**Auto MPG**

**Source:**

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition.

**Data Set Information:**

This dataset is a slightly modified version of the dataset provided in the StatLib library. In line with the use by Ross Quinlan (1993) in predicting the attribute "mpg", 8 of the original instances were removed because they had unknown values for the "mpg" attribute. The original dataset is available in the file "auto-mpg.data-original".

"The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes." (Quinlan, 1993)

**Attribute Information:**

1. mpg: continuous
2. cylinders: multi-valued discrete
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: multi-valued discrete
8. origin: multi-valued discrete
9. car name: string (unique for each instance)

**Problem Definition**

The dataset contains features such as mpg, cylinders, displacement, weight, acceleration, model year etc. We have to build repressor model to predict mpg of car.

**Data Analysis**

We can see that the dataset has the following columns:

* **mpg**: it is the result column which is continuous
* **cylinders**: multi-valued discrete
* **displacement**: continuous
* **horsepower**: continuous
* **weight**: continuous
* **acceleration**: continuous
* **model year**: multi-valued discrete
* **origin**: multi-valued discrete
* **car name**: string

The Auto MPG data set is the collection of 398 records and 9 attributes are present where one columns is the result column. In the dataset, input variables are a blend of numerical and categorical types, where the non-numerical columns are represented using strings. The car name attribute contains the string data type. Also the attribute named as horsepower is having data type as object and contains the float values, which need to be convert in float data type. In this data set we have to predict the mpg which is having data type as float. By checking the data set with the null function we will see none of the attribute will have a null values present in it.

Unique value: The data set contains the less unique values in attribute cylinder and origin. Cylinder is having unique value count of 5, and origin having count of unique value is 3. Rather than this two attributes other columns are having unique value count more than 10. Most of the unique values are present in weight and car name. In this data set we are having 305. Most of the car brand is same but due to the car model it is having more count.

If we print the unique values for each attribute one by one we will see that ‘?’ character are present in the data set, in horsepower attribute with the value count of 6, we need to remove those values in data processing. Let’s move to EDA.

**EDA**

As we have seen horsepower having some missing values with character ‘?’, to process further we need to remove those characters from dataset, to remove ‘?’ from dataset we have to ways to do either we can fill it with the mean or we can remove those records from our data set. As the count of ‘?’ is 6 which is less, so we will remove those values from data set. After removing, character from horsepower we will convert it into float as it is in object by using astype('float').

As we can see there are many unique values are present in car name and we cannot encode this much of unique value which will result us into poor model performance so to overcome this we will use some feature engineering techniques to reduce car names.

After looking into car name column we have noticed it has the car brands in first words of our and model name as second letter of car name column, and can we extract this out for searching relations between them. Also car name column have some typos for brand name. First we will remove the brand name from the car name using extract function. After separating brand name and model name we will replace typo with correct name. After doing this the car name attribute is ready for encoding.

After doing feature engineering on data set, we will check for the correlation of matrix of data set using the heatmap. There we will observe that:

* Mpg attribute is having good correlation with cylinder, displacement and weight.
* Cylinder, displacement and weight are having negative correlation.

Further we will check for the data distribution using the pair plot.

* From the pair plot we will see that displacement, horsepower and weight have strong negative relationship.
* Also acceleration is not much significant in predicting mpg.

Further we will check for description of data set, from the description we will see that there are huge difference between min and max of horsepower and weight.

It shows that their might be some outliers are present or data might be noisy. To check whether data is noisy or having outliers we will plot a graph using boxplot. The boxplot will show that the data distribution with min and max also the data present outside the box. In our case, there are very few element present which is ignorable and it seems to be normal data set.

**Pre-processing Pipeline**

Now we will split the data to x and y so we can mode to next step of preparing model for that we have to make our data normal distributed. To distribute data into x and y we will initialize two variables and will assign the attributes to x and y respectively. As mpg is the attribute that has to predict we will pass mpg into y variable, and other attributes will assign to x. Before moving further we have to check for the shape of x and y to make sure we have select the attributes that we need.

As x is the attributed which are independent it has to be scaled and normal distributed data. To achieve this we will check for the skewness of independent data i.e. for x.

Skewness: To check the skewness of the data we will use the skewness function, skew() is the function which is use to see the skewness of the data. By checking the skewness we will find that all the features are right skewed. An attribute acceleration is looks like it is normally distributed. To process the model further, we need to remove skewness from data set, here we will use sqrt function, and will remove the skewness only from the attributes where skewness value is greater than 0.5, because the normal range for skewness is from 0.5 to -0.5, if data set exceeds this range then it will be consider as skewness. After implement sqrt(x) we will see that the data will be normally distributed.

**Building Machine Learning Models**

Now that we know what our data looks like, let’s use some machine learning models to predict the value of MPG given the values of the factors. We will use pythons scikit learn to train test and tune various regression models on our data and compare the results. We will build our model by testing multiple regression model using train test split and by using cross validation score. As it is a regression problem we will calculate r2 score. The models will be calculate on r2 score. We will calculate r2 score for each model individually and will store it into a list.

To prepare the model and calculate the r2 score we will use a for loop to test our model for each random state, to do this we will create a function name as best\_model() where we will initialize max\_r2\_score variable with initial value as 0, this variable will contain the max r2 score that will get from the r2\_score function in the loop. Then we will run a ‘for’ loop with the random state range from 42 to 101.

Inside the for loop we will divide our data into train and test using train\_test\_split, where we will pass data size of 0.2 and random state values will be coming from for loop and will store train and test values in train\_x, train\_y, test\_x, test\_y. After this we will calculate r2 score for each model and will compare each r2 score with the previous r2 score until each random state get execute, also we will calculate mean squared error, mean absolute error and root mean squared error, then we will return max r2 score. As it is regression model, here we will run all regression model one by one and will also plot distribution of residual and will also plot scatterplot with test and predicted data. Distribution of residual will be used to calculate the difference between the observed values of the target variable i.e. y and the predicted value pred\_y. i.e. the error of the prediction.

We will also run our model with cross value score by passing each model instance or object to cross value score function. Here also we will create a common function for cross value which will accept one parameter i.e. model instance, by using cross value score we will calculate it for r2 score with cross fold of 10. In the cross value score function we look for mean r2 score and standard deviation.

We will use the following regression models:-

* Linear Regression
* Gradient Boosting Regression
* Decision Tree Regression
* Random Forest Regression
* KNR Regression

Linear Regression: By passing the linear regression object to our best\_model function to calculate r2 score we get 88 as r2 score from random state 50, and the distribution of residual looks normal.

Gradient Boosting Regression: By passing the gradient boosting regression object to our best\_model function to calculate r2 score we get 94 as r2 score from random state 44, and the distribution of residual looks normally distributed.

Decision Tree Regression: By passing the gradient boosting regression object to our best\_model function to calculate r2 score we get 88 as r2 score from random state 84, and the distribution of residual looks normally distributed.

KNR: To implement KNR we will use GridSearchCV to find the best parameters for KNR, in GridSearchcCV we will pass "n\_neighbors":range(1,30), and cross fold of 10, by running the Grid SearchCV KNR get the paramters of n\_neighbors = 5. Now to run the KNR we will pass n\_neighbors = 5 as the parameter of KNR. By passing the KNR object to our best\_model function to calculate r2 score we get 91 as r2 score from random state 42, and the distribution of residual looks normally distributed.

Random Forest Regression: To implement RFR we will use GridSearchCV to find the best parameters for RFR, in GridSearchcCV we will pass " n\_estimators":[50,100,200], and cross fold of 10, by running the Grid SearchCV RFR get the paramters of n\_ estimators = 50. Now to run the RFR we will pass n\_estimators = 50 as the parameter of RFR. By passing the RFR object to our best\_model function to calculate r2 score we get 91 as r2 score from random state 42, and the distribution of residual looks normally distributed.

By running all regression module we got the best r2 score Gradient Boosting Regression.

After running the train test split now we will check same for cross val score by passing each model instance to cross val function one by one with the same parameters that we got by using Grid search. After passing each model through cross val and getting r2 score we will compare each r2 score. We will visualize the r2 score in barplot by preparing a list of r2 scores along with their model name.

After getting max r2 score for Gradient boosting we will also check with Lasso and Ridge score. To do so Lasso and Ridge need to be import from sklearn.linear\_model by checking Lasso and Ridge score.

Now, we can consider Gradient Boosting Regression as our final model as it has performed better than other model on random state of 44 and gave r2 score of 94.

**Saving a model**

To save our model we will use our best accuracy score model Gradient Boosting with random state 44. By using train test split we will calculate r2 score, mean squared error and mean absolute error. After this we will store our model for future use. We will save our model in pickle by importing joblib from sklearn.externals.

**Conclusion**

In this project, we've overcome missing values and applied a data transformation technique which is Label Encoder to convert categorical data into numerical data, also used scaling technique to scale our data. The model is facing under fitting also. But to make new prediction of MPG based on attributes our model in very much accurate and gave the good prediction that we can say by r2 score.