LMP1210H Winter 2025 Assignment 2

Yuliang Xiao – yl.xiao@mail.utoronto.ca

February 12, 2025

Instructors: Rahul G. Krishnan & Bo Wang

By turning in this assignment, I agree by the UofT honor code and declare that all of this is my own work.

Problem 1 - Decision Tree for Heart Failure Prediction

To get the most important risk that contributes to heart failure, the information gain formula for each feature and entire dataset should be used:

$$IG(\mathcal{S}, \mathcal{X}) = \mathcal{H}(\mathcal{S}) - \sum_{v \in \mathcal{X}} |\frac{\mathcal{S}_v}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_v) \tag{1}$$

where $\mathcal{S}, \mathcal{S}_v$ represent the entire dataset and subset of dataset where the feature $\mathcal{X}$ selects the value v respectively. $\mathcal{H}(\mathcal{S})$ is the Shannon entropy of the dataset:

$$\begin{gathered} \mathcal{H}(\mathcal{S}) = -\sum_{i \in \mathcal{C}} \mathcal{P}_i \log_2 \mathcal{P}_i \\ \sum_{i \in \mathcal{C}} \mathcal{P}_i = 1 \end{gathered} \tag{2}$$

$\mathcal{C} \in \{0, 1, 2, .., N\}$ is the label class of the dataset and $\mathcal{P}_i$ is the probability or fraction of class label i in the dataset. Given Eq. (1) and Eq. (2), we could compute the information gain of each feature and find the most important risk.

- Step 1 - Compute $\mathcal{H}(\mathcal{S})$ with respect to Heart Failure

  Since the dataset only has two label classes $\mathcal{C} \in \{0, 1\}$

$$\begin{gathered} \mathcal{P}_0 = \frac{2}{6} = \frac{1}{3}, \mathcal{P}_1 = \frac{4}{6} = \frac{2}{3} \\ \mathcal{H}(\mathcal{S}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.918 \texttt{ bits} \end{gathered} \tag{3}$$

- Step 2 - Compute $IG(\mathcal{S}, \texttt{Chest Pain})$

  Split the **Chest Pain** feature into 2 groups {0, 1} and compute entropy for each group. The **Chest Pain** feature has 4 positive patients (Patient ID 1, 2, 3, 5) and 2 negative patients (Patient ID 4, 6). In positive group, Patient ID 1, 2, 3 and 5 all have the heart failure=1, therefore:

$$\begin{aligned} \mathcal{P}_0 = 0, \mathcal{P}_1 = 1 \\ \mathcal{H}(\mathcal{S}_{v=1}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.00 \texttt{ bits} \end{aligned} \tag{4}$$

  In negative group, Patient ID 4 and 6 both have heart failure=0, hence the entropy is:

$$\begin{aligned} \mathcal{P}_0 = 1, \mathcal{P}_1 = 0 \\ \mathcal{H}(\mathcal{S}_{v=0}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.00 \texttt{ bits} \end{aligned} \tag{5}$$

$$\begin{aligned} IG(\mathcal{S}, \mathcal{X} = \texttt{Chest Pain}) &= \mathcal{H}(\mathcal{S}) - \sum_{v \in \mathcal{X}} |\frac{\mathcal{S}_v}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_v) \\ &= \mathcal{H}(\mathcal{S}) - |\frac{\mathcal{S}_{v=0}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=0}) - |\frac{\mathcal{S}_{v=1}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=1}) \\ &= 0.918 - \frac{2}{6} \times 0 - \frac{4}{6} \times 0 = 0.918 \texttt{ bits} \end{aligned} \tag{6}$$

- Step 3 - Compute $IG(\mathcal{S}, \texttt{Male})$

  Split the **Male** feature into 2 groups {0, 1} and compute entropy for each group. The **Male** feature has 5 positive patients (Patient ID 1, 2, 4, 5, 6) and 1 negative patients (Patient ID 3). In positive group, Patient ID 1, 2, 5 have the heart failure=1 while Patient ID 4 and 6 have the heart failure=0, therefore:

$$\begin{aligned} \mathcal{P}_0 = \frac{2}{5}, \mathcal{P}_1 = \frac{3}{5} \\ \mathcal{H}(\mathcal{S}_{v=1}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.971 \texttt{ bits} \end{aligned} \tag{7}$$

  In negative group, Patient ID 3 has heart failure=1, hence the entropy is:

$$\begin{aligned} \mathcal{P}_0 = 0, \mathcal{P}_1 = 1 \\ \mathcal{H}(\mathcal{S}_{v=0}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.00 \texttt{ bits} \end{aligned} \tag{8}$$

$$\begin{aligned} IG(\mathcal{S}, \mathcal{X} = \texttt{Male}) &= \mathcal{H}(\mathcal{S}) - \sum_{v \in \mathcal{X}} |\frac{\mathcal{S}_v}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_v) \\ &= \mathcal{H}(\mathcal{S}) - |\frac{\mathcal{S}_{v=0}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=0}) - |\frac{\mathcal{S}_{v=1}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=1}) \\ &= 0.918 - \frac{1}{6} \times 0 - \frac{5}{6} \times 0.971 = 0.109 \texttt{ bits} \end{aligned} \tag{9}$$

- Step 4 - Compute $IG(\mathcal{S}, \texttt{Smokes})$

  Split the **Smokes** feature into 2 groups {0, 1} and compute entropy for each group. The **Smokes** feature has 3 positive patients (Patient ID 2, 5, 6) and 3 negative patients (Patient ID 1, 3, 4). In positive group, Patient ID 2, 5 have the heart failure=1 while Patient ID 6 has the heart failure=0, therefore:

$$\begin{aligned} \mathcal{P}_0 = \frac{1}{3}, \mathcal{P}_1 = \frac{2}{3} \\ \mathcal{H}(\mathcal{S}_{v=1}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.918 \texttt{ bits} \end{aligned} \tag{10}$$

  In negative group, Patient ID 1, 3 have heart failure=1 and Patient ID 4 has heart failure=0, hence the entropy is:

$$\begin{aligned} \mathcal{P}_0 = \frac{1}{3}, \mathcal{P}_1 = \frac{2}{3} \\ \mathcal{H}(\mathcal{S}_{v=0}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.918 \texttt{ bits} \end{aligned} \tag{11}$$

$$\begin{aligned} IG(\mathcal{S}, \mathcal{X} = \texttt{Smokes}) &= \mathcal{H}(\mathcal{S}) - \sum_{v \in \mathcal{X}} |\frac{\mathcal{S}_v}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_v) \\ &= \mathcal{H}(\mathcal{S}) - |\frac{\mathcal{S}_{v=0}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=0}) - |\frac{\mathcal{S}_{v=1}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=1}) \\ &= 0.918 - \frac{3}{6} \times 0.918 - \frac{3}{6} \times 0.918 = 0.00 \texttt{ bits} \end{aligned} \tag{12}$$

- Step 5 - Compute $IG(\mathcal{S}, \texttt{Exercises})$

  Split the **Exercises** feature into 2 groups {0, 1} and compute entropy for each group. The **Exercises** feature has 4 positive patients (Patient ID 1, 4, 5, 6) and 2 negative patients (Patient ID 2, 3). In positive group, Patient ID 1, 5 have the heart failure=1 while Patient ID 4 and 6 have the heart failure=0, therefore:

$$\begin{aligned} \mathcal{P}_0 = \frac{1}{2}, \mathcal{P}_1 = \frac{1}{2} \\ \mathcal{H}(\mathcal{S}_{v=1}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 1.00 \texttt{ bits} \end{aligned} \tag{13}$$

  In negative group, Patient ID 2 and 3 have heart failure=1, hence the entropy is:

$$\begin{aligned} \mathcal{P}_0 = 0, \mathcal{P}_1 = 1 \\ \mathcal{H}(\mathcal{S}_{v=0}) = -\mathcal{P}_0 \log_2 \mathcal{P}_0 - \mathcal{P}_1 \log_2 \mathcal{P}_1 = 0.00 \texttt{ bits} \end{aligned} \tag{14}$$

$$\begin{aligned} IG(\mathcal{S}, \mathcal{X} = \texttt{Exercises}) &= \mathcal{H}(\mathcal{S}) - \sum_{v \in \mathcal{X}} |\frac{\mathcal{S}_v}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_v) \\ &= \mathcal{H}(\mathcal{S}) - |\frac{\mathcal{S}_{v=0}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=0}) - |\frac{\mathcal{S}_{v=1}}{\mathcal{S}}| \mathcal{H}(\mathcal{S}_{v=1}) \\ &= 0.918 - \frac{2}{6} \times 0 - \frac{4}{6} \times 1 = 0.251 \texttt{ bits} \end{aligned} \tag{15}$$

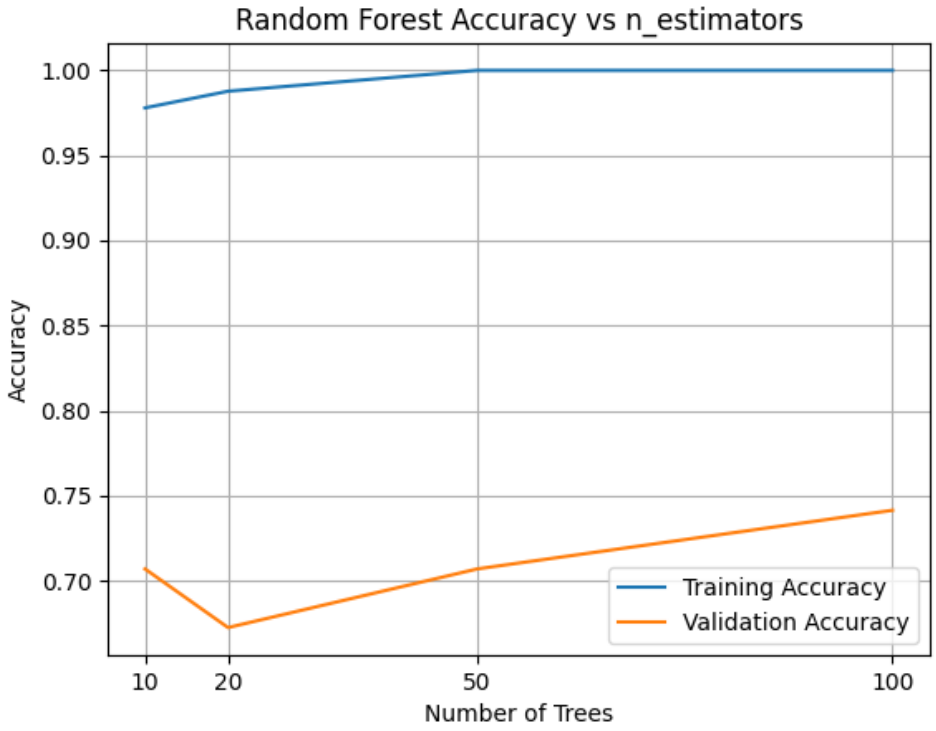
Feature Information Gain Report (bits)			
CHEST PAIN	MALE	SMOKES	EXERCISES
0.918	0.109	0.00	0.251

Table 1: Information gain report of each feature

According to the report table, it is clear to conclude that **CHEST PAIN** has the largest information gain at the root level of decision tree and thus is the most important risk to predict if the patient has the heart failure.

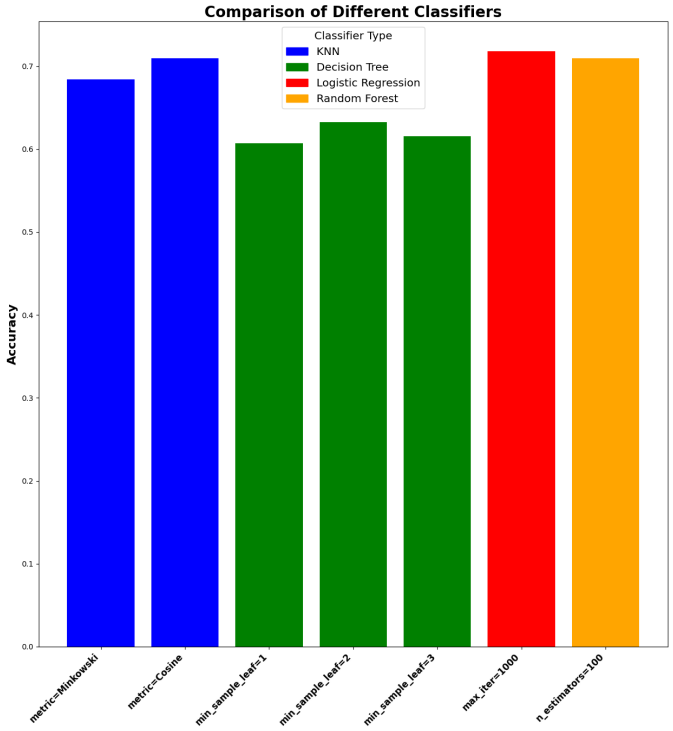
Problem 2 - Revisiting patient classification with Random Forests

(a) Here are the results of different n_estimators:



Random Forests Test Accuracy Report	
best n_estimators	100
accuracy	0.709

Table 2: Test Accuracy Report of Random Forests

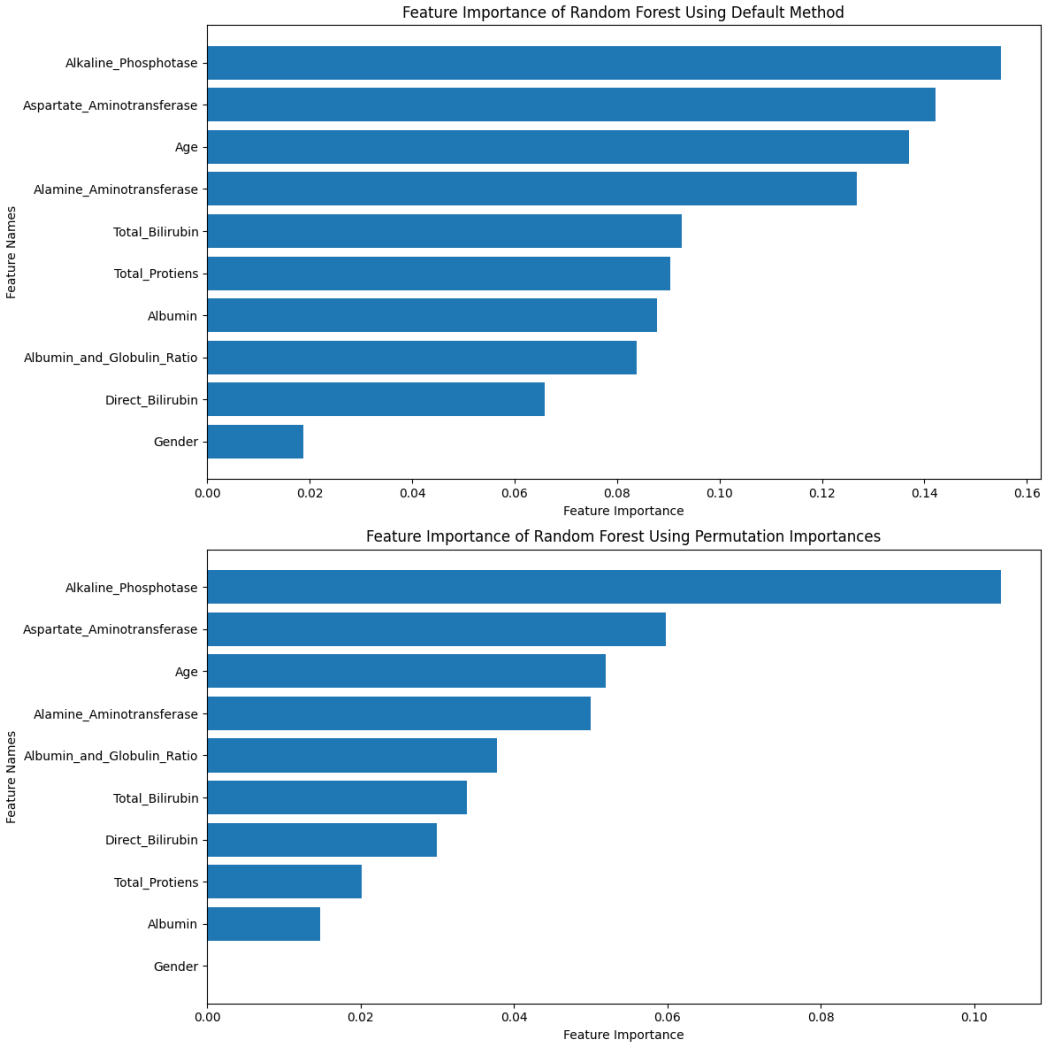


Classification Algorithm Report							
Algorithm	KNN(Minkowski)	KNN(Cosine)	DT(min_leaf=1)	DT(min_leaf=2)	DT (min_leaf=3)	LR	RF($n_est = 100$)
Accuracy	0.684	0.709	0.607	0.632	0.615	0.718	0.709

Table 3: Algorithm comparison report. DT: Decision Tree; LR: Logistic Regression; RF: Random Forests

According to the first figure and Table 2, we can find that the best number of trees in the forest is 100. Additionally, I compared the Random Forest with KNN (Minkowski), KNN (Cosine), Decision Tree (min_sample_leaf=1, 2, 3) and Logistic Regression and draw the bar plot to find the best algorithm. According to the second figure and Table 3, we can conclude that the **Logistic Regression** has the best performance among these algorithms.

(b) Here are two plots of feature importance using different methods:



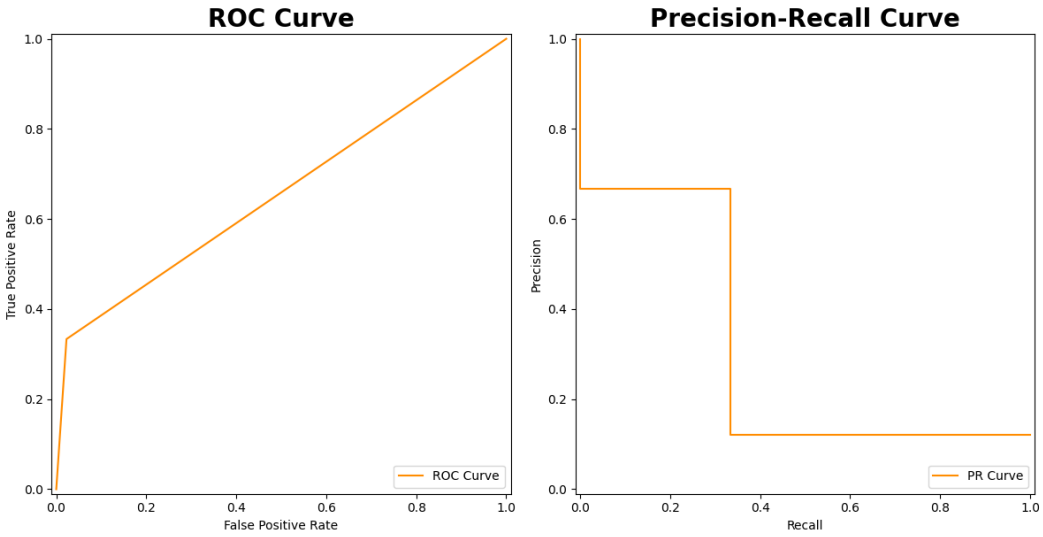
Compared to two plots, we can find that **Alkaline_Phosphotase, Aspartate_Aminotransferase, Age, Alamine_Aminotransferase** are similarly important in two methods while **Alkaline_Phosphotase** plays a dominant role in the permutation importance method. Furthermore, **Albumin_and_Globulin_Ratio** is more important in the permutation methods than default feature importance while **Direct_Bilirubin** is less important in default algorithm than the permutation method. Lastly, there is one significant difference at **Gender** feature. In both algorithm, **Gender** does not play a significant role. However, the permutation method gives 0 importance while default method assigns about 0.02. The permutation method indicates user to ignore **Gender** while the default suggests put low attention on this feature but do not discard it.

Problem 3 - Binary Classification with Imbalanced Data

(a) Here is the report of the logistic regression:

Logistic Regression Test Accuracy Report			
Accuracy	Precision	Recall	F1 Score
0.900	0.667	0.333	0.444

Table 4: Test Accuracy Report of Logistic Regression

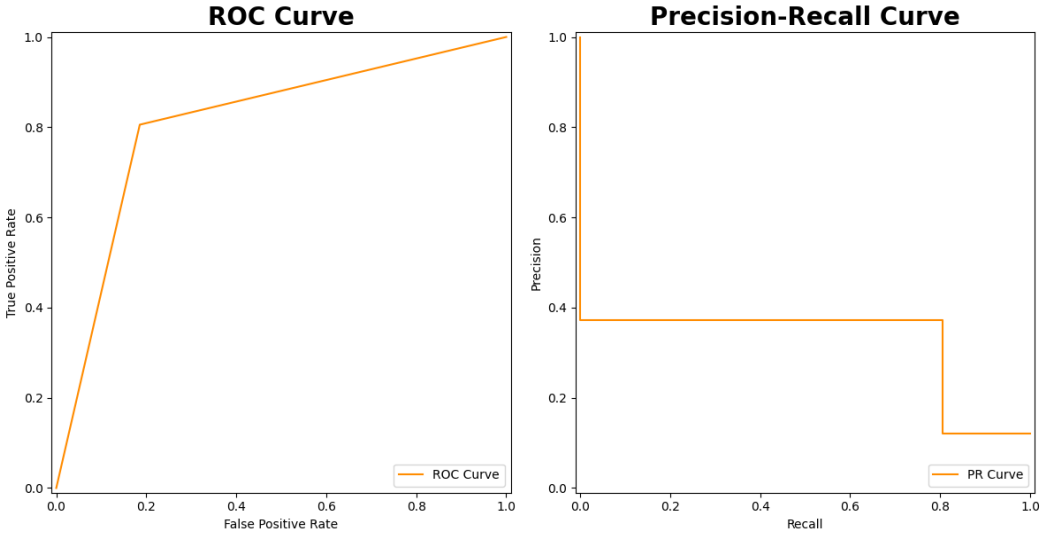


The logistic regression achieves a good accuracy on test data, but the recall is a little bit small which tells that we does not perform well on prediction among all positive prediction results and this is rational since we are working an imbalanced dataset.

(b) Here is the report of the improved logistic regression:

Logistic Regression Test Accuracy Report			
Accuracy	Precision	Recall	F1 Score
0.813	0.372	0.806	0.509

Table 5: Test Accuracy Report of Logistic Regression

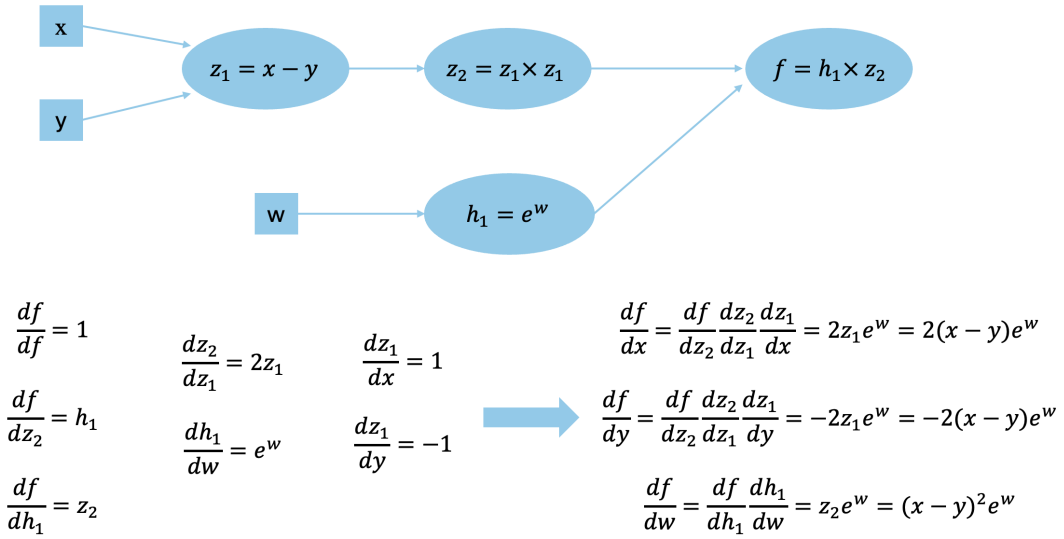


To improve the recall, we can use weights for each class to tell the model the class distribution so that model can be robust during the training. In the **sklearn.linear_model.LogisticRegression** has one argument called **"class_weight"**, we can assign **"balanced"** to this argument variable and make label calculate and adjust weights automatically and dynamically. According to the Table 5, we can find that the recall is improved significantly from 0.333 to 0.806 while we get some loss on precision from 0.667 to 0.372. It is a trade-off for us to balance the precision and recall which should be considered carefully for different downstream tasks.

Problem 4 - Iterative and Analytic Solutions of Linear Regression

Please click and check this Google Colab Notebook.

Problem 5 - Computation Graph and Backpropagation



Here is the computation graph and the process to compute derivative for x, y, and w. We firstly compute the gradient at the baseline and then move one step back to calculate the derivative $\frac{df}{dz_2}$ and $\frac{df}{dh_1}$. We repeat the same process for each branch of the graph until reaching the root node of the branch. Finally, we use the chain rule that multiples the gradients from baseline to the input and understand how changes of input impact on the function f.

Problem 6 - Cell type assignment for single-cell RNA-seq

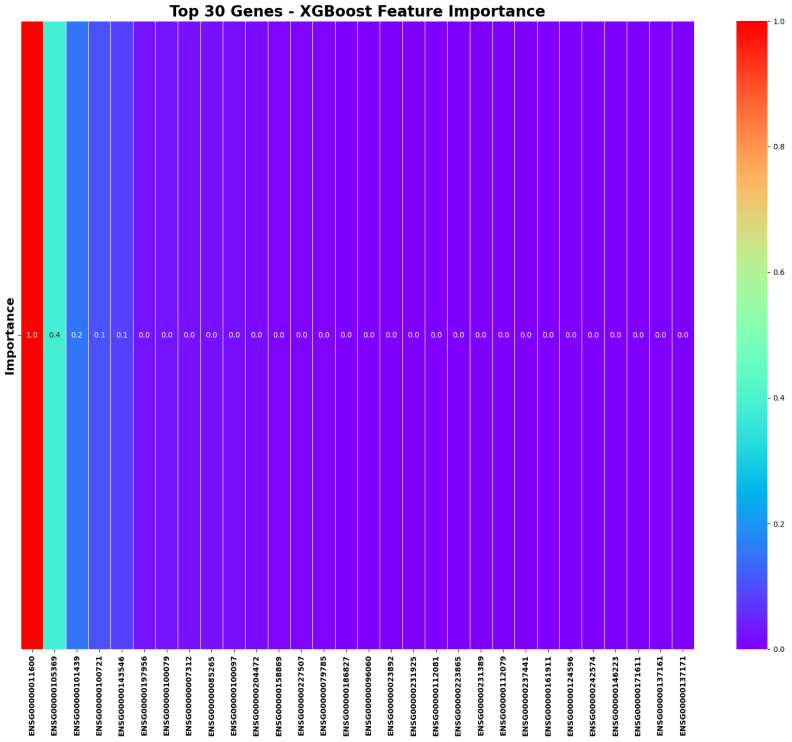
(a) Here is the test accuracy report of XGBoost and MLP model:

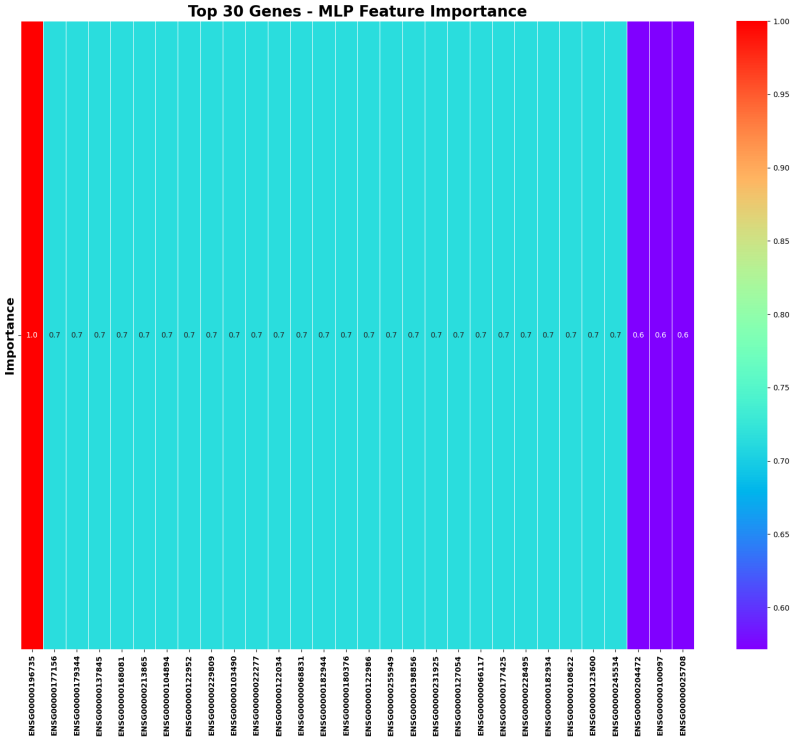
Test Classification Accuracy Report	
Model	Accuracy
XGBoost	0.992
MLP	0.996

Table 6: Test accuracy of XGBoost and MLP models (All values are rounded to 3 significant figures).

According to Table 6, we can conclude that both XGBoost and MLP achieve very good accuracy on the dataset and MLP is slightly better than XGBoost, which indicates more power of MLP than XGBoost model.

(b) Here are the heatmap of feature importance of XGBoost and MLP models:


Top 30 Genes - XGBoost Feature Importance
Importance
1.0 0.4 0.2 0.1 0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
ENSG00000011600
ENSG00000105369
ENSG00000101439
ENSG00000100721
ENSG00000143546
ENSG00000197956
ENSG00000100079
ENSG00000007312
ENSG00000085265
ENSG00000100097
ENSG00000204472
ENSG00000158869
ENSG00000227507
ENSG00000079785
ENSG00000186827
ENSG00000096060
ENSG00000023892
ENSG00000231925
ENSG00000112081
ENSG00000223865
ENSG00000231389
ENSG00000112079
ENSG00000237441
ENSG00000161911
ENSG00000124596
ENSG00000242574
ENSG00000146223
ENSG00000171611
ENSG00000137161
ENSG00000137171
1.0
0.8
0.6
0.4
0.2
0.0


Top 30 Genes - MLP Feature Importance
Importance
1.0 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.6 0.6 0.6
ENSG00000196735
ENSG00000177156
ENSG00000179344
ENSG00000137845
ENSG00000168081
ENSG00000213865
ENSG00000104894
ENSG00000122952
ENSG00000229809
ENSG00000103490
ENSG00000022277
ENSG00000122034
ENSG00000068831
ENSG00000182944
ENSG00000180376
ENSG00000122986
ENSG00000255949
ENSG00000198856
ENSG00000231925
ENSG00000127054
ENSG00000066117
ENSG00000177425
ENSG00000228495
ENSG00000182934
ENSG00000108622
ENSG00000123600
ENSG00000245534
ENSG00000204472
ENSG00000100097
ENSG00000025708
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60

In this problem, I adopt the default feature importance and permutation method for XGBoost and MLP respectively. In the XGBoost feature importance figure, there are only 5 features or genes having non-zero importance and other genes play no role when making classification. Furthermore, the first gene has the importance 1 which is dominantly larger than rest 4 genes. This phenomenon is rational for XGBoost because it implicitly make feature selection and only train with the most informative features. If the feature is not helpful on decision making, then it will be ignored.

Unlike XGBoost, the MLP has a wider range of important genes which are evenly distributed and most genes have importance around 0.7. Since MLP is the distributed learning which means each feature will make a small contribution to the output. During the training, the fully connected layer connects each neuron with all input feature, making the feature importance distribute smoothly. Additionally, it will update weights for each feature every iteration, even the feature is not very important, it still has tiny importance instead of ignoring.