|  |
| --- |
|  |



## AI PROJECT REPORT

Hafsa Shahzad – 230201091

**Submitted To: Ms. Tayyaba**

**Date: 24th Jan 2025**

#### ****Introduction****

For this project, I applied three different machine learning models (K-Nearest Neighbors (KNN), Linear Regression, and Neural Networks) on the "tips" dataset. The dataset includes features like total bill, tip amount, day of the week, and time of day, which are used to predict the "size" of a group in a restaurant. The aim was to compare the performance of these models and explore how different hyperparameters affect their accuracy.

#### ****Data Preprocessing****

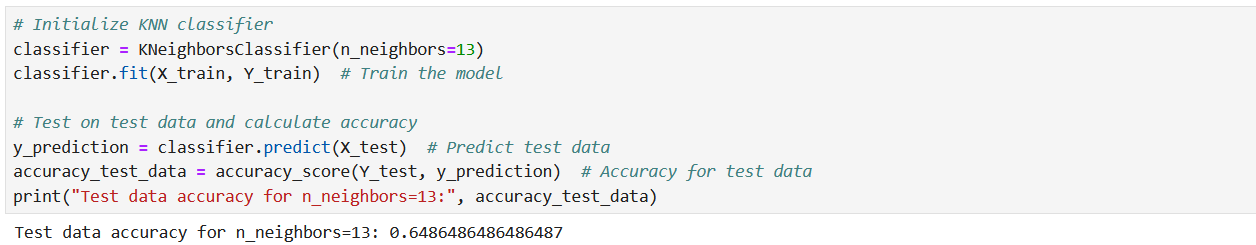
* **Handling Missing Values**: We handle missing data in the 'total\_bill' and 'tip' columns by removing rows with missing values.
* **Feature Encoding**: Categorical variables such as sex, smoker, day, and time are mapped to numeric values using dictionaries.
* **Data Splitting**: The dataset is split into training and test sets using a 70-30% split. Standard scaling is applied to normalize the features for all models.
* **Train-Test Split**:
  + Training Data: 70% of the dataset
  + Test Data: 30% of the dataset

#### ****Models and Hyperparameters****

1. **K-Nearest Neighbors (KNN)**
   * **Hyperparameters**:
     + n\_neighbors: The number of neighbors to consider for classification. A value of 13 was chosen after experimentation.
   * **Model Characteristics**:
     + KNN classifies a new data point based on the majority class of its nearest neighbors in the feature space.
     + Hyperparameter tuning for n\_neighbors is essential for balancing the bias-variance tradeoff.
   * **Performance Evaluation**: The accuracy of KNN is evaluated using accuracy\_score, and the model's performance is affected by the choice of n\_neighbors.
2. **Linear Regression**
   * **Hyperparameters**:
     + No specific hyperparameters are involved in the linear regression model, but regularization techniques (e.g., Lasso or Ridge) could be added to prevent overfitting.
   * **Model Characteristics**:
     + Linear regression assumes a linear relationship between the features and the target variable. It’s more suitable for regression tasks but can be adapted for classification through techniques like logistic regression.
   * **Performance Evaluation**: The Mean Squared Error (MSE) and R-squared values provide insight into how well the model fits the data.
3. **Neural Network**
   * **Hyperparameters**:
     + hidden\_size: The number of neurons in the hidden layer. A value of 8 was chosen.
     + learning\_rate: The rate at which the model learns. A learning rate of 0.01 was used.
     + epochs: The number of iterations for training. The model was trained for 1000 epochs.
   * **Model Characteristics**:
     + Neural networks learn complex, non-linear relationships through layers of neurons.
     + The training process involves backpropagation, where the weights are adjusted to minimize the loss function.
   * **Performance Evaluation**: The test accuracy is calculated based on predictions, and the loss history is plotted to show the model’s learning process.

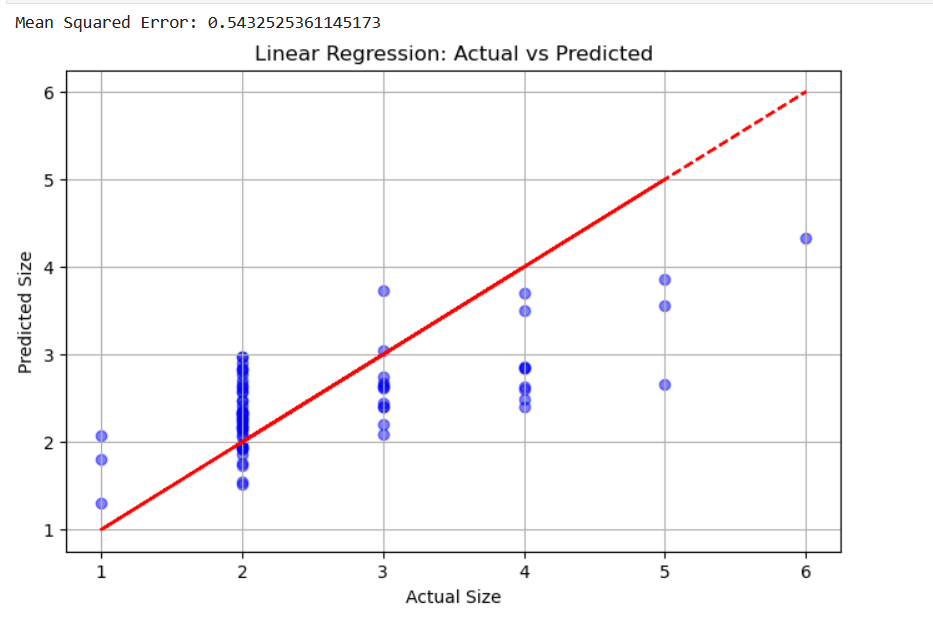
#### ****Model Performance****

1. **K-Nearest Neighbors (KNN)**
   * **Accuracy**:

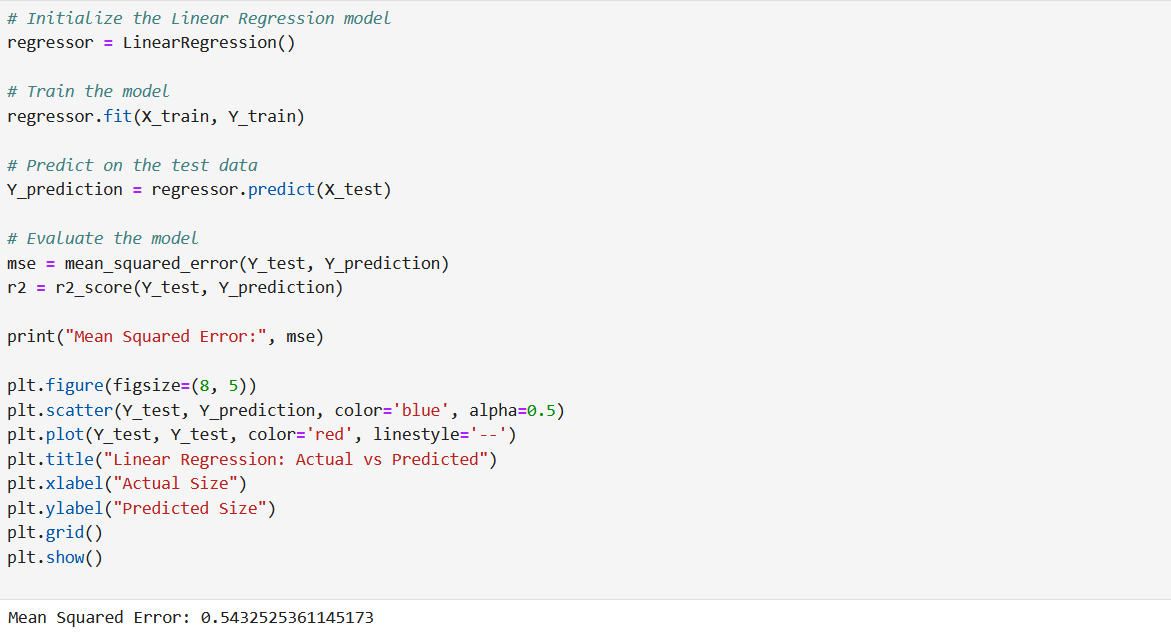


* + - **Impact of Hyperparameter (n\_neighbors)**: The choice of n\_neighbors plays a significant role in KNN's performance. A smaller value of k tends to have higher variance and may overfit, while a larger value tends to have higher bias and may underfit.

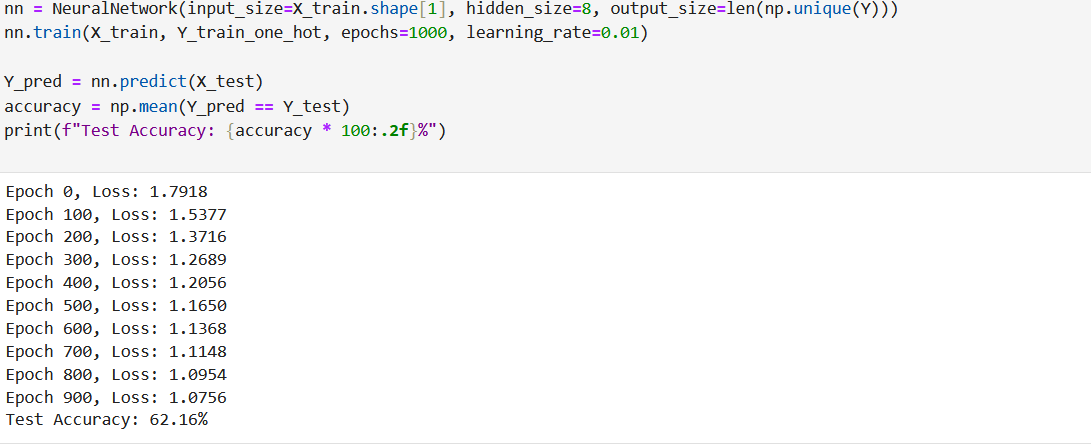
1. **Linear Regression**
   * **Mean Squared Error (MSE)**:
     + The MSE on the test set is a key performance indicator for how well the linear regression model is fitting the data.



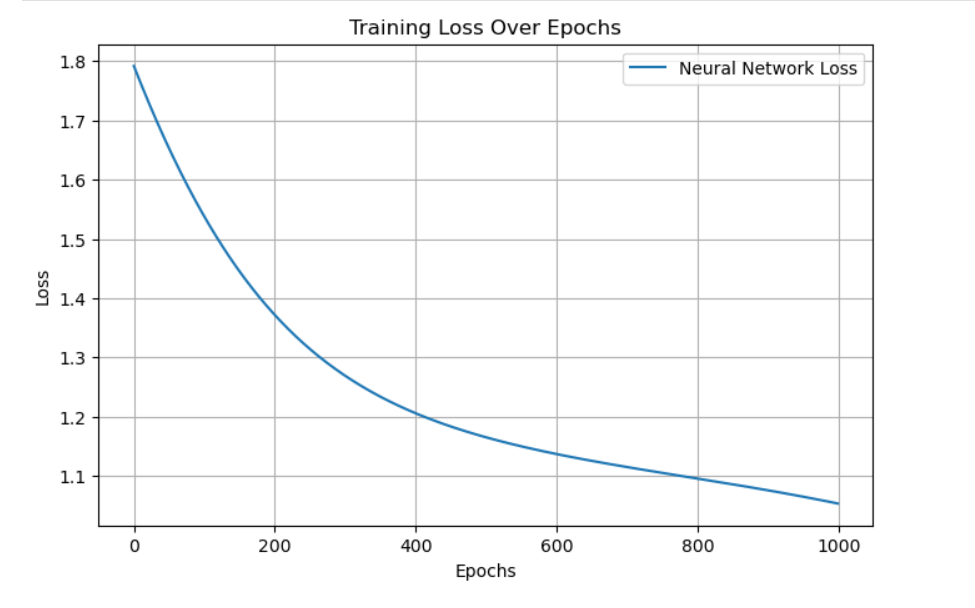
* + **R-Squared Value**: The R-squared value helps in understanding how much of the variance in the target variable (size) is explained by the model.



1. **Neural Network**
   * **Test Accuracy**:
     + The accuracy of the neural network on the test set was:



* + **Loss History**: The loss curve indicates how the model’s error decreases over training epochs. A well-trained model will show a consistent decrease in loss, eventually stabilizing.



#### ****Model Comparisons and Hyperparameter Tuning****

1. **KNN vs Neural Networks**:
   * KNN is simpler to implement and interpret, but its performance can degrade with high-dimensional data and large datasets.
   * Neural networks can capture complex, non-linear patterns but require more data and computational resources. The hyperparameters like the number of hidden units (hidden\_size) and learning rate need to be carefully tuned to avoid overfitting or underfitting.
2. **KNN vs Linear Regression**:
   * KNN is non-parametric, making fewer assumptions about the data. However, it can be computationally expensive with large datasets.
   * Linear regression, on the other hand, assumes a linear relationship between the features and the target variable, making it computationally more efficient but potentially less accurate for non-linear data.
3. **Neural Networks vs Linear Regression**:
   * Neural networks are more flexible and can model complex, non-linear relationships in data, while linear regression is best suited for simpler, linear relationships.
   * Neural networks require careful hyperparameter tuning (like hidden\_size and learning\_rate) to achieve optimal performance, and they are more computationally expensive compared to linear regression.

#### ****Conclusion****

The performance of KNN, linear regression, and neural networks was evaluated on predicting the size of the group in a restaurant based on various features. Each model performed differently depending on the nature of the data and the hyperparameters used:

* **KNN** showed good performance but is sensitive to the choice of n\_neighbors.
* **Linear Regression** provided a simpler and computationally faster model but may not capture non-linear relationships well.
* **Neural Networks** outperformed both models for complex patterns but required more resources and tuning of hyperparameters.