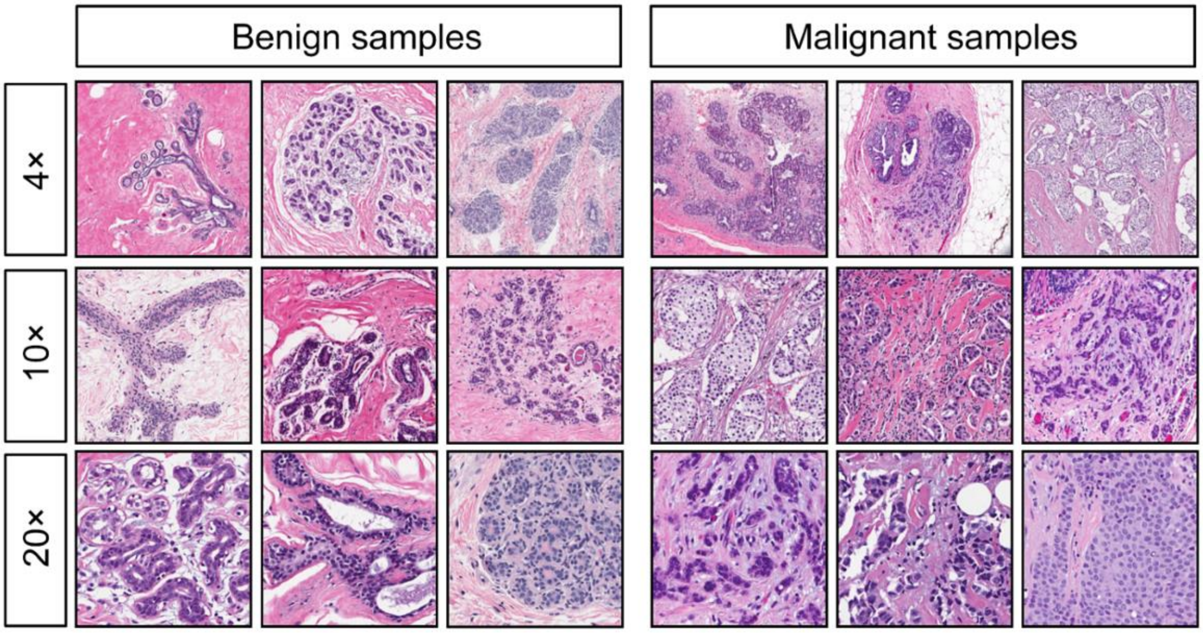
BITS F464 - Machine Learning

Assignment – 1

Submission Time & Date: 2359Hrs on 26th Mar, 1023
Marks: 30

In this assignment you should build machine learning models to predict whether a tumor is benign or malignant.



- ✓ You should make use of data as specified in “Data for Assignment 1.xls”.
- ✓ The final deliverables of the assignment should include the code and detailed report illustrating all the details.
- ✓ You should make use of random 67% of the data to train the model and 33% of the data to test the model.
- ✓ As and when applicable, perhaps at all places, you should have several (at least 10) random training and testing splits of the data and put up results of models of all these splits along with the average and variance of performance metrics.
- ✓ You will be asked to perform certain feature engineering tasks and these tasks are defined and described in detail as and when required.
- ✓ Implementation Requirements:
  - ○ Libraries like Sckitlearn, Tensorflow, Keras and Pytorch should not be used for this assignment.
  - ○ Usage of Numpy, Pandas and Matplotlib libraries is allowed.

- The above two rules apply to all the learning Tasks, along with Feature Engineering Tasks.

Feature Engineering Task 1: For some of the data points, one or more feature values are missing in the data set. You need to fill in those values so that the data tuple can be made use of in building the model as well as testing the model. You should impute the missing value with the most frequent value in the case of categorical feature and impute missing value with the average of the existing values of the corresponding feature if the feature takes continuous numerical values.

Feature Engineering Task 2: You should perform typical normalization of each feature by making use of the following formula: (X' = (X - μ) / σ where μ represents the mean of feature value, and σ represents the standard deviation of feature values).

Part A - Perceptron Learning Algorithm:

Learning Task 1: Build a classifier (Perceptron Model - PM1) using the perceptron algorithm. Figure out whether the data set is linearly separable by building the model. By changing the order of the training examples, build another classifier (PM2) and outline the differences between the models – PM1 and PM2.

Learning Task 2: Build a classifier (Perceptron Model - PM3) using the perceptron algorithm on the normalized data and figure out the difference between the two classifiers (PM1 and PM3).

Learning Task 3: Change the order of features in the dataset randomly. Equivalently speaking, for an example of feature tuple (f1, f2, f3, f4, . . , f32), consider a random permutation (f3, f1, f4, f2, f6, ….., f32) and build a classifier (Perceptron Model – PM4). Would there be any change in the model, PM4, as compared to PM1. If so, outline the differences in the models and their respective performances.

Part B – Fisher's Linear Discriminant Analysis (You can use sckit-learn library only for Fischer's Linear Discriminant projection to 1 Dimension in this Part B. Or alternatively you may code the same in Numpy, Pandas and Matplotlib libraries of python):

Learning Task 1: Build Fisher's linear discriminant model (FLDM1) on the training data and thus reduce 32dimensional problem to univariate dimensional problem. Find out the decision boundary in the univariate dimension using generative approach. You may assume gaussian distribution for both positive and negative classes in the univariate dimension.

Learning Task 2: Change the order of features in the dataset randomly. Equivalently speaking, for an example of feature tuple (f1, f2, f3, f4, . . , f32), consider a random permutation (f3, f1, f4, f2, f6, ….., f32) and build the Fisher's linear discriminant model (FLDM2) on the same training data as in the learning task 1. Find out the decision boundary in the univariate dimension using generative approach and you may assume gaussian distribution for both positive and negative classes in the univariate dimension. Outline the difference between the models – FLDM1 and FLDM2 - and their respective performances.

Part C – Logistic Regression:

Learning Task 1: Build a classification model (LR1) using Logistic Regression. What happens to testing accuracy when you vary the decision probability threshold from 0.5 to 0.3, 0.4, 0.6 and 0.7.

Learning Task 2: You should apply Feature Engineering Task 1 and Feature Engineering Task 2 and then build a classification model (LR2) using Logistic Regression. What happens to testing accuracy when you vary the decision probability threshold from 0.5 to 0.3, 0.4, 0.6 and 0.7.

You must implement the above two learning tasks using Batch Gradient Descent, Mini-batch Gradient Descent and Stochastic Gradient Descent using learning rates 0.01, 0.001 and 0.0001. Plot the learning curves (Cost Function vs Iterations) for the same.

Part D – Comparative Study:

Learning Task 1: Perform a comparative study of models PM1, PM3, PM4, FLDM1, FLDM2, LR1 and LR2. The average performance metrics of 10 random training and testing splits should be considered for this comparative study. Find out the best performing model and if possible explain the reasons for that model to outcast other models.

Contact for clarifications

In case of any queries please contact the Teaching Assistants (TAs) of the course by email and any other communication is invalid. You should write mail to all the following TAs for any clarification.

1. Mr. Akshat Agrawal, f20190264@hyderabad.bits-pilani.ac.in
2. Ms. Chavali Lalita, p20190423@hyderabad.bits-pilani.ac.in
3. Ms. Chaitra C R, p20210024@hyderabad.bits-pilani.ac.in
4. Ms. P.Priyanka Chowdary, p20210022@hyderabad.bits-pilani.ac.in