1. Curse of dimensionality refers to the problem that the space of possible sets of parameter values grows exponentially with the number of unknown parameters, severely impairing the search for the globally optimal parameter values.
   Regularization is a technique that discourages learning a complex model to avoid overfitting. This means regularization makes the model easy to learn and removes less important data to keep the model getting complex. Or else if the model gets complex we might not get anything or the result we want. So, basically regularization helps to get the desired result we want.
2. For categorical variables integer encoding is not enough it can be misleading. One hot encoding can handle categorical attributes. First it maps the categorical values and turns them into integer values then each integer value is represented as binary values 0 and 1.
3. Data Pre Processing has –many steps. Scaling of features is one of them. It is important to perform scaling of features in a data set before applying machine learning methods. Real world dataset contains features which highly vary in magnitudes, units, and range. So, scaling of feature should be performed to simplify the data set. For example: If we have a data set of 100000 peoples. And we have to find the number of diabetic people. And the data set has features blood type, name, age, sex, sugar level etc. To find the diabetic level we mainly need sugar level and age. Other features are not needed. So, before applying machine learning methods other features like name, sex, blood type should be scaled for simple methods.
4. We need to cross validate the training data to check that our model is giving us correct result or not. And the difference between training data result and test data result and to minimize the deference between the test data set and training data set.
   Trade-off between Bias and Variance:

Bias	Variance
difference between the Predicted Value and the Expected Value	the amount of changes will happen to the target function if different training data was used.
Over Simplifies the model and has high error on both train data and test date set.	Pay's lots of attention on train data and test data has high error

5. F1 score is the balance or average between Precision and Recall. I will use F1-score is when the False Negatives and False Positives are crucial. F1-score is a better metric when there are imbalanced classes.
   There is some ways to handle the missing or corrupted data. To have a better accuracy missing or corrupted data should be replaced or dropped. Python pandas has a two function one is isnull() and other is dropna(). Isnull() function full the missing or corrupted data with NULL and dropna() just drops the values. Another way is fill(). It generates a values depending on the rest of the values and full them in the missing or corrupted places.

6. Spliting the data set in 80 and 20:
   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
   Finding the Accuracy:
   for k in range (1,120):
      knn = KNeighborsClassifier(n_neighbors=k)
      knn.fit(X_train, y_train)
      y_pred = knn.predict(X_test)
      scores[k] = metrics.accuracy_score(y_test,y_pred)
      scores_list.append(metrics.accuracy_score(y_test,y_pred))
      print(accuracy_score(y_test,y_pred))
7. Precision = TP / (TP+FP)
   Recall = TP / (TP+FN)
   Average precision-recall score: 0.75
8. Gradient Descent is the process of minimizing a function by following the gradients of the cost function. This involves knowing the form of the cost as well as the derivative so that from a given point you know the gradient and can move in that direction, e.g. downhill towards the minimum value.
   The main problems are:
   i. converging to a local minimum can be quite slow
   ii. if there are multiple local minima, then there is no guarantee that the procedure will find the global minimum (Notice: The gradient descent algorithm can work with other error definitions and will not have a global minimum.

   Overall problems with gradient descent include choosing a proper learning rate so that we avoid slow convergence at small values, or divergence at larger values and applying the same learning rate to all parameter updates wherein if the data is sparse we might not want to update all of them to the same extent.