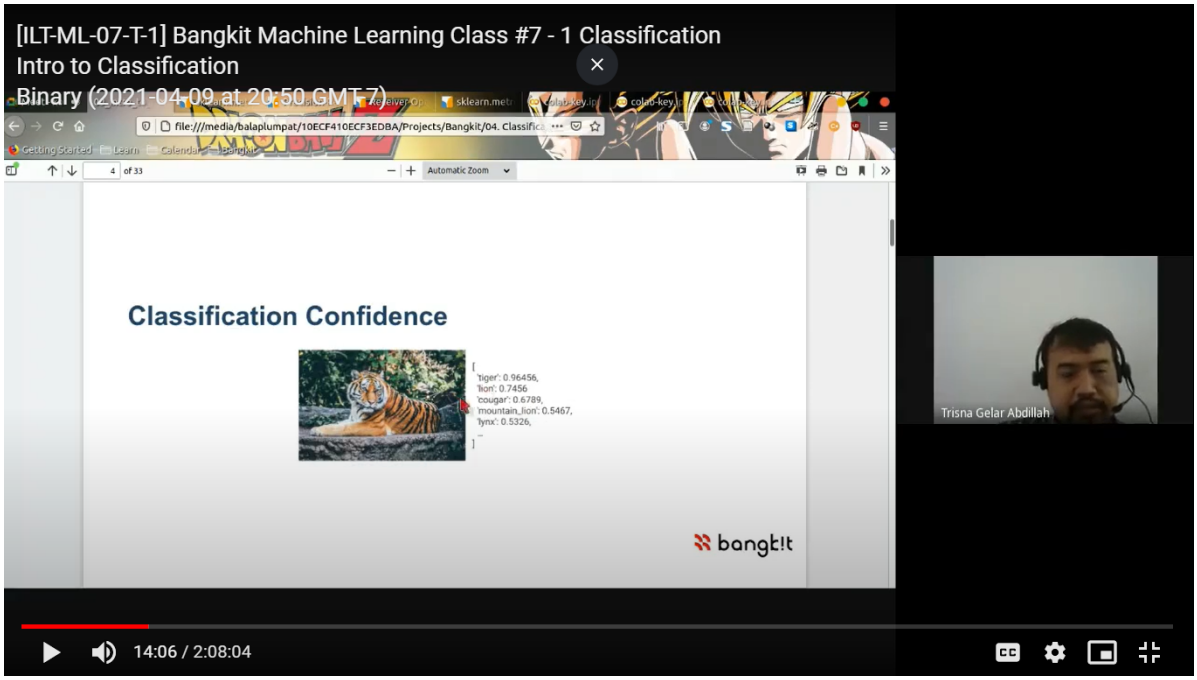
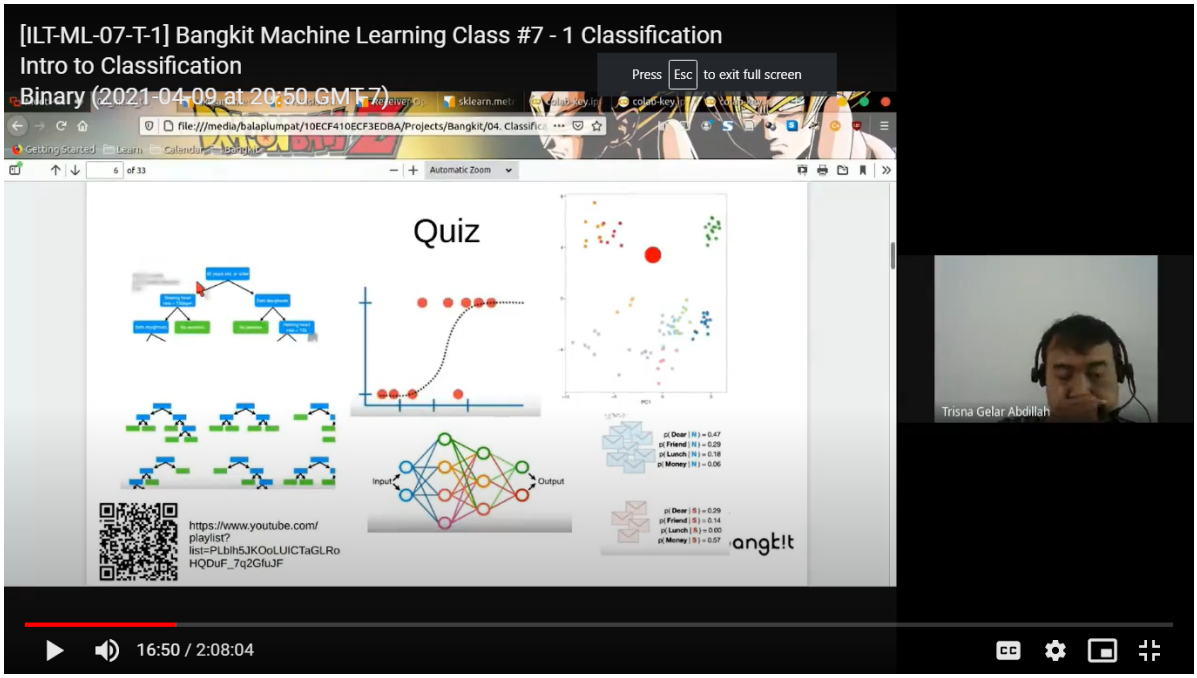
Intro to Classification (ILT-ML-07-T-1)

- The difference between classification and regression. In the simple way we can see the different from output. Classification will give a categorical output refer as class or label. While, regression will give continous output.
- There are two category of classification:
  1. Binary → two class problem (ex: yes or no, spam or not spam, etc)
  2. Multiclass
- Classification confident is probability likelihood of the model to predict class



In the picture 'tiger' class hass the biggest probability likelihood, so the image will be classify as a tiger. But sometimes the model can confuse if the probability not so different. If that happen, then we need to overcome it. One way to solve that problem is to use another algorithm and preprocess the picture.

- There are some classification problem



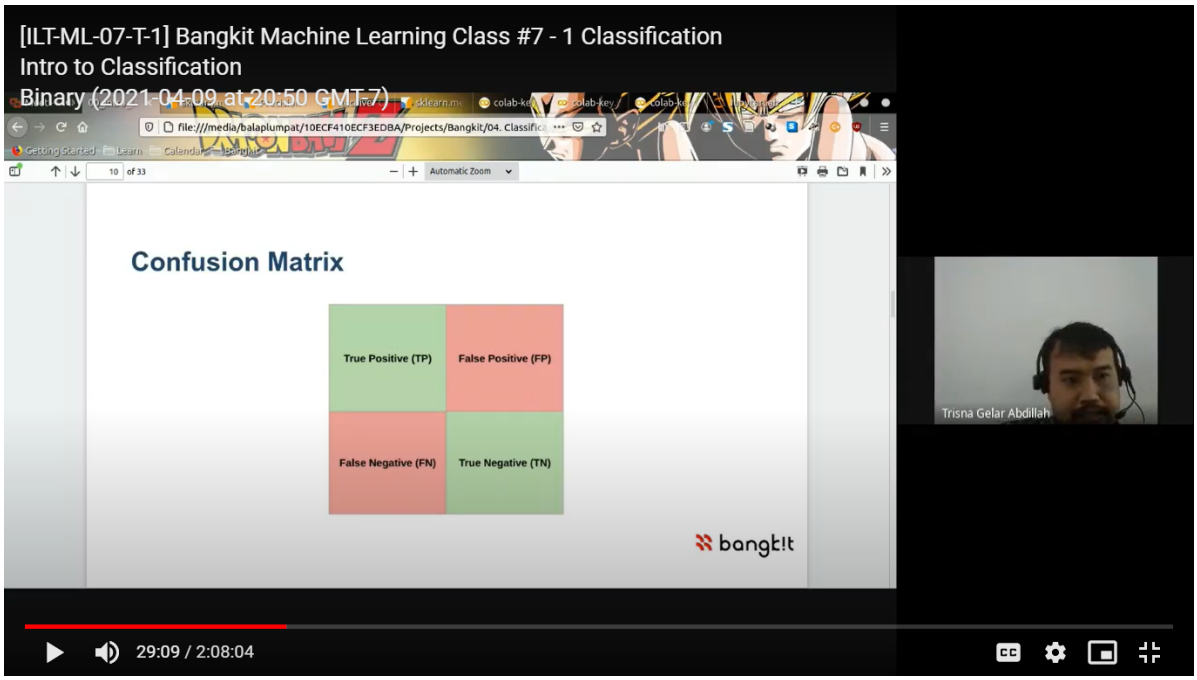
1. Decision Tree. In the picture it locate in the top left.
2. Random Forest. In the picture it locate in bottom left. In Random Forest, we select small feature in each small tree then combine the result.
3. Logistic Regression. In the picture it locate in top middle. Logistic Regression is similar to linear regression but it use log function or softmax function.
4. Neural Network. In the picture, it locate in bottom middle. We have 3 layers in Neural Network: Input Layer, Hidden Layer, and Output Layer. The hidden layer can consist of more than one layer.
5. K-Nearest Neighbor. In the picture, it locate in top right.
6. Naive Bayes. In the picture, it locate in bottom right.

Advantage and Disadvantage Classification Algorithm

Aa Name	Advantage	Disadvantage
Logistig Regression	Good performance with small datasets	Data assumptions are needed to be complied

Name	Advantage	Disadvantage
Untitled	Its output can be interpreted as probability	It can only provide linear solutions
K-Nearest Neighbors	Intuitive algorithm	Number of neighbors must be defined by user
Untitled		High relative computational complexity
Naive Bayes	Performs well in small datasets if conditional independent assumption holds	Assumption of independence between feature
Support Vector Machines	It can provide non-linear solutions	To achieve good performance, they require knowledge about the kernel employed
Decision Trees Ensembles	They can handle categorical features	Interpretability of ensemble can be questioned
Untitled	Few parameters to tune	
Untitled	They prgormm well in datasets with large number of features	
Neural Networks	State-of-the-art results	Many parameters to fine-tune
Untitled	Direct complex immage processing	Large number of samples are required to achieve good performance

- Confusion matrix is used for binary problem



  - True Positive: Model predict positive, actual class positive
  - False Positive: Model predict positive, actual class negative
  - False Negative: Model predict negative, actual class positive
  - True Negative: Model predict negative, actual class negative
- Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

  Accuracy can't give absolute impression about the quality of the model. It is not good if the class have imbalance data.
- Precision $\frac{TP}{TP+FP}$

  Probability of the model got it right when predicting positive
- Recall $\frac{TP}{TP+FN}$

  Probability of the model correctly identify a positive case
- If we want to get high precision we will get lower recall. So, ideally we get balance result. But it is not always the case. In some case like predict tumor we want to get high recall since it is important to identify positive patient, so they will get the treatment they need.
- F1 Score $\frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$

To get balance precision and recall we may use F1 Score. The best score that we expected from F1 Score is 1 and the worst is 0.

- If the situation is critical we should prioritize recall, otherwise precision.
- Receiver Operating Characteristic (ROC)
  - x axis = False Positive Rate
  - y axis = True Positive Rate
  - If the curve leaning towards top left, it means the model is good
  - ROC may used in critical case like cancer prediction.

After the theory, the class continue to try some code in google colab

1. Binary Classification
   - Classify orange and grapefruit based on diameter, weight, and color.
   1. EDA
      - Describe the dataset overall and specifically

        The diameter and weight of grapefruit mostly bigger than the oranges
      - Visualize with boxplot
        - Sample the data because altair can't visualize 20.000 data
        - From the visualize, we can see that the diameter and weight of grapefruit mostly bigger than the oranges
      - See correlation between diameter and weight

        With high correlation like we get in between diameter and weight, we might be able to use it to remove a column from our training data without negatively affecting our model
      - Checking color values

        We plot the color to see if values fall within a reasonable range
   2. Make Simple Logistic Model

      Since we only care whether the fruit is orange or not. To make it easier we will make the label as 'is_orange' which the value is true if it is orange and

otherwise.

3. Split the data to train and test split

   After we split the data, we can see that the data seems unbalanced. To overcome that we will do stratified test split.

4. Stratified from test split
5. Create the training model using logistic regression
6. Measure model performance
7. Using GridSearchCV to optimize the model.

   We use GridSearchCV class to tune hyperparameters of the scikit-learn logistic regressor

2. Multiclass Classification
   - OvO vs OvA
   - Cross-fold validation: Cross-fold validation is useful for small dataset like iris dataset

     We only had few data points in iris datasets. If we are going to be hyperparameter tuning, we'll need a test and validation holdout, which will leave us very little data to train on. One way to get around this is to use cross-validation. Cross-validation splits the data into a fixed number of tranches and trains on `n-1` of the tranches. Then it calculates a score using the holdout tranche. It does this repeatedly, holding out one tranche of data for each training pass.

3. Classification with TensorFlow

   Classify UCI Heart Disease dataset with neural network using tensorflow.