ITD105 Case Study #1

Comparing Machine Learning Algorithms

Name: Adam Russel Shane P. Oguis

I CLASSIFICATION

Train the **classification dataset** using various machine learning algorithms designed for classification. Evaluate and compare these models by applying different resampling techniques and utilizing appropriate performance metrics.

Classification Dataset

Dataset Name: Employee

Features: Education, JoiningYear, City, PaymentTier, Age, Gender, EverBenched, ExperienceInCurrentDomain, LeaveOrNot

Set A

Resampling Technique: Train/Test Split (80:20)

Classification Metric: Confusion Matrix and Classification Report

ML Algorithm (Classification)	Confusion Matrix *(Provide the matrix and classification report of each algorithm)*
CART (Classification and Regression Trees)	Accuracy: 84.533% # Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # CART (Classification and Regression Trees) PaymentTier(target) Confusion Matirix and Classification Report Confusion Matrix: [[2 9 39] [1 142 37] [1 57 643]] Classification Report: precision recall f1-score support 1 0.50 0.04 0.07 50 2 0.68 0.79 0.73 180 3 0.89 0.92 0.91 701 accuracy 0.85 931 macro avg 0.69 0.58 0.57 931 weighted avg 0.83 0.85 0.83 931
Gaussian Naive Bayes/Naive Bayes	Accuracy: 73.792% # Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # Gaussian Naive Bayes/Naive Bayes PaymentTier(target) Confusion Matirix and Classification Report Confusion Matrix: [[2 20 28] [3 141 36] [16 141 544]] Classification Report: precision recall f1-score support 1 0.10 0.04 0.06 50 2 0.47 0.78 0.59 180 3 0.89 0.78 0.83 701 accuracy 0.74 931 macro avg 0.49 0.53 0.49 931 weighted avg 0.77 0.74 0.74 931

Gradient Boosting Machines (AdaBoost)	Accuracy: 82.277% # Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # Gradient Boosting Machines (AdaBoost) PaymentTier (target) Confusion Matrix and Classification Report Confusion Matrix: [[0 9 41] [0 102 78] [0 37 664]] Classification Report: precision recall f1-score support 1 0.00 0.00 0.00 50 2 0.69 0.57 0.62 180 3 0.85 0.95 0.89 701 accuracy 0.82 931 macro avg 0.51 0.50 0.51 931 weighted avg 0.77 0.82 0.79 931
K-Nearest Neighbors (K-NN)	Accuracy: 80.559% # Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # K-Nearest Neighbors (K-NN) PaymentTier (target) Confusion Matrix and Classification Report Confusion Matrix: [[2 12 36] [4 101 75] [8 46 647]] Classification Report: precision recall f1-score support 1 0.14 0.04 0.06 50 2 0.64 0.56 0.60 180 3 0.85 0.92 0.89 701 accuracy 0.81 931 macro avg 0.54 0.51 0.52 931 weighted avg 0.77 0.81 0.79 931
Logistic Regression	# Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # Logistic Regression Accuracy: 78.947% PaymentTier (target) Confusion Matrix and Classification Report Confusion Matrix: [[0 11 39] [0 75 105] [0 41 660]] Classification Report: precision recall f1-score support 1 0.00 0.00 0.00 50 2 0.59 0.42 0.49 180 3 0.82 0.94 0.88 701 accuracy 0.79 931 macro avg 0.47 0.45 0.46 931 weighted avg 0.73 0.79 0.75 931
Multi-Layer Perceptron (MLP)	# Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # Multi-Layer Perceptron (MLP) Accuracy: 73.255% PaymentTier (target) Confusion Matrix and Classification Report Confusion Matrix: [[0 22 28] [0 152 28] [0 171 530]] Classification Report: precision recall f1-score support 1 0.00 0.00 0.00 50 2 0.44 0.84 0.58 180 3 0.90 0.76 0.82 701 accuracy 0.73 931 macro avg 0.45 0.53 0.47 931 weighted avg 0.77 0.73 0.73 931
Perceptron	# Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # Perceptron Accuracy: 19.334% PaymentTier (target) Confusion Matrix and Classification Report Confusion Matrix: [[0 50 0] [0 180 0] [0 701 0]] Classification Report: precision recall f1-score support 1 0.00 0.00 0.00 50 2 0.19 1.00 0.32 180 3 0.00 0.00 0.00 701 accuracy 0.19 931 macro avg 0.06 0.33 0.11 931 weighted avg 0.04 0.19 0.06 931

Random Forest	# Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # Random Forest Accuracy: 81.955% PaymentTier (target) Confusion Matrix and Classification Report Confusion Matrix: [[1 15 34] [2 117 61] [11 45 645]] Classification Report: precision recall f1-score support 1 0.07 0.02 0.03 50 2 0.66 0.65 0.66 180 3 0.87 0.92 0.90 701 accuracy 0.82 931 macro avg 0.53 0.53 0.53 931 weighted avg 0.79 0.82 0.80 931
Support Vector Machines (SVM)	# Resampling Technique: Train/Test Split (80:20) # Classification Metric: Confusion Matrix and Classification Report # Support Vector Machines (SVM) Accuracy: 75.295% PaymentTier (target) Confusion Matrix and Classification Report Confusion Matrix: [[0 0 50] [0 0 180] [0 0 701]] Classification Report: precision recall f1-score support 1 0.00 0.00 0.00 50 2 0.00 0.00 0.00 180 3 0.75 1.00 0.86 701 accuracy 0.75 931 macro avg 0.25 0.33 0.29 931 weighted avg 0.57 0.75 0.65 931

Set B *(should use different resampling technique and classification metric)*

Resampling Technique: K-Fold Cross Validation (k=10)
Classification Metric: Classification Accuracy

ML Algorithm (Classification)	**Classification Accuracy** (Target: PaymentTier)
CART (Classification and Regression Trees)	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy CART (Classification and Regression Trees) Average Accuracy: 0.83258
Gaussian Naive Bayes/Naive Bayes	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy Gaussian Naive Bayes/Naive Bayes Average Accuracy: 0.72276
Gradient Boosting Machines (AdaBoost)	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy Gradient Boosting Machines (AdaBoost) Average Accuracy: 0.81453
K-Nearest Neighbors (K-NN)	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy K-Nearest Neighbors (K-NN) Average Accuracy: 0.79368
Logistic Regression	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy Logistic Regression Average Accuracy: 0.79260
Multi-Layer Perceptron (MLP)	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy Multi-Layer Perceptron (MLP) Average Accuracy: 0.74703
Perceptron	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy Perceptron Average Accuracy: 0.63985
Random Forest	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy Random Forest Average Accuracy: 0.81517

Support Vector Machines (SVM)	Resampling Technique: K-Fold Cross Validation (k=10) Classification Metric: Classification Accuracy Support Vector Machines (SVM) Average Accuracy: 0.75048

Set C *(should use different resampling technique and classification metric)*

Resampling Technique: Repeated Random Train-Test Splits

Classification Metric: Logarithmic Loss

ML Algorithm (Classification)	Logarithmic Loss
CART (Classification and Regression Trees)	(repeat 10 times with 80:20 split) Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss CART (Classification and Regression Trees) Average Logarithmic Loss: 0.42249
Gaussian Naive Bayes/Naive Bayes	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss Gaussian Naive Bayes/Naive Bayes Average Logarithmic Loss: 0.84547
Gradient Boosting Machines (AdaBoost)	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss Gradient Boosting Machines (AdaBoost) Average Logarithmic Loss: 1.06075
K-Nearest Neighbors (K-NN)	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss K-Nearest Neighbors (K-NN) Average Logarithmic Loss: 2.56152
Logistic Regression	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss Logistic Regression Average Logarithmic Loss: 0.54068
Multi-Layer Perceptron (MLP)	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss Multi-Layer Perceptron (MLP) Average Logarithmic Loss: 0.78766
Perceptron	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss Perceptron Average Logarithmic Loss: 29.07496
Random Forest	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss Random Forest Average Logarithmic Loss: 1.16001
Support Vector Machines (SVM)	Resampling Technique: Repeated Random Train-Test Splits Classification Metric: Logarithmic Loss Support Vector Machines (SVM) Average Logarithmic Loss: 0.60805

Results interpretation (Set A , Set B and Set C):

In Set A, I used Train/Test Split with a ratio of 80:20 as my Resampling Technique. I also used Confusion Matrix and Classification Report for Classification Metric. Based on the results, CART (Classification and Regression Trees) ML Algorithm has the highest Accuracy and Classification Report. Thus, the best model for the Employee Dataset here is Train/Split Test for the Resampling Technique, Confusion Matrix and Classification Report for the Classification Metric, and CART (Classification and Regression Trees) for the ML Algorithm.

In set B, I used K-Fold Cross Validation (k=10) for my Resampling Technique, Classification Accuracy for my Classification Metric. Based on the result, still the CART (Classification and Regression Trees) has the highest Average Accuracy, while Perceptron has the lowest. Thus, the CART (Classification and Regression Trees) is the best ML Algorithm that suit in the Resampling Technique and Classification Metric.

In Set C, I used Resampling Technique: Repeated Random Train-Test Splits and Classification Metric: Logarithmic Loss. Still the CART (Classification and Regression Trees) has the best output with the Average Logarithmic Loss of 0.42249. It is the closest to 0 and in logarithmic loss, output that is closest to 0 is the best score. This means CART (Classification and Regression Trees) is the best ML Algorithm for these models.

Based on the results, perform algorithm/hyperparameter tuning (at least 3) of the chosen ML algorithm.

ML Algorithm: CART (Classification and Regression Trees)
Resampling Technique: Repeated Random Train-Test Splits
Classification Metric: Logarithmic Loss
Dataset: Employee

Model	CART Hyperparameters				
	random_seed	max_depth	min_samples_split	min_samples_leaf	Accuracy
Model I	50	5	2	1	0.42249
Model II	25	5	2	1	0.49321
Model III	50	2	5	5	0.54746

Interpretation:

All of the models have different accuracy level. This means we need to configure what hyperparameters that best fit for our dataset. Model I is the model that I used in the Set C and it has the best accuracy report.

II REGRESSION

Train the **regression dataset** using various machine learning algorithms designed for regression. Evaluate and compare these models by applying different resampling techniques and utilizing appropriate performance metrics.

Regression Dataset
Dataset Name : Salary_dataset
Features: YearsExperience, Salary

Set A
Resampling Technique : Split into train and test sets
Regression Metric : Mean Absolute Error

ML Algorithm (Regression)	Mean Absolute Error
CART (Classification and Regression Trees)	Mean absolute error: 8601.166666666666
Elastic Net	Mean absolute error: 4936.132276518997
Gradient Boosting Machines (AdaBoost)	Mean absolute error: 7330.543055555555
K-Nearest Neighbors (K-NN)	Mean absolute error: 7325.0
Lasso Regression	Mean absolute error: 8647.435251109975
Ridge Regression	Mean absolute error: 7092.156578070826
Linear Regression	Mean absolute error: 8651.33696552308
Multi-Layer Perceptron (MLP)	Mean absolute error: 80581.9304590343
Random Forest	Mean absolute error: 8623.33333333333

Set B *(should use different resampling technique and regression metric)*

Resampling Technique: K-fold Cross Validation

Regression Metric: Mean Squared Error

ML Algorithm (Regression)	**Mean Squared Error**
CART (Classification and Regression Trees)	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error CART (Classification and Regression Trees) Average Mean Squared Error: 40429214.53942
Elastic Net	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error Elastic Net Average Mean Squared Error: 48395681.59193
Gradient Boosting Machines (AdaBoost)	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error Gradient Boosting Machines (AdaBoost) Average Mean Squared Error: 32504402.90499
K-Nearest Neighbors (K-NN)	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error K-Nearest Neighbors (K-NN) Average Mean Squared Error: 34350860.64533
Lasso Regression	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error Lasso Regression Average Mean Squared Error: 40379529.76033
Ridge Regression	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error Ridge Regression Average Mean Squared Error: 39774687.47978
Linear Regression	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error Linear Regression Average Mean Squared Error: 40380614.38331
Multi-Layer Perceptron (MLP)	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error Multi-Layer Perceptron (MLP) Average Mean Squared Error: 6012427864.26636
Random Forest	Resampling Technique: K-fold Cross Validation Regression Metric: Mean Squared Error Random Forest Average Mean Squared Error: 35175301.11740

Result Interpretation Set A and Set B:

Set A has different result based on what ML Algorithm I used. Elastic Net is the best ML algorithm and Linear Regression is the least. In Set B, all values go beyond 1 million since the values I refer is the Salary and salaries are above a thousand.

Based on the results, perform at algorithm tuning (at least 3) of the chosen ML algorithm.

Resampling Technique: K-fold Cross Validation
Regression Metric: Mean Squared Error
ML Algorithm: Elastic Net

	SVM Hyperparameters		
	alpha	**l1_ratio**	**MSE**
Model I	1.0	0.5	48395681.59193
Model II	0.5	0.5	44369117.03290
Model III	1.0	1.0	40379529.76033

Results interpretation:

All three models have different result, mean squared error. Results are based on the hyperparameters. This means configuration of hyperparameters is essential in creating the best model.

Submit the following:

a. Pdf copy of the results.
b. Video link demonstrating the case study.