right classifier is very important.

1. **Factor to determine the best classifier**

Accuracy: the model needs to perform well on training data, and it also need to perform as good on the data it has not seen before – validation data (not underfit and overfit).

Computational time: the faster the model runs to train the data / make prediction, the better the model is

1. **Methods**
2. **Data preprocessing**

I used pandas to read the data, and I discarded the meaningless id column. Then, I dropped the columns that contains ‘?’ in one of the feature values. The final data contains 683 entries, with 9 features for the input and 1 class (of value 0 for benign and 1 for malignant). I map the output from 2 and 4 to 0 and 1 since it is more intuitive for machine learning/deep learning models (0 signals noncancerous and 1 signals cancerous).

1. **Code packages/library used**

I used python with the following packages:

* Pandas and NumPy to preprocess the data
* Scikit-learn to build and train the models that are not Neural Network (k-nearest neighbors, Decision tree)
* Tensorflow to build and train the Neural Network models
* Matplotlib to plot the result of cross-validation

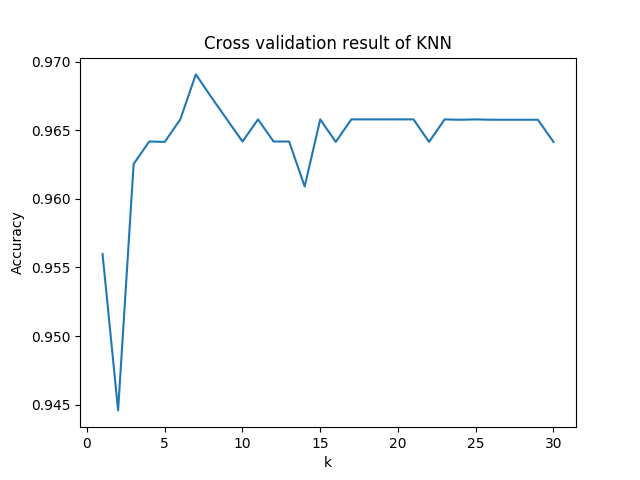
1. **Result**

For each model, I perform 10-fold cross-validation and graph the average score (accuracy) of each model with specific hyperparameter on the validation set.

1. **k-nearest neighbors**

k-nearest neighbors is a machine learning algorithm that based on the similarity defined by distance between data points to make prediction. When encounter an unseen observation, kNN will perform majority vote (or get the average for regression problem) on k seen observations that is closest to the unseen one to determine the class value of it.

I performed cross-validation on k values ranging from 1 to 30, and this is the result:



1. **Decision tree**

t

1. **Random Forest**

t

1. **SVM using the polynomial kernel**

t

1. **SVM using the RBF kernel**

helo

1. **Deep neural network with sigmoid activation**

helo

1. **Deep neural network with ReLU activation**

helo