Course	B9BA102 APPLIED STATISTICS AND MACHINE LEARNING
Name (Student Id)	Shashank Borse (20038928) Shudhanshu Desai (20039165)

Question 1

Data Preparation (What steps would you take to prepare your data? Discuss your approach)

Answer

The data provided had the following information:

- Number of records: **1,721**
- Number of attributes: **23**
    - Input features: **22**
    - Output feature: **1**

Step 1: Upload Dataset and Read

The first step is to upload our dataset csv file to *Google Drive* folder and read that file in colab using Python's pandas library.

Step 2: identify and Encode Attributes

The dataset has a mix of attributes which we need to encode to make it more readable. In our case, we have the following types of attributes:

- Numerical attributes: **5 (Inches, Ram, Weight, ScreenW, CPU_freq).**
- Binary attributes: **3 (Touchscreen, IPSpanel, RetinaDisplay).**
- Categorical attributes: **10**
    - Nominal attributes: **6 (Company, TypeName, GPU_company, OS, CPU_model, GPU_model).**
    - Ordinal attributes: **4 (CPU_company, PrimaryStorage, SecondaryStorage, Screen)**

Based on the attribute type, we encode the data in 3 steps:

1. **Map Functions**:
    - The binary and ordinal attributes which are *TouchScreen, IPSpanel, RetinaDisplay, Screen, PrimaryStorageType and SecondaryStorageType* have been converted into numerical values.

2. **Target Encoding**:
    - We use Target Encoding on the *Company, TypeName, OS, CPU_model and GPU_model* columns to replace categorical values to numerical values from the mean of the target column (Price_euros), this will ensure the reduction of dimensions while maintaining the interpretability of the data.
3. **One-Hot Encoding**:
    - For the nominal attributes which are *CPU_company and 'GPU_company* we will convert each unique value to a different binary column, dropping the first category to avoid redundancy.

Step 4: Scale the input data

We scale the input data using the *StandardScaler* function which will make sure that each value has a mean of 0 with standard deviation of 1. This allows the machine to not let the value with higher numbers dominate the prediction.

Step 5: Handle Multicollinearity

By using heatmap we have identified columns with high correlation and the columns with high correlation were dropped to reduce the redundancy and multicollinearity.

Step 6: Dimensionality Reduction Using PCA

We applied Principal Component Analysis (PCA) to numerical columns which are *Ram, Weight, CPU_freq, PrimaryStorage, SecondaryStorage* to again reduce dimensionality while retaining majority of the data variance. This will also help in increasing the performance of the model.

Question 2

Impact of L1, L2, and elastic net regularization on linear regression coefficients, performance, and interpretability. (Note - models built with fewer variables are considered more interpretable)

Answer

The given dataset ran through Linear Regression with and without regularisation, using Elastic Net as the main regularisation technique. The finalised model for deployment was the Random Forest Regressor, as it outperformed both the Linear models in prediction of the laptop price.

1. **Without Regularisation**
   When the Linear regression was used without any penalty, the model was trained on the data as it is, without controlling the large coefficients or multicollinearity which indicated signs of overfitting. Keeping all the features in the model made the model less interpretable. Also, the performance of the model which was measured by R-squared was not as optimal when compared to the regularised model.

2. **Elastic Net Regularisation**
   When the Linear regression was used with Elastic Net it combined the L1 (Lasso) and L2 (Ridge) regularisation. Also by using *GridSearchCV*, the model optimises the regularisation strength (alpha) and the balance between L1 and L2 (l1_ratio).
   a. **L1 (Lasso):** In L1, some of the coefficients which were not as relevant for the target feature were reduced to zero, simplifying the model making it more interpretable.
   b. **L2 (Ridge):** In L2, some of the coefficients were stabilised by shrinking them depending on multicollinearity without eliminating the features.

   By using Elastic Net we balanced both L1 and L2 which resulted in better R-squared and adjusted R-squared value compared to the unregularised model. It balanced interpretability (using L1) and performance (using L2).

Even though we get better results with regularisation of Linear Regression, Random Forest provided the highest R-squared score by capturing non-linear relationships and interactions. Because of feature selection we were able to get insights on key features which are directly linked to the target variable which was compensated for the lack of coefficient-based interpretability. As a result, even though Elastic Net improved the Linear regression model by balancing performance and interpretability, Random Forest is best suited for this dataset.

Question 3

Impact of L2 regularization on support vector regression performance and interpretability.

Answer

The Support Vector Regression (SVR) model in the code uses L2 regularisation which is controlled by the hyperparameter C. While using GridSearchCV we optimise C, along with Kernel and epsilon to achieve the best performance.

1. **Performance:**
   L2 regularisation in SVR penalises large coefficients, which helps in preventing overfitting in the model. A lower value of C increases regularisation in SVR which helps in generalising the data but risking the problem of underfitting. A higher value of C increases the complexity of the model while risking the problem of overfitting. A very good value of R-squared while using SVR shows that the model was able to effectively capture the non-linear relationships between features. Although it was good, it was slightly lower than Random Forest which was finalised as the deployment model.

2. **Interpretability:**
   SVR models are not dependent on feature coefficients like Linear models, which makes the interpretability a little less direct. L2 regularisation stabilises the data making sure no single feature dominates the output / target variable. However, interpretability in SVR is limited to understanding the support vectors and Kernel effects, where L2 regularisation has little to no involvement in it.

L2 regularisation helps improve the generalisation of the data and makes the SVR model stable ensuring the performance with limited overfitting. As the interpretability of SVR remains less informative compared to Linear model regardless of its regularisation. Random Forest was selected for deployment due to its optimal R-squared score and feature selection insights.

Question 4

If you were to implement random forest regression, then its comparative performance and interpretability with respect to regularized linear regression and regularized support vector regression models.

Answer

The Random Forest Regressor has better results compared to Linear Regression and Support Vector Regressor (SVR) in terms of performance and generalisation, which we can identify based on the best R-squared score.

1. **Performance:**
   a. Elastic Net gave a better score compared to Linear Regression but was not able to identify complex non-linear relationships due to its linear assumptions.
   b. Support Vector Regressor performed better than Linear Regression model with Elastic Net as it was able to identify non-linear relationships between features which resulted in better R-squared score. Even though it is better than Linear it was slightly worse than Random Forest which gave a better R-squared score.
   c. Random Forest Regressor performed better than both Linear and Support Vector Regressor while capturing and understand the non-linear relationships and interactions between features giving us the best R-squared score among all models.

2. **Interpretability:**
   a. Elastic Net is the most interpretable model, providing features coefficients and feature selection using L1 regularisation.
   b. Random Forest doesn't depend on coefficients but gives feature importance metrics, which helps in identifying the key predictors. This trade-off between interpretability and performance makes it less understandable than Elastic Net but is more practical in understanding the feature relevance.
   c. Support Vector Regressor is the least interpretable of all models compared as it doesn't depend on the features but work with support vectors and kernel effects, which makes it difficult to explain the features but gives better performance in real-world applications.

Conclusion:

While Elastic Net offers the best interpretability and SVR makes the best use of non-linear patterns and relationships between features, Random Forest gives the highest predictive accuracy and feature insights using feature importance which makes it the best model to choose for this dataset.

Question 5

Performing a prediction for one of your models using new data.

Answer

Performing a Prediction Using Random Forest Regression
A prediction was performed using the Random Forest Regression, which was trained and tuned for the best prediction and performance.

New Data Input:
['Apple', 'Ultrabook', 16, 32, 'MacOS', 1.4, 3, 3072, 1920, 1, 1, 1, 'Intel', 3.1, 'Core i7', 512, 0, 3, 0, 'AMD', 'Radeon Pro 560']

Pipeline Processing:
The pipeline was created using the encoding (binary, target, and one-hot), scaling and feature reduction (PCA). This is then transformed into DataFrame to feed the trained Random Forest model for prediction.

Prediction Output:
The model predicted the laptop price as:
Predicted Price: €3137.12

Conclusion:
The Random Forest model was able to successfully predict the laptop price for the new data points, demonstrating the utility of the finalised model as deployment-ready model for this task. Its ability to analyse, handle complex relationships and predict the price made it the best choice for accurate price prediction.

Contribution

Team Member	Specific Contributions	Contribution (in %)
Shashank Borse (20038928)	Implemented and optimised Support Vector Regression and Random Forest models; also, the implementation of the final prediction pipeline. Wrote answers for Questions 3 - 5 and reviewed the final report to ensure flow and accuracy.	50%
Shudhanshu Desai (20039165)	Implemented and optimised Linear Regression models (with and without Elastic Net), data preprocessing, including encoding and PCA, and calculation of R-squared scores. Wrote answers for Questions 1 and 2, making sure that the answers match the code results.	50%