1. Task #1: Regression

I. Introduction

In this project, the goal was predicting the prices of laptops from different features (e.g., processor type, RAM size, screen size, etc.) For this analysis, we will be using a postprocessed version of Kaggle laptop price dataset, arguably one of the most famous and useful datasets. Our goal was to construct regression models to predict laptops price and tested them through various techniques of machine learning method: Linear Regression, Ridge Regression and Decision Trees. This was intended to verify the models according to the MSE and R^2 (R-squared) values.

II. Data Collection

This data is one of the few archived sets of popular laptop price data from Kaggle, which has been used across various platforms. For each Laptop in the dataset, there are relevant attributes like brand, processor type, RAM size, storage size, screen size, and GPU type that enable us to build a model to predict the laptop price.

This dataset had missing values, categorical variables and few outliers which were all handled against best practices to preserve prediction accuracy. We one-hot encoded categorical variables like brand and processor type and scaled the numerical features like RAM and storage for optimal performance of models trained on them.

III. Data Preprocessing

Several preprocessing steps were performed prior to applying the models:

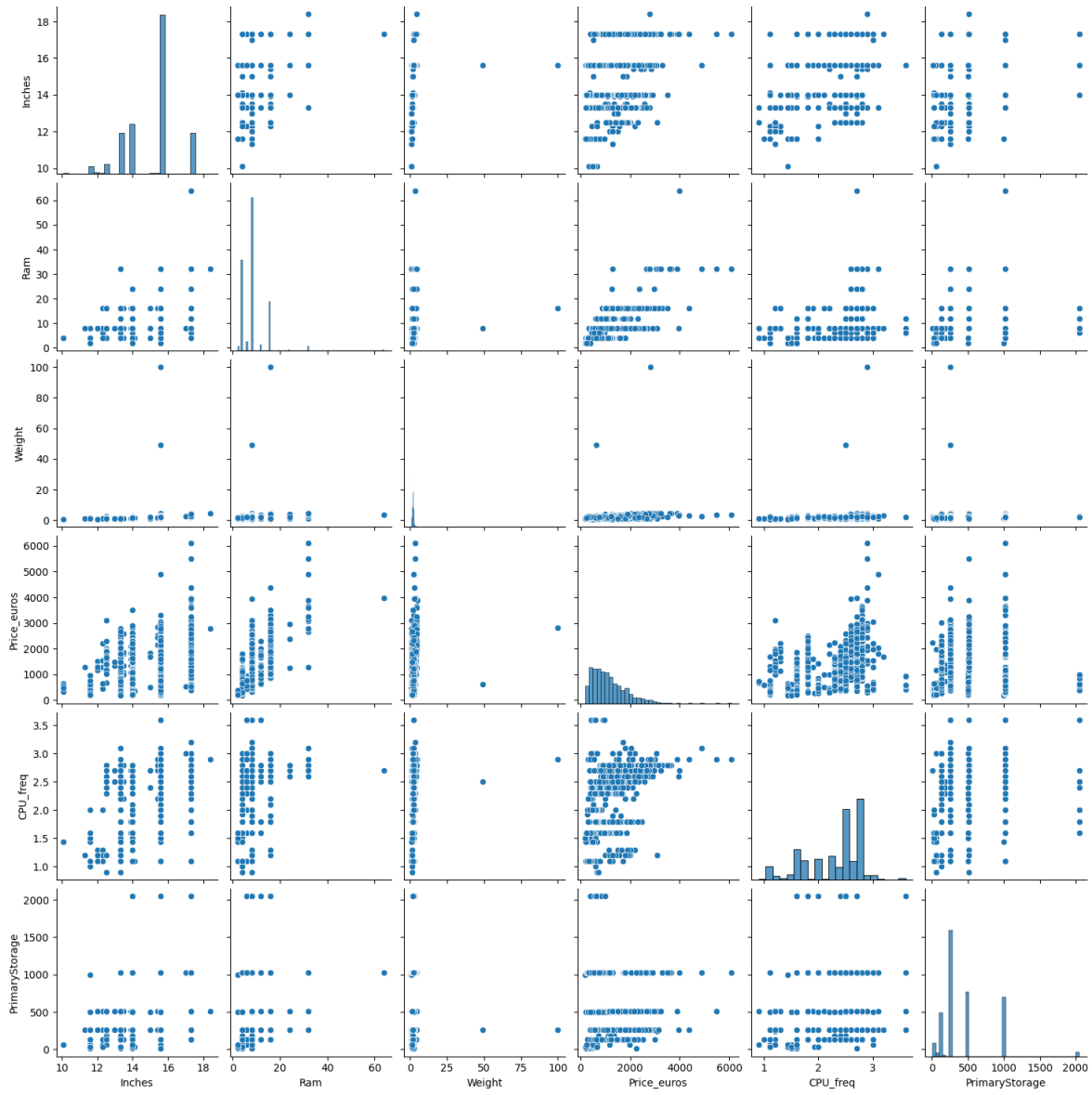
Missing Values: There were no noteworthy missing values in the pre-processed dataset; this was dealt with in the earlier stage.

Duplicated Values: There was only one duplicated value which was removed.

One-Hot Encoding: Categorical Variables like brand and processor type were one-hot encoded to create binary variables with a value of True or False.

IV. Exploratory Data Analysis

Several strong correlations among laptop features can be seen from the pair plot. Price is strongly related to the attributes RAM and CPU frequency, as attributes with a higher value make the laptops more expensive. There is also a positive correlation between RAM and CPU frequency, indicating that indeed, a higher CPU frequency is usually coupled with larger RAM in laptops. Furthermore, we can note that Inches (screen size) and Weight has high positive correlation since devices with larger screen will also weigh slightly, but not always because it could vary for device type. These relationships underline that performance-related qualities such as RAM and CPU frequency are significant contributors to laptop prices, while physical features such as screen size and weight are closely interrelated.



V. Model Development

We implemented and trained the models on linear regression, ridge regression and decision tree. The data was split into 80% train and 20% test, with a random seed of 42 to be certain that the models were never evaluated on the same seen data.

Hyperparameters were tuned for each model to improve performance. For example, the regularization strength (alpha) for both Ridge models was tuned.

VI. Performance Evaluation

The performance of each model was evaluated using Mean Squared Error (MSE) and R-squared (R^2). These metrics were used to compare how well the models fit the data.

Model	**MSE**	**R^2**
Linear Regression	91,789.900	0.79063
Ridge Regression	67,523.382	0.84598
Decision Tree	122,744.617	0.72002

Linear Regression: The best MSE achieved by a linear model was 91,789.900, while R^2 was 0.79063, good results but not as good as the regularized models.

Ridge Regression: The best performing model, with an MSE of 67,523.382 and R^2 of 0.84598 (albeit with a slight amount of overfitting), indicating that regularization implemented by the model improved the generalization capacity.

Decision Tree: Below, you can see the results of the Decision Tree model, which was the least performing model: MSE: 122744.617, R^2 is 0.72002. This suggests that since Decision Trees are prone to overfitting the training data when they are not appropriately regularized, it may be the case that there has been an overfitting situation.

VII. Interpretation

Ridge Regression gave us the lowest MSE and highest R^2, showing that this generalized the best on the test data. Regularization was useful in avoiding overfitting thus allowing us to strike a good mean between bias and variance. The linear regression model has a relatively high R^2 value of 0.79, meaning it explains 79% of the variance in the target variable. This is generally a good result, suggesting that linear regression provides a reasonable fit. However, the MSE is also relatively high, indicating that there is still substantial error in the predictions.

Though it is interpretable, the decision tree model performed badly because of overfitting. It was the one with the highest MSE and the lowest R^2 suggesting that it had captured noise in the data.

VIII. Conclusion

The goal of predicting laptop prices was successfully achieved with different regression techniques. Ridge Regression yielded the best results, striking a good balance between complexity and performance. We can see Decision Trees are poor in terms of performance, underlining the need for tuning and regularizing to avoid overfitting. Linear Regression was a simple and effective model yet surpassed by Ridge models.

2. Task #2: Classification

I. Introduction

In this task, we attempted to predict if the students would pass or fail or drop out based on their academic features and demography. For this analysis, we will use the Student Academic Performance dataset obtained from the UC Irvine Machine Learning Repository. using the labels above, the task is a binary classification problem deciding whether the student will graduate (success) or drop out. We ran 3 classification algorithms: Logistic Regression, KNN and Naive Bayes. The prediction models were assessed by the metrics accuracy, precision, recall, F1-score, confusion matrix, etc.

II. Data Collection

The dataset used in this analysis is from the UC Irvine Machine Learning Repository and consists of academic and demographic records of students, including personal data, academic performance, target variable etc.

III. Data Preprocessing

The following preprocessing steps were applied before training the models:

Addressing Duplicate Values: There were no duplicate values in the dataset which requires no imputation.

Addressing Missing Values: There were few missing values in the dataset which were removed.

Categorical Features: One-hot encoding of categorical features e.g. address (urban or rural) was also used to enable our model to learn.

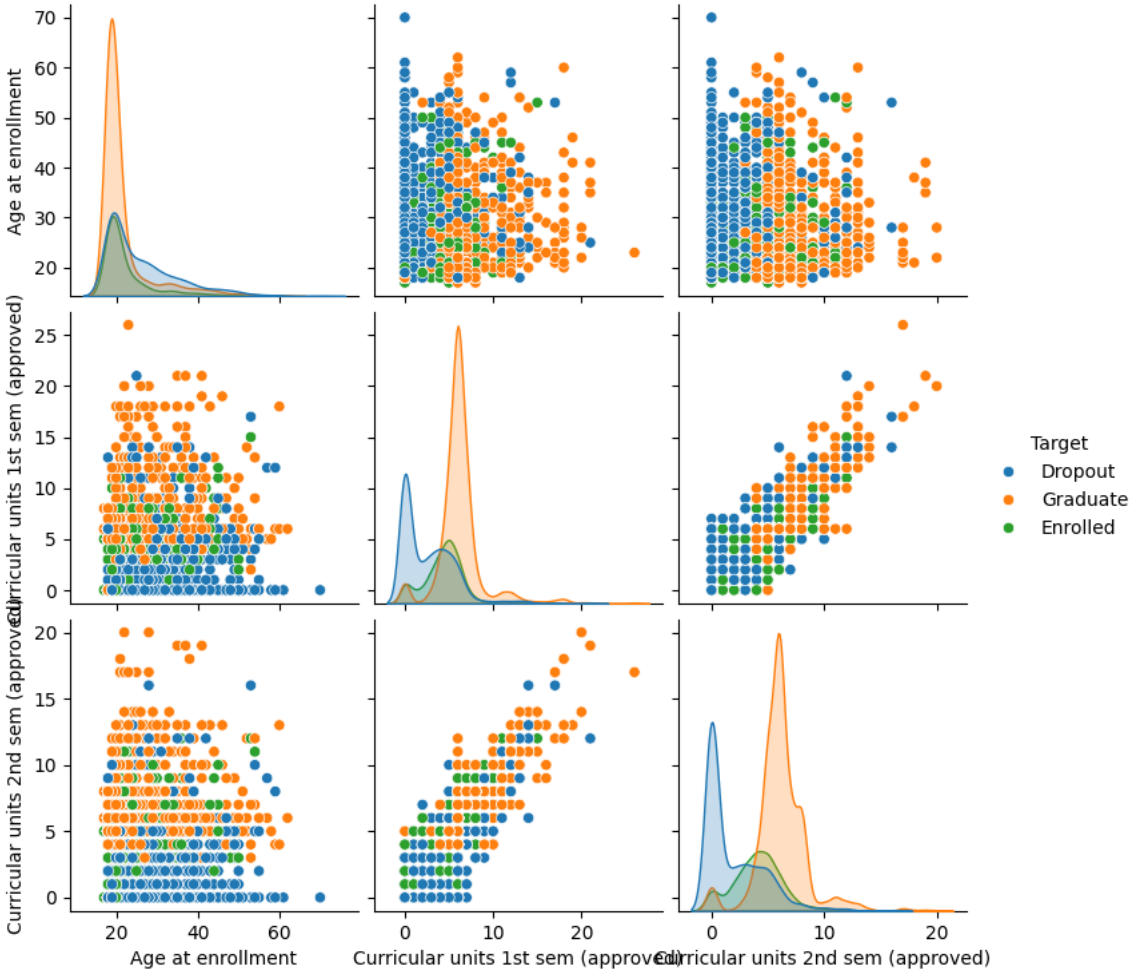
Feature Scaling: Logistic Regression requires feature scaling. As a result, StandardScaler was used to scale features.

Train-Test Split: We split the dataset into 80% training data and 20% testing data to determine how well does the model perform on unseen instances.

IV. Exploratory Data Analysis

During EDA stage, we looked at the relationships (correlations) between different features and target.

These features showed much difference in pair plot for student success or dropout. The orange clusters depict a clear majority of the younger age group (20–30) who graduated, while the dropouts (in blue) show a much wider age distribution and a stronger concentration at low curricular unit approvals. In addition, the first and second semester show a strong correlation with each other, with graduates achieving higher approvals and dropouts concentrating at the bottom. Enrolled students (green) overlap with both dropouts and graduates but typically show moderate to high performance in approvals. These trends suggest that early academic performance and age at enrollment are strong predictors of student outcomes, with low performance in the first semester being a key indicator of dropout risk.



V. Model Development

We applied two models for the classification task: Logistic Regression, KNN and Naïve Bayes.

Hyperparameters such as the number of neighbors were tuned for KNN. The Logistic Regression and Naive Bayes model did not require hyperparameter tuning but was affected by feature scaling.

VI. Performance Evaluation

The accuracy, confusion matrix, and classification report metrics (precision, recall, F1-score) were used to evaluate the performance of both models.

Logistic Regression

- Accuracy: 88.45%
- Confusion Matrix:
  [[567 44]
  [58 214]]

- Classification Report:

Class	Precision	Recall	F1-score	Support
0	0.91	0.93	0.92	611
1	0.83	0.79	0.81	272
Macro avg	0.87	0.86	0.86	883
Weighted avg	0.88	0.88	0.88	883

The overall precision and recall for the Logistic Regression is high for both classes, but it is slightly better for the successful students (Class 0) than for dropout students (Class 1).

The precision for Class 1 (dropout students) is slightly lower, which shows that the classifier mislabels some dropout students as successful.

The recall for Class 1 is also less than 100%, which also means that the model is not predicting all the dropout students (false negatives).

KNN

- Accuracy: 84.60%
- Confusion Matrix:
  [[576 35]
  [101 171]]

- Classification Report:

Class	Precision	Recall	F1-score	Support
0	0.85	0.94	0.89	611
1	0.83	0.63	0.72	272
Macro avg	0.84	0.79	0.80	883
Weighted avg	0.84	0.85	0.84	883

KNN performed well with an accuracy of 84.60%. It achieved good precision for both classes but had a lower recall for Class 1 (dropouts). The model was more likely to identify successful students correctly but misclassified some dropouts.

Naïve Bayes

- Accuracy: 34.20%
- Confusion Matrix:
  [[34 574]
  [7 265]]

- Classification Report:

Class	Precision	Recall	F1-score	Support
0	0.84	0.06	0.11	611
1	0.32	0.97	0.48	272
Macro avg	0.58	0.52	0.30	883
Weighted avg	0.68	0.34	0.23	883

The evaluation gave a poor score of 34.20% to Naive Bayes overall.

The recall for dropout students (Class 1) was very high (0.97) as a trade off against very low precision (0.32). This indicates the model was able to accurately identify most dropout students but also had a high rate of false positives where successful students were labelled as dropouts.

The precision for Class 0 (successful students) is very low (0.06), meaning the model had a lot of false positives for inferring this class.

VII. Interpretation

Logistic Regression has edge over all other classifiers in terms of accuracy and equal for overall f1-score but also has good f1-score in terms of class for Success Student. Both models performed similarly in metrics for recall for dropout students, thus, indicating that both models are able to identify students who will drop out of school.

Overall, KNN gave slightly worse results than Logistic Regression and SVM's results, however KNN still got good accuracy (84.60%) and precision for both classes. It also had lower recall for dropout students (Class 1) so some dropouts were misclassified.

Naive Bayes was the worst, at an accuracy of 34.20%. The model correctly identified most of the dropouts (high recall) but made a false-positive error by misclassifying many successful students as dropouts.

VIII. Conclusion

For classification, Logistic Regression is a reasonable choice here, giving us a good balance between accuracy and F1-score, though it might miss some dropout students. KNN is another good option, although this can also be optimized, including adjusting the number of neighbors. Overall naive bayes did not perform well and is not recommended for this problem.