ITSOLERA PVT LTD

AUTO FUEL EFFICIENCY PREDICTION

Executive Summary

The "Predict Fuel Efficiency with Machine Learning" project aims to develop a model that predicts a car's fuel economy based on factors like weight and engine size. By using advanced machine learning to analyze historical data, the project seeks to identify key factors affecting fuel use, aiding consumers and manufacturers in making informed, sustainable transportation decisions.

PROJECT BY

TEAM ZETA

Team Lead

Mahnoor Khawar

teamzeta@itsolera.com
E-mail Address

+123-456-7890
Phone Number

123 Anywhere St., Any City, ST 12345
Official Address

INTRODUCTION

The goal of the "Predict Fuel Efficiency with Machine Learning" project is to create a prediction model that can calculate a car's fuel economy based on several factors, including weight, engine size, horsepower, and more. The project seeks to find important factors impacting fuel use by analysing historical data using advanced machine learning methods. This initiative advances the development of sustainable transportation solutions while highlighting the value of machine learning in automotive engineering. The goal of the results is to help consumers and manufacturers alike make better decisions about the design and purchasing of vehicles by offering useful information.

Motivation

This project's motivation is the increasing need to tackle climate change and maximize energy use. Reducing greenhouse gas emissions, helping automakers in creating environmentally friendly automobiles, and assisting consumers in making informed choices can all be greatly helped by accurate fuel efficiency estimates. Moreover, the project's observations can help authorities create rules that encourage environmental sustainability and energy conservation.

OUR TEAM

- MAHNOOR KHAWAR (TEAM LEAD)
- JAWAD
- AMNA
- Fahad
- Akif Mehmood Dahri
- Tayyaba
- Ibrahim Jawad
- Talha Tariq

METHEDOLOGY

The project involved several key steps. Data Collection used Pandas, OS, and NumPy to handle and verify data files. Data Preprocessing included filling missing values, encoding categorical variables, and scaling numerical features. Feature Engineering created new features and applied logarithmic transformations. Model Selection tested algorithms like Linear Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, and Neural Networks. Model Training and Evaluation used a scikit-learn Pipeline, followed by performance evaluation with metrics like R^2 score and mean squared error.

Data Collection

The necessary libraries are imported which include Pandas to manipulate data, OS is used to interact with the operating system and read data files, and
NumPy is used for numerical operations. The input data files are in the read only "/Kaggle/input" directory. In our project, we print the path to each file as
we go through the directory containing the data files. It helps in the data files' identification and verification. To put it simply, before moving on to preprocessing, we checked to make sure the data files were loaded successfully and that the paths were appropriately referenced. We also verified the data integrity and structure. The source from where we have collected data is given bellow:

Data Collection

https://co2cars.apps.eea.europa.eu/?source=%7B%22track_total_hits%22%3Atrue%2C%22query%22%3A%7B%22bool%22%3A%7B%22must%22%3A%5B%7B%22constant_score%22%3A%7B%22filter%22%3A%7B%22bool%22%3A%7B%22must%22%3A%5B%7B%22bool%22%3A%7B%22should%22%3A%5B%7B%22term%22%3A%7B%22year%22%3A2023%7D%7D%5D%7D%7D%2C%7B%22bool%22%3A%7B%22should%22%3A%5B%7B%22term%22%3A%7B%22scStatus%22%3A%22Provisional%22%7D%7D%5D%7D%7D%5D%7D%7D%7D%7D%5D%7D%7D%2C%22display_type%22%3A%22tabular%22%7D

DATA PREPROCCESSING

Data preprocessing is used for ensuring the quality and accuracy of the data used in the model. The dataset had been imported from a CSV file into a pandas Data Frame. The preprocessing steps listed below are done:

01

Handling Missing Values:

The means of the corresponding columns were used to fill in the blanks after each column's values were examined for missing values.

02

Encoding Categorical Variables:

Since machine learning models need numerical input, categorical columns were encoded into numerical values.

03

Scaling Numerical Features:

The features' means and standard deviations to zero and one, respectively. This scaling helps in the model's faster convergence.

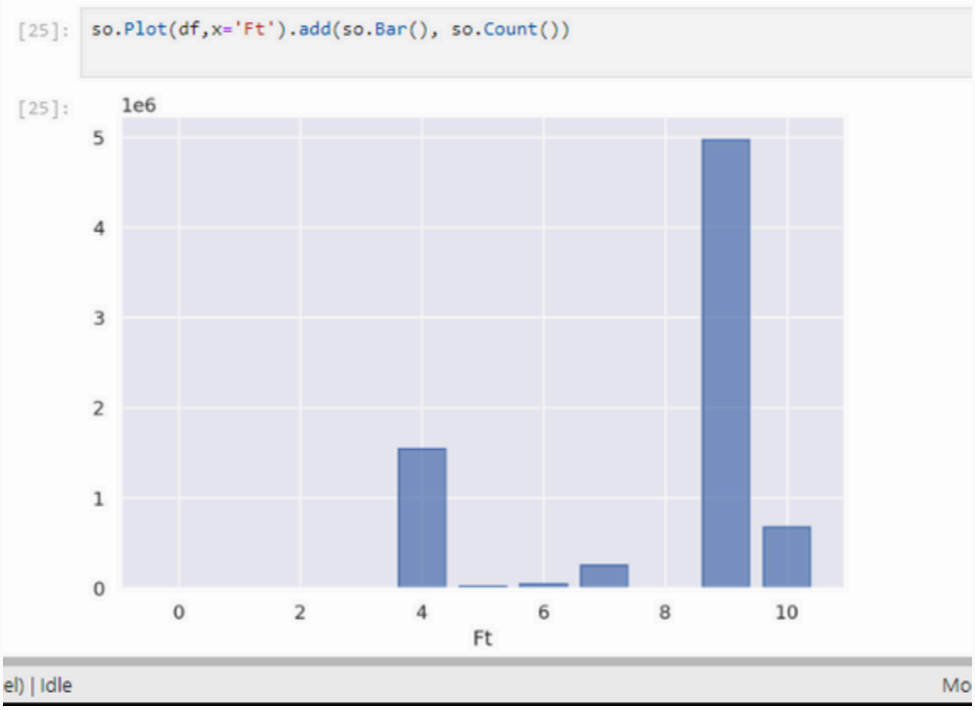
The data preprocessing stage effectively handled missing values, encoded categorical variables, and scaled numerical features, ensuring a clean and standardized dataset. These steps were crucial in preparing the data for accurate and efficient model training, ultimately enhancing the reliability and performance of the machine learning models used in the project.

FEATURE ENGINEERING

Two feature engineering pipelines are set up using the given code to preprocess data before the model training. In order to avoid multicollinearity, the encoder_pipe pipeline drops the first category and ignores unknown categories while converting categorical data into a binary matrix using OneHotEncoder. The standardization of numerical features is achieved by the scaling_pipe pipeline using StandardScaler, which ensures that each feature has a mean of 0 and a standard deviation of 1. The features are then mapped to a uniform or normal distribution using Quantile Transformer. When combined, these processes make sure that the data is appropriately scaled and converted for both numerical and categorical variables, improving its suitability for machine learning models.

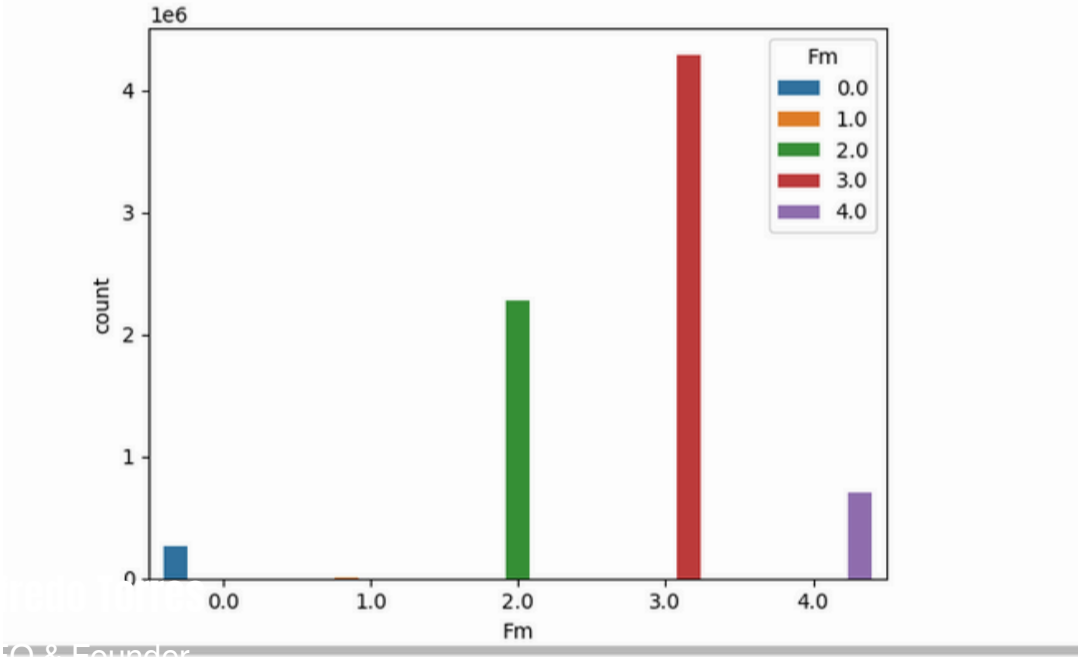
DATA GRAPHS

Ft

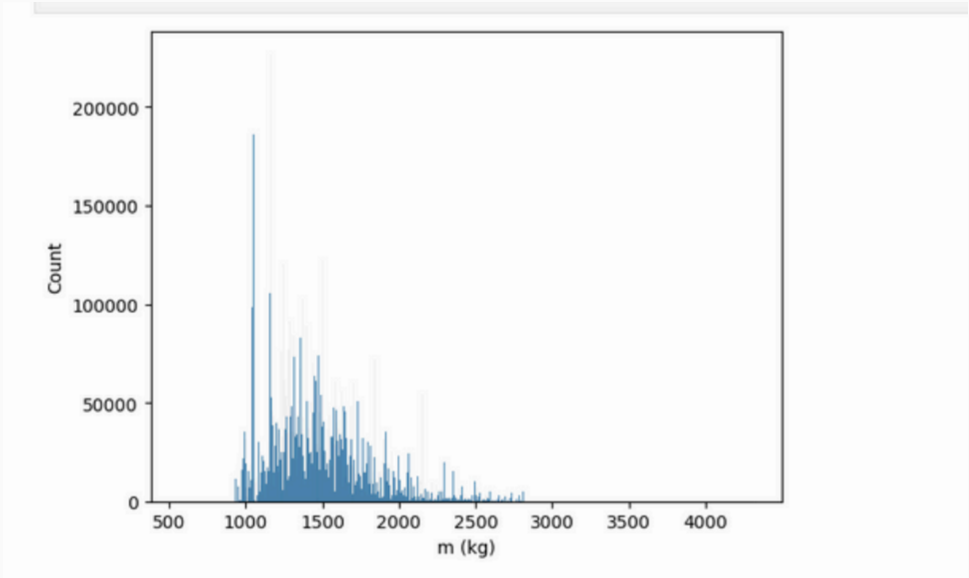


Fm

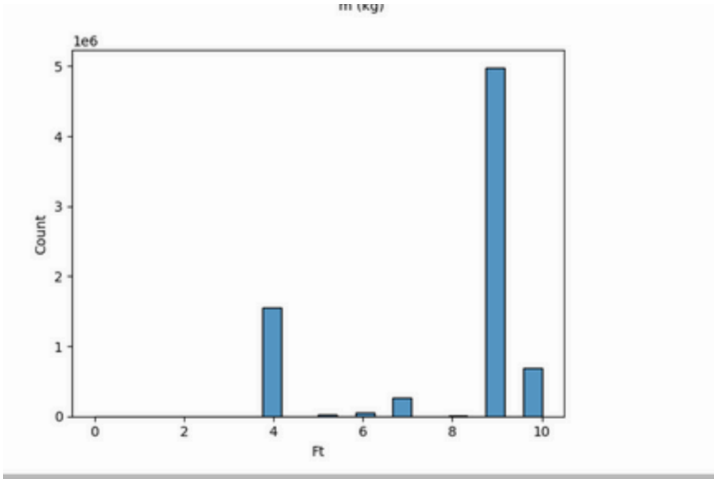
```
[91]: sns.countplot(df,x='Fm',hue='Fm')
      plt.show()
```

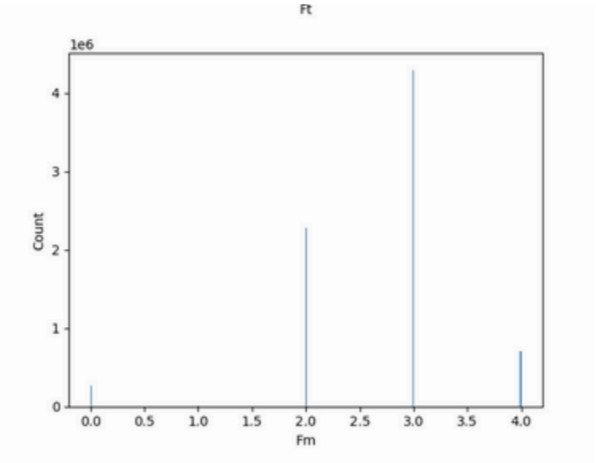


Mass

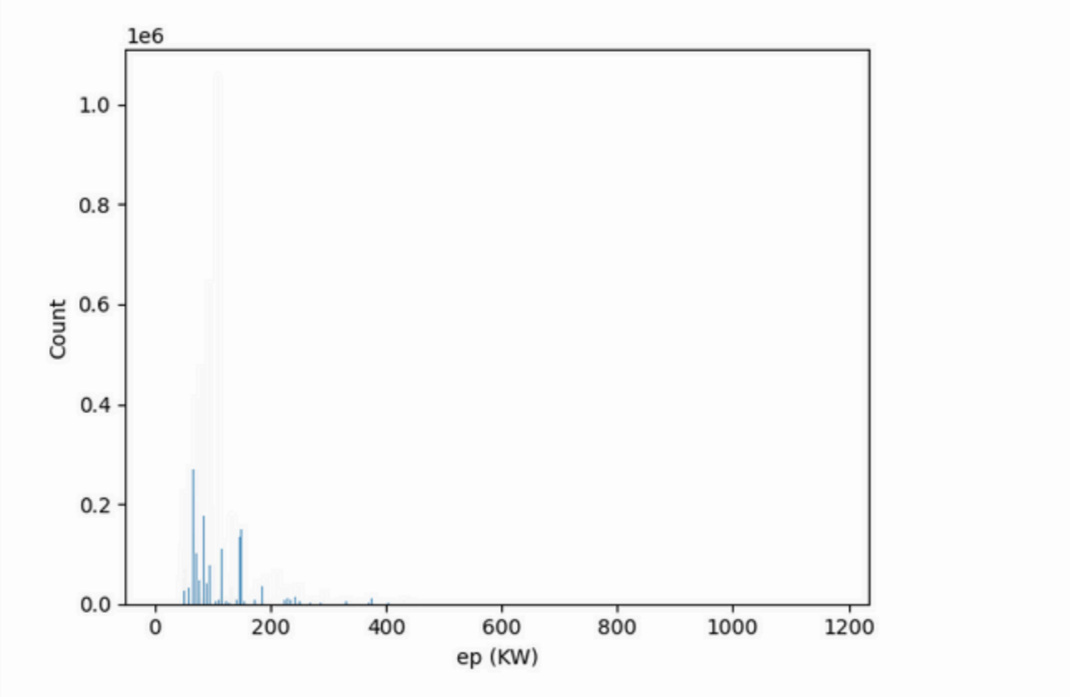


Fm and Fm

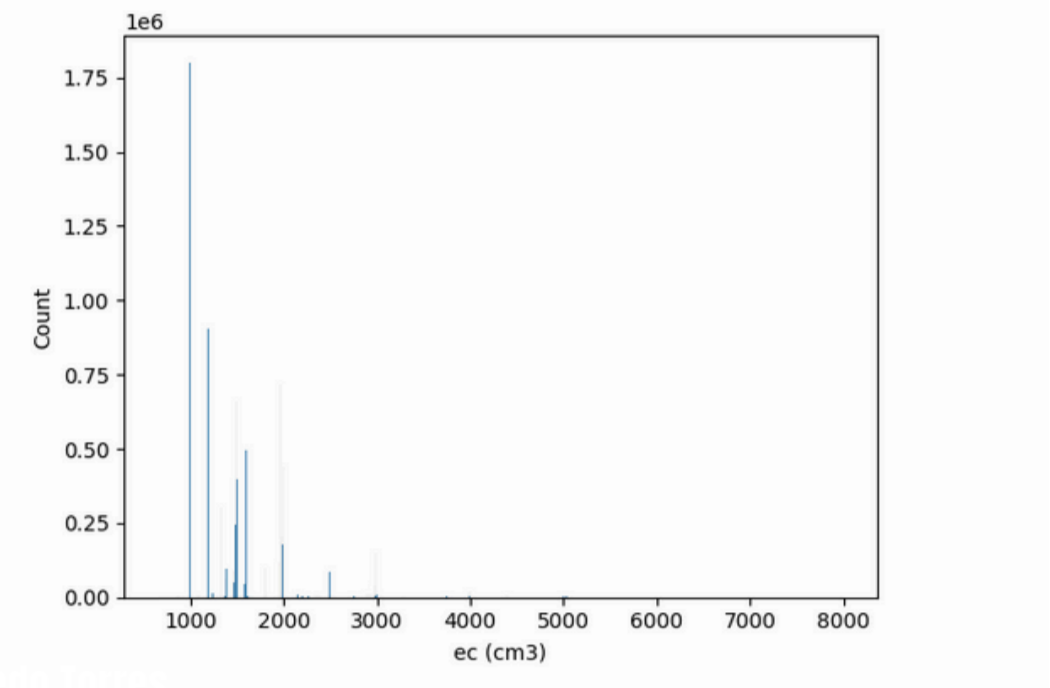




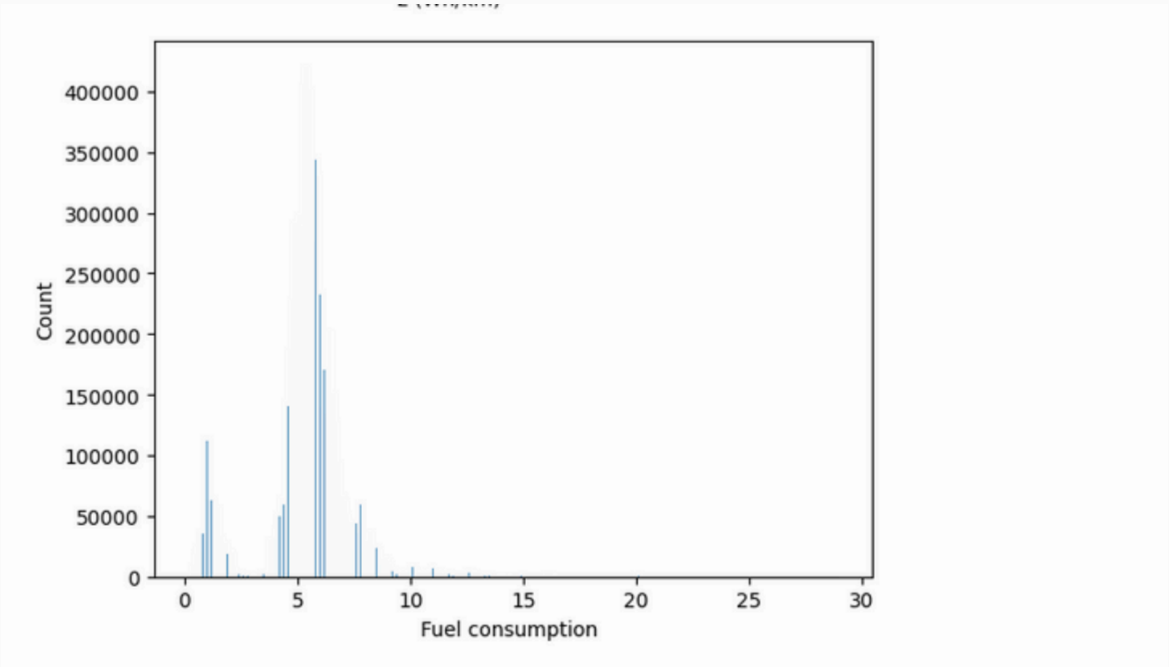
EP



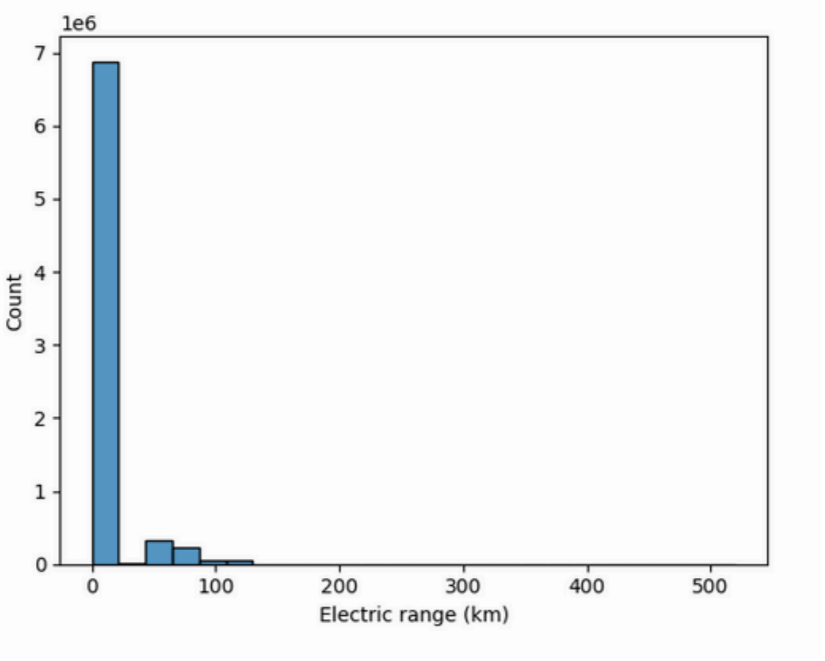
Ec



FUEL CONSUMPTION



ELECTRIC RANGE



MODEL SELECTION

We have imported neural network components from TensorFlow and other machine learning techniques from Sklearn. The models that we have used are as follow:

Linear Regression:

Linear regression is a simple model that assumes a linear relationship between the input features and the target variable.

Decision Tree Regressor:

It is a tree-based model that splits the data into subsets based on feature values.

Random Forest:

It is an ensemble model that combines multiple decision trees to improve performance and reduce overfitting.

Gradient Boosting Regressor:

It is another ensemble model that builds trees sequentially, each correcting the errors of the previous ones.

Support Vector Machine:

It uses support vector machines to find the best hyperplane that predicts the target variable.

Neural Network:

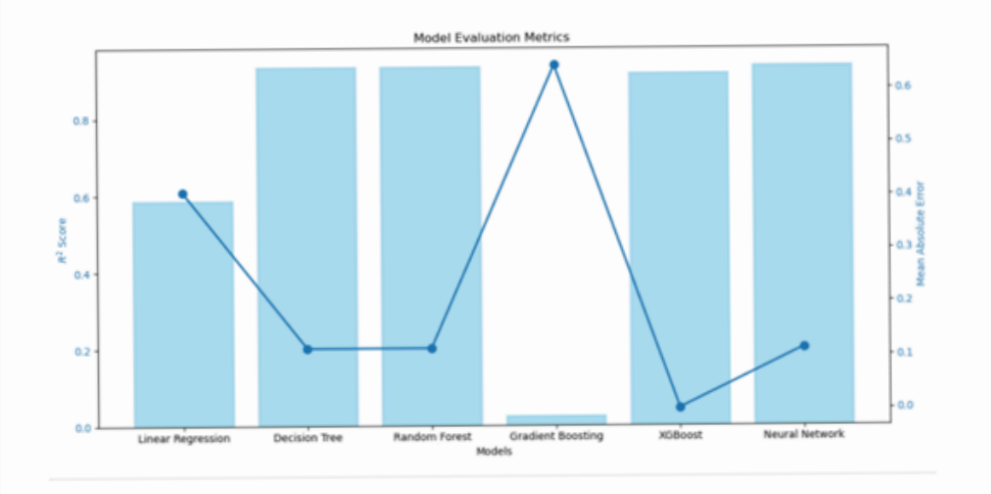
A sequential neural network model with an input shape that matches x_train is defined. It has six hidden layers with different unit sizes (128, 256, 64, 128, 32, and 64 units) and linear activation functions. After every dense layer, there are two layers: one for regularization (dropout rate of 0.2) and another for normalization (Batch Normalization). Regression tasks are suitable for the single linearly activated unit in the output layer. This architecture uses regularization techniques to prevent overfitting and learn complex patterns from the data.

MODEL TRAINING AND EVALUATION

We have trained the model within a scikit-learn Pipeline. It includes preprocessing steps defined by preprocess. The regressors are configured with parameters such as low learning rate, estimators, and settings for tree complexity in some models like Gradient Boost and Random Forest (max depth, minimum samples, and minimum samples per leaf). Predictions are made on x_test after the model has been fitted to x_train and y_train. On the y_test, performance metrics are calculated to evaluate the prediction accuracy and generalization of the model, including R^2 score, mean squared error, explained variance score, and mean absolute error.

RESULTS

We have taken evaluation metrics for each model to decide which is the best and make a comparative graph. There are several models in the notebook, and evaluations of metrics like Mean Absolute Error (MAE) and R-squared (R^2) are included for each model. With the second-lowest MAE scores at 0.1131 and the greatest R2 score of 0.9343, the Neural Network Model is the best model. It indicates that it provides an ideal balance of low error and high predicted accuracy. These metrics are shown graphically in the bellow graph, which makes it easier to compare all models by displaying MAE as dots and R2 values as bars.



TEAM ZETA

MACHINE LEARNING

TEAMZETA@ITSOLERA.COM
E-mail Address

+123-456-7890
Phone Number

123 Anywhere St., Any City, ST 12345
Official Address