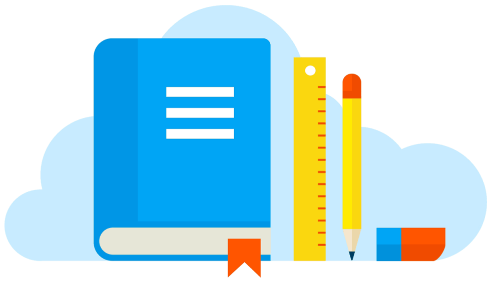


# Machine Learning challenge

# Regression problem

Here is the description of our steps in the group project:

## 1. Data loading and Early Data Inspection

Load: To check the effect of supplemental data on the results, the training data were loaded once without it and once aggregated with it and then loaded.

Basic information: In Both phases, we checked the basic information of datasets by de.head(), df.shape, df.info(), df.Descriptions()

Domain Knowledge: Gain insights about the data and its context by understanding the domain.

## 2. EDA

Considering the importance of EDA, in any data analysis, we attempted to get some hints by analyzing:

● Distributions of numeric features (Histogram and Box-Plot)

● Distributions of target (Histogram and Box-Plot)

● Checking the correlation between numerical features and Target (label) by correlation plot

● Checking the unique members in categorical features by df ['featuere'].unique()

● Checking the noises in some features such as minus inputs, sudden spaces and so on.

## 3. Representations & Feature Engineering

### 3- 1 Extracting features

● Timeutc: After analysis of this feature and because of having some seasonal effect or change in emotional well-being based on the time of the day, we extracted year/month/day/hour

● Creating two new features based on ‘ResponseValue‘ ( the Label) and UserID. Then we joined it to Test_data and imputed the other values for the users that aren’t in common with Train_data.

### 3-2 Transforming features:

We used the following methods to improve the performance of a machine-learning model.

● Label Encoding: for ‘QuestionTiming’ because it has 2 members. Also, for Time Utc extracted feature

● One-Hot encoding: for ‘CurrentGameMode’, ‘CurrentTask’ and ‘LastTask’

● Standardization: for ‘CurrentSessionLength’

## 4. Train-test splitting

We used two approaches to splitting our training data to train and validation set

● Hold-out approach: We selected 20% of Train_data as the Validation set.

● K-fold approach: We tried to implement it multiple times, but ran into memory/capacity errors.

## 5. Training & Evaluation

● Learning algorithm(s): here is a list of algorithms used in our assignment:

- ☐ Dummy Regressor
- ☐ Linear regression
- ☐ DecisionTreeRegressor
- ☐ RandomForestRegressor
- ☐ Neural Network
- ☐ GradientBoostingRegressor
- ☐ MLPRegressor Pipeline (2 polynomial)
- ☐ KNeighborsRegressor
- ☐ SVR
- ☐ SVC
- ☐ Lasso
- ☐ ElasticNet Ridge

● Hyperparameter tuning: We tried to implement it multiple times, but ran into memory/capacity errors.

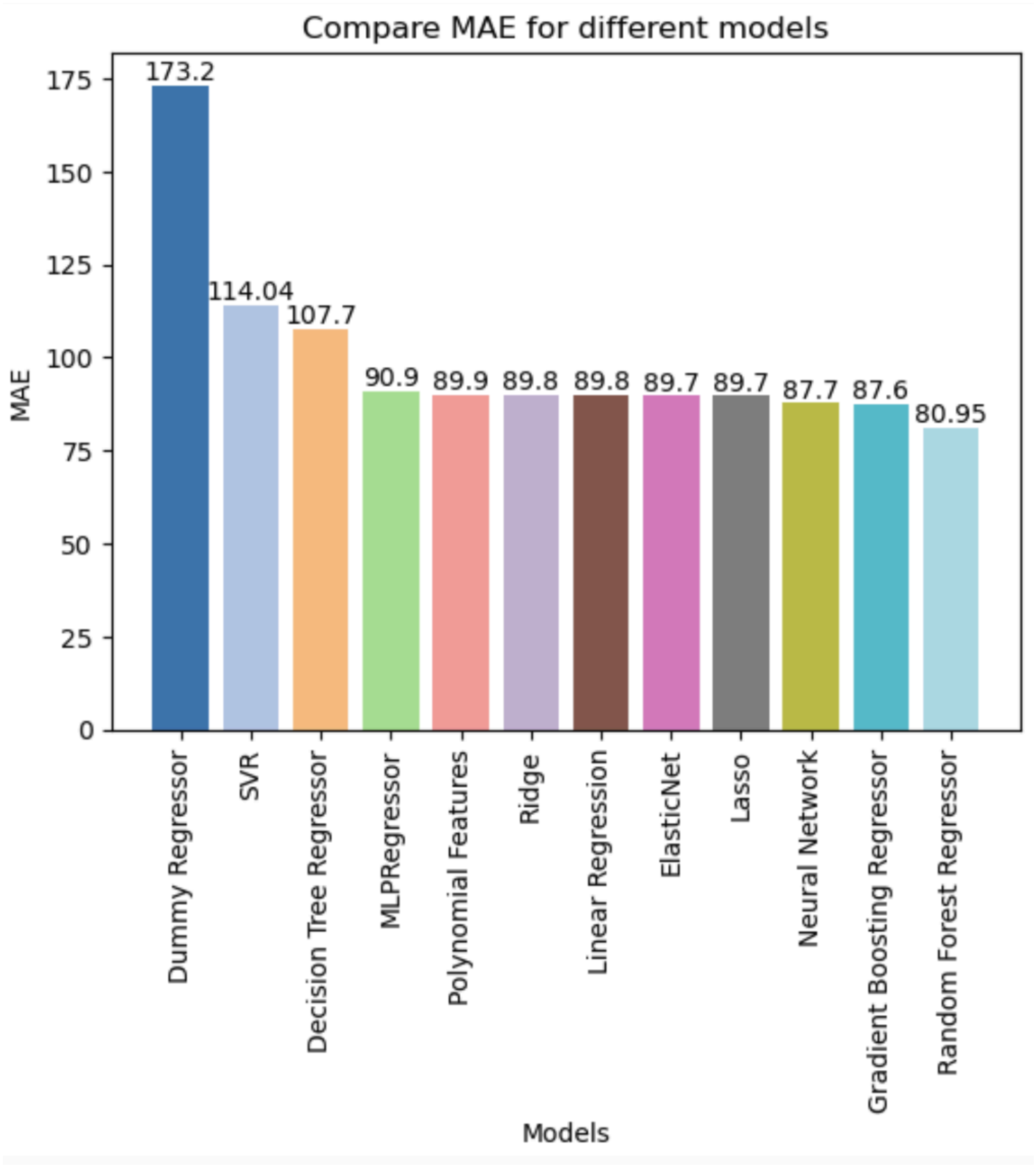
● Evaluation: We used MAE as the assignment specified it.

● Selecting the best model: The best model (with an MAE of 111) was achieved using the RandomForestRegressor

## 6. Discussion of the performance of your solution

Here we summarize our results and generalization of the Model. Comparing our last output of MAE=111 (as the only assessment metric permitted in the assessment) with the dummy regressor result which is considered as the baseline,(172), and also comparing it with the best result in the class (MAE=100) showed that the model had adequate generalization to the test set.

Comparing the Random Forest’s output with other models' output demonstrates its wonderful capacity to generalize the model. The Gradient Boosting Regressor and Neural Network showed comparable performance, but because they have less interpretability, a reason for higher results may be their difficulty to read and understand how decisions are made.


Compare MAE for different models
MAE
Models
Dummy Regressor
SVR
Decision Tree Regressor
MLPRegressor
Polynomial Features
Ridge
Linear Regression
ElasticNet
Lasso
Neural Network
Gradient Boosting Regressor
Random Forest Regressor