**Model Refinement**

1. **Overview**

The model refinement phase plays a crucial role in the machine learning workflow, shifting the focus from model construction to performance optimization. While the initial model exhibited errors, subsequent iterations combining hyperparameter tuning and feature engineering yielded optimal results based on established evaluation metrics. For instance, the linear regression model achieved a perfect fit to the data, while the ANN model initially displayed significantly higher errors in MSE, MAE, and R-squared. This aligns with the known tendency of linear regression models to improve fit with increasing feature count. After a thorough data examination, less significant features were eliminated, leading to satisfactory results for both models.

1. **Model Evaluation**

Based on the initial model construction, the following results were observed. First the regression model exhibited perfect fit for all evaluation metrics. In practice this is extremely unlikely to occur. On the other hand, ANNs model has high R-squared while the MSE and MAE were reasonably satisfactory. On the k-mean clustering algorithm, within cluster sum of squares evaluation metrics were used to determine the optimal number of clusters. There results of the linear regression model and artificial neurol networks are presented in table 1.

1. **Refinement Techniques**

To address both multicollinearity and overfitting, several parameters were adjusted to refine and enhance the utility of the regression model. Firstly, several variables were removed due to their low correlation with the target variable, thereby reducing multicollinearity. Secondly, a regularization parameter was introduced to the model, which shrinks the coefficients, leading to reduced variance and potentially mitigating overfitting and multicollinearity. Tuning the regularization parameter (lambda) controls the trade-off between model complexity and generalizability. Similarly, in the artificial neural network (ANN) model, several variables were removed, and optimizer settings were adjusted accordingly. Using the same variables as both the linear regression and ANN models, K-means clustering was applied. Additionally, principal component analysis (PCA), a dimensionality reduction technique, was used for plotting clusters.

1. **Hyperparameter Tunning**

In the artificial neural network (ANN) model, several hyperparameters were optimized. These parameters included the learning rate, the number of hidden layers, the number of neurons per layer, the dropout rate, and the optimizer. After optimizing these parameters, a configuration with a single hidden layer, 8 neurons per layer, a learning rate of 0.01, and the SGD optimizer was identified as optimal based on the Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics.

1. **Cross-Validation**

Initially, standard 5-fold cross-validation was employed for the linear regression model. Although a common and reliable technique for linear models, it offers unbiased and relatively low-variance performance estimates. However, a potential drawback lies in the high variance across folds, leading to potential bias if folds do not perfectly represent the data distribution. This led to the adoption of 10-fold cross-validation in the final model.

1. **Feature Selection**

To select the most relevant features and identify and remove redundant or irrelevant ones, a correlation analysis was performed. This approach helped to reduce model complexity and potentially improve generalizability.

**Test Submission**

1. **Overview**

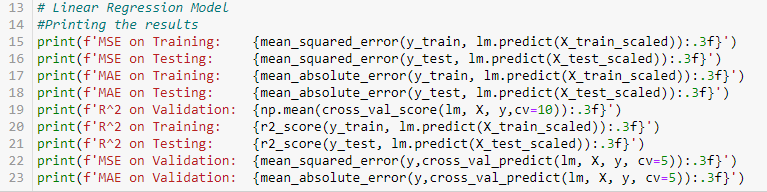
To evaluate the performance and usability of the proposed model, a test dataset was used. Prior to training, the dataset was split into training and test subsets. The training set was used to train and validate the model, while the separate test set was used to assess its performance on unseen data.

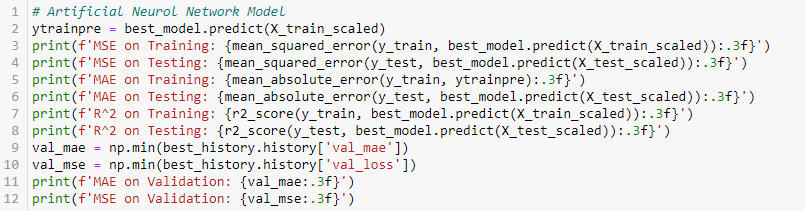
1. **Data Preparation for Testing**

The test dataset was prepared for training by inputting missing values, removing outliers, applying necessary transformations, removing redundant features, and normalizing the data.

1. **Model Application**

After training, the model was applied to the test dataset to generate predictions for unseen data points. These predictions were then evaluated using appropriate evaluation metrics. Here is some code snippets from the relevant part.





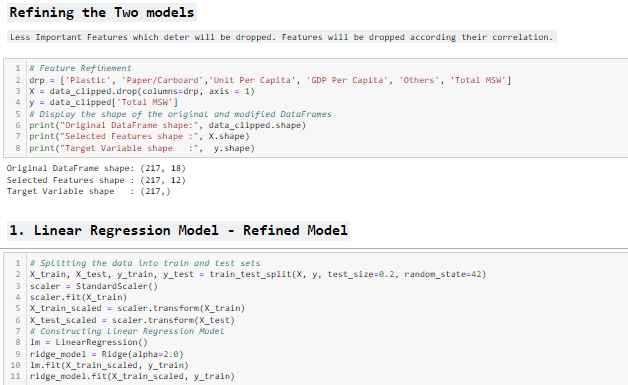
1. **Test Metrics**

The table below show the comparison between the test, train and validation performance.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dataset Type | Linear Regression Model | | | Artificial Neurol Network Model | | |
| Evaluation Metrics | | | | | |
| MSE | MAE | R-squared | MSE | MAE | R-squared |
| Training | 0.198 | 0.319 | 0.961 | 0.443 | 0.526 | 0.913 |
| Validation | 0.217 | 0.342 | 0.953 | 0.427 | 0.470 | - |
| Testing | 0.188 | 0.324 | 0.961 | 0.272 | 0.417 | 0.944 |

1. **Code Implementation**

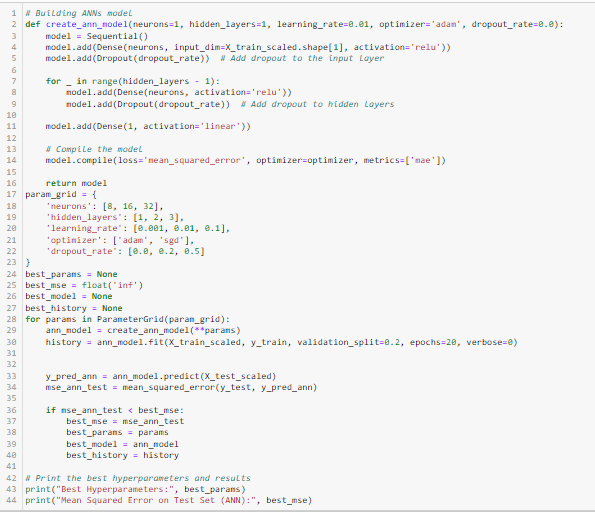
Here is the relevant code snippets for both model refinement and test submission phases.





A computer screen shot of a computer code

Description automatically generated



1. **Conclusion**

Through the utilization of multiple linear regression and artificial neural network models, the study identified and analyzed primary factors influencing urban waste generation patterns. Notably, the results indicated that the multiple linear regression model outperformed the ANN model in predicting urban waste generation. Additionally, clustering algorithms, particularly K-means clustering, effectively categorized urban areas into distinct groups based on waste generation characteristics. These clusters provide valuable insights for targeted interventions and collaborative efforts to enhance sustainable waste practices globally. As future directions, the project suggests exploring additional features, incorporating geospatial and temporal analyses, evaluating advanced modeling techniques, and collaborating with stakeholders to further refine waste management predictions and strategies.

1. **References**

*No external sources were used other than python libraries.*