Research AI- Task 5:

Review Questions- 20th October:

1. Where do we use standardisation and normalisation?
   When to Use Standardization:
   - Algorithms sensitive to feature scale: Algorithms like support vector machines (SVMs), linear regression, and gradient descent often benefit from standardization. This is because they are sensitive to the scale of features, and standardization can help prevent features with larger magnitudes from dominating the learning process.
   - Data with Gaussian distribution: Standardization is often preferred when the data is assumed to be normally distributed. It transforms the data to have a mean of 0 and a standard deviation of 1, which is a standard normal distribution.

   When to Use Normalization:

   - Distance-based algorithms: Algorithms like k-nearest neighbors (KNN) and clustering techniques that rely on distance calculations often require normalized data. Normalization ensures that all features are on a scale between 0 and 1, preventing features with larger ranges from dominating the distance calculations.
   - Data with outliers: Normalization can be helpful in dealing with outliers, as it scales the data to a specific range. However, it's important to be aware that normalization can be sensitive to outliers, as they can significantly affect the minimum and maximum values.

2. What is binning?
   Binning is a data preprocessing technique used to categorize continuous numerical data into discrete intervals or bins. This process is also known as quantization or bucketing.
   Binning is used to:
   - Handle large datasets: Binning can reduce the size of a dataset by grouping similar values together, making it easier to manage and analyze.
   - Improve algorithm performance: Some algorithms, like decision trees and some clustering algorithms, work better with categorical data. Binning can convert continuous data into categorical form.
   - Handle outliers: Binning can help mitigate the impact of outliers by grouping them into specific bins.

- Visualize data: Binning can make it easier to visualize data distributions using histograms or bar charts.

How does binning work?

- Determine the number of bins: The number of bins depends on the desired level of granularity. Too few bins may obscure important patterns, while too many bins may make the data too granular.
- Define bin edges: The edges of each bin are defined. This can be done using equal-width intervals or equal-frequency intervals.
- Assign data points to bins: Each data point is assigned to the bin that it falls into based on its value.

Types of binning:

- Equal-width binning: Bins have equal width.
- Equal-frequency binning: Bins contain approximately the same number of data points.
- K-means binning: Uses the k-means clustering algorithm to determine bin boundaries.

3. Difference between metric and cost function

   Metric:

   - Purpose: A metric is a quantitative measure used to assess the performance of a model on a specific task. It provides a way to evaluate how well the model is doing in comparison to a baseline or other models.
   - Examples: Accuracy, precision, recall, F1-score, mean squared error (MSE), mean absolute error (MAE).

   Cost function:

   - Purpose: A cost function is a mathematical function that quantifies the "error" or "discrepancy" between a model's predicted values and the actual values. It's used to guide the model's training process by minimizing the cost.
   - Examples: Mean squared error (MSE), cross-entropy loss, hinge loss.

   Key Differences:

   - Purpose: Metrics evaluate model performance, while cost functions guide model training.
   - Usage: Metrics are typically used after training to assess the final performance of a model, while cost functions are used during training to update the model's parameters.
   - Relationship: Cost functions are often used as metrics to evaluate model performance, but not all metrics are directly derived from cost functions.

4. R2 Score, F1 Score and Adjusted R2 Score

   R2:

   R-squared (Coefficient of Determination): R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform. In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context. So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides. The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically, R2 squared calculates how must regression line is better than a mean line. Hence, R2 squared is also known as Coefficient of Determination or sometimes also known as Goodness of fit.

$$\textbf{R2 Squared} = 1 - \frac{SSr}{SSm}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

Adjusted R2:

The disadvantage of the R2 score is while adding new features in data the R2 score starts increasing or remains constant but it never decreases because It assumes that while adding more data variance of data increases. But the problem is when we add an irrelevant feature in the dataset then at that time R2 sometimes starts increasing which is incorrect. Hence, to control this situation Adjusted R Squared came into existence.

$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right) \times (1 - R^2)\right]$$

where:
n = number of observations
k = number of independent variables
R_a^2 = adjusted R^2

F1 Score:

F1-score is a harmonic mean of precision and recall, providing a single metric that balances both measures. It's particularly useful when there is an imbalance between positive and negative classes in the dataset.

Precision and Recall:

- Precision: Measures how many of the positive predictions made by the model were actually correct. It's calculated as:

  Precision = True Positives / (True Positives + False Positives)

- Recall: Measures how many of the actual positive instances were correctly predicted by the model. It's calculated as:

  Recall = True Positives / (True Positives + False Negatives)

F1-Score Calculation:

The F1-score is calculated as the harmonic mean of precision and recall:

F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

Interpretation:

- Higher F1-score: Indicates better overall performance, balancing precision and recall.
- Lower F1-score: Suggests a need to improve either precision or recall, or both.
- When to Use F1-Score
- Imbalanced datasets: When the number of positive and negative examples is significantly different.
- Equal importance of precision and recall: When both precision and recall are important for the task.

5. Correlation Matrix
   - A correlation matrix is a table that shows the pairwise correlations between variables in a dataset. It's a valuable tool for understanding the relationships between different features.
   - Values: Correlation coefficients range from -1 to 1:
     1) -1: Perfect negative correlation (when one variable increases, the other decreases)
     2) 0: No correlation
     3) 1: Perfect positive correlation (when one variable increases, so does the other)
   - Visualization: Correlation matrices can be visualized as heatmaps, where colors represent the strength and direction of the correlations.

6. What is regularisation?
   Regularization is a technique used in machine learning to prevent overfitting. Overfitting occurs when a model becomes too complex and learns the training data too well, leading to poor performance on new, unseen data. Regularization helps to address this by introducing a penalty term to the loss function.
   Common regularization techniques:
   - L1 regularization (Lasso):
   1) Adds a penalty term proportional to the absolute value of the weights.
   2) Tends to produce sparse models, where many weights are zero.
   3) Useful for feature selection.
   - L2 regularization (Ridge):
   1) Adds a penalty term proportional to the square of the weights.
   2) Tends to produce models with smaller weights.
   3) Helps to prevent overfitting by reducing the influence of any single feature.
   - Elastic Net:
   1) Combines L1 and L2 regularization.
   2) Provides a balance between feature selection and reducing model complexity.

7. Do we use accuracy in linear regression?
   In linear regression, accuracy is not typically used as a performance metric because it is more relevant for classification tasks, where the goal is to predict discrete class labels. Instead, linear regression focuses on continuous outcomes, and other metrics are used to evaluate model performance, like MSE, RMSE, MAE, R2.

Revised Linear Regression:

The dataset consists of 205 rows and 26 columns, where each row represents a unique car and each column describes a feature of the car, with the goal of predicting the car's price. The dependent variable is the price, while the independent variables include features like symboling, CarName, fueltype, aspiration, doornumber, carbody, drivewheel, enginelocation, and wheelbase, among others. The dataset is split into an 80-20 train-test split to maximize predictive accuracy. Categorical features like symboling, fueltype, aspiration, doornumber, carbody, drivewheel, and enginelocation were likely converted into numerical representations using label encoding. A correlation matrix was created to analyze the relationships between features. This helps identify any highly correlated features that might introduce redundancy and improve model efficiency. A heatmap

visualization of the correlation matrix provides a visual representation of the relationships, making it easier to identify strong positive or negative correlations. Based on the correlation matrix, features with low correlations to the target variable (price) have been dropped. A threshold can be set to determine which features are considered "low correlated." This helps reduce the dimensionality of the data and potentially improve model performance by focusing on the most relevant features. Features were normalized to scale them within a similar range. This helps prevent features with larger magnitudes from dominating the learning process and improves model convergence. The model's cost function is defined and optimized using gradient descent to minimize error, ensuring predictions are close to actual values. The chosen algorithm's hyperparameters (e.g., learning rate, regularization strength) might have been tuned to optimize performance. Finally, various evaluation metrics are employed to assess model performance and accuracy, providing insight into how closely the model's predictions align with the actual car prices.

Review Questions:

1. How to identify outliers?
   When exploring data, the outliers are the extreme values within the dataset. That means the outlier data points vary greatly from the expected values—either being much larger or significantly smaller. For data that follows a normal distribution, the values that fall more than three standard deviations from the mean are typically considered outliers.
   Finding outliers in your data should follow a process that combines multiple techniques performed during your exploratory data analysis. I recommend following this plan to find and manage outliers in your dataset:
   - Use data visualization techniques to inspect the data's distribution and verify the presence of outliers - histogram, boxplot, scatterplot
   - Use a statistical method to calculate the outlier data points - Using the IQR, the outlier data points are the ones falling below Q1–1.5 IQR or above Q3 + 1.5 IQR.
   - Apply a statistical method to drop or transform the outliers.

2. What is standard deviation?
   - The standard deviation is a measure of how the values in your data differ from one another or how spread out your data is.
   - The standard deviation measures how far apart the data points in your observations are from each. You can calculate it by subtracting each data point from the mean

value and then finding the squared mean of the differenced values; this is called Variance. The square root of the variance gives you the standard deviation.

- Like how the mean tells you where the data is centered, the standard deviation gives you the width of your bell curve. It tells you how narrow or wide the bell curve is.

3. Metrics for Linear and Logistic Regression
   Evaluation Metrics for regression are essential for assessing the performance of regression models specifically. These metrics help in measuring how well a regression model is able to predict continuous outcomes. Common regression evaluation metrics for regression include:
   - Mean Absolute Error (MAE): MAE is a very simple metric which calculates the absolute difference between actual and predicted values. Take the sum of all the errors and divide them by a total number of observations
   - Mean Squared Error (MSE): It represents the squared distance between actual and predicted values. we perform squared to avoid the cancellation of negative terms and it is the benefit of MSE
   - Root Mean Squared Error (RMSE): It is a simple square root of mean squared error.
   - R-squared (Coefficient of Determination): R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform. In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context. So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides. The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically, R2 squared calculates how must regression line is better than a mean line. Hence, R2 squared is also known as Coefficient of Determination or sometimes also known as Goodness of fit.

4. What is deterministic and statistical data?
   Mathematical models can be classified as either deterministic models or statistical models.
   - A deterministic model is a mathematical model in which the output is determined only by the specified values of the input data and the initial conditions. This means that a given set of input data will always generate the same output.
   - A statistical model is a mathematical model in which some or all of the input data have some randomness, for example as expressed by a probability distribution, so that for a given set of input data the output is not reproducible but is described by a probability distribution. The output ensemble is obtained by running the model a large number of times with a new input value sampled from the probability distribution each time the model is run.

5. Assumptions we make before performing Linear Regression

- Linear Relationship: The core premise of multiple linear regression is the existence of a linear relationship between the dependent (outcome) variable and the independent variables. This linearity can be visually inspected using scatterplots, which should reveal a straight-line relationship rather than a curvilinear one.
- Multivariate Normality: The analysis assumes that the residuals (the differences between observed and predicted values) are normally distributed. This assumption can be assessed by examining histograms or Q-Q plots of the residuals, or through statistical tests such as the Kolmogorov-Smirnov test.
- No Multicollinearity: It is essential that the independent variables are not too highly correlated with each other, a condition known as multicollinearity. This can be checked using correlation matrices, where correlation coefficients should ideally be below 0.80.
- Variance Inflation Factor (VIF): With VIF values above 10 indicating problematic multicollinearity. Solutions may include centering the data (subtracting the mean score from each observation) or removing the variables causing multicollinearity.
- Homoscedasticity: The variance of error terms (residuals) should be consistent across all levels of the independent variables. A scatterplot of residuals versus predicted values should not display any discernible pattern, such as a cone-shaped distribution, which would indicate heteroscedasticity. Addressing heteroscedasticity might involve data transformation or adding a quadratic term to the model.

6. Types of Gradient Descent
    - Batch Gradient Descent
      Iterations: Each iteration involves passing the entire dataset through the model to calculate the gradient, updating the model parameters once per epoch.
      Epochs: One epoch is completed after the entire dataset has been processed through a single iteration.
      Pros: Stable and often leads to a smoother convergence.
      Cons: Requires a lot of memory and computation time, especially with large datasets, as it processes all data at once.

    - Mini-Batch Gradient Descent
      Iterations: The dataset is divided into small batches, and each batch is used in one iteration. The model parameters are updated multiple times per epoch (based on the number of batches).
      Epochs: One epoch is completed after all mini-batches have been processed. The number of iterations per epoch is equal to the number of mini-batches.
      Pros: Offers a balance between computational efficiency and convergence stability. It also allows for faster training and benefits from parallel processing.
      Cons: Less stable than batch gradient descent but generally more stable than stochastic gradient descent.

- Stochastic Gradient Descent (SGD)
  Iterations: Each iteration processes a single data point, and the model parameters are updated after every data point.
  Epochs: One epoch is completed after every data point in the dataset has been processed individually.
  Pros: Highly efficient in terms of memory usage and can escape local minima due to frequent updates.
  Cons: Can lead to more noisy updates and a less stable convergence compared to batch and mini-batch methods.

7. What iterations and epochs?
   - Epochs: An epoch is one complete pass through the entire training dataset. Number of times the entire dataset is passed through the model
   - Iterations: An iteration is a single update of the model's parameters. Number of parameter updates, which may be multiple times within each epoch if using mini-batches
8. Standardisation Vs Normalisation:
   - Normalization or Min-Max Scaling: It is used to transform features to be on a similar scale. The new point is calculated as:
     X_new = (X - X_min)/(X_max - X_min)
     This scales the range to [0, 1] or sometimes [-1, 1]. Geometrically speaking, transformation squishes the n-dimensional data into an n-dimensional unit hypercube. Normalization is useful when there are no outliers as it cannot cope up with them.

   - Standardization or Z-Score Normalization: It is the transformation of features by subtracting from mean and dividing by standard deviation. This is often called as Z-score.
     X_new = (X - mean)/Std
     Standardization can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Geometrically speaking, it translates the data to the mean vector of original data to the origin and squishes or expands the points if std is 1 respectively. We can see that we are just changing mean and standard deviation to a standard normal distribution which is still normal thus the shape of the distribution is not affected. Standardization does not get affected by outliers because there is no predefined range of transformed features.

9. What makes MSE, RMSE, and other loss functions good loss functions?
   - Mean Squared Error (MSE):
     Smoothness: Differentiable, suitable for gradient-based optimization.

Penalty on Large Errors: Heavily penalizes large deviations due to squaring, ideal for applications needing to minimize large errors.
Convexity: Guarantees a unique minimum, aiding in consistent optimization.

- 2. Root Mean Squared Error (RMSE):
  Interpretability: Outputs error in the same unit as the target, making it more understandable.
  Sensitivity to Outliers: Penalizes large errors like MSE but provides interpretable error magnitudes.
- 3. Other Loss Functions:
  Mean Absolute Error (MAE): Less sensitive to outliers, useful when all errors are equally significant.

Linear Regression:

The dataset consists of 205 rows and 26 columns, where each row represents a unique car and each column describes a feature of the car, with the goal of predicting the car's price. The dependent variable is the price, while the independent variables include features like symboling, CarName, fueltype, aspiration, doornumber, carbody, drivewheel, enginelocation, and wheelbase, among others. The dataset is split into an 80-20 train-test split to maximize predictive accuracy. One-hot encoding is applied to transform categorical features into a numerical format suitable for the model, particularly for machine learning algorithms that require numerical inputs, and normalization is used to scale features within a similar range, which improves model training by standardizing the inputs. The model’s cost function is defined and optimized using gradient descent to minimize error, ensuring predictions are close to actual values. Finally, various evaluation metrics are employed to assess model performance and accuracy, providing insight into how closely the model's predictions align with the actual car prices.

GitHub repo:

https://github.com/MSarayu512/Sarayu_MRM