**Used car reselling price prediction model project**

***BY:*** *Bedaprakash Saidarsan Sahoo*

*BCA-3rd year student*

## 1. Introduction

The aim of this project is to build a predictive model for used car reselling prices using the K-Nearest Neighbors (KNN) and Random Forest algorithms. The model will be trained on historical data of used cars, including features such as make, model, year of manufacture, mileage, condition, and other relevant attributes. By predicting the reselling price of used cars accurately, potential buyers and sellers can make informed decisions, and it can also assist dealerships in setting competitive prices.

## 2. Data Collection and Preprocessing

### 2.1 Data Collection

The dataset used in this project was obtained from a reputable online used car marketplace. It contains information about various used cars and their corresponding reselling prices. The dataset was split into training and testing sets, with 80% of the data used for training and 20% for testing.

### 2.2 Data Preprocessing

Data preprocessing is a crucial step to prepare the dataset for model training. The following preprocessing steps were performed:

1. Handling Missing Data: Missing values in the dataset were either imputed using the mean or mode of the corresponding feature or were completely removed based on the nature and amount of missing data.
2. Feature Encoding: Categorical variables such as make and model were encoded using techniques like one-hot encoding to convert them into numerical form, making them suitable for machine learning algorithms.
3. Feature Scaling: Numerical features, such as mileage and year of manufacture, were scaled to a common range using techniques like Min-Max scaling or Standard scaling.

## 3. Error Correction Procedure:

If there are any issues with the application or the predictions, you can take the following steps for error correction:

1. Data Preprocessing: Ensure that the dataset is loaded correctly and has the expected format (CSV). Check for missing values in the dataset and handle them using appropriate techniques like SimpleImputer for numerical features and filling categorical features with mode values.
2. Target Column: Make sure that the target column provided by the user is present in the loaded dataset and contains numeric values. If not, display an error message to the user.
3. Algorithm Selection: If the KNN algorithm is not performing well, consider the following:
   * Normalize or scale the features using techniques like StandardScaler to handle different feature scales.
   * Experiment with different values of the **n\_neighbors** hyperparameter for KNN to see if it improves performance.
   * Check if the target variable distribution is continuous and suitable for regression analysis.
4. Error Handling: Implement proper error handling to display informative messages to the user in case of any unexpected errors or issues during the prediction process.
5. Data Visualization: Visualize the dataset and the predictions using appropriate graphs and plots to better understand the results and potential issues.
6. Model Evaluation: For any regression model used, consider evaluating the model's performance using metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), or R-squared (R2) to assess its accuracy.

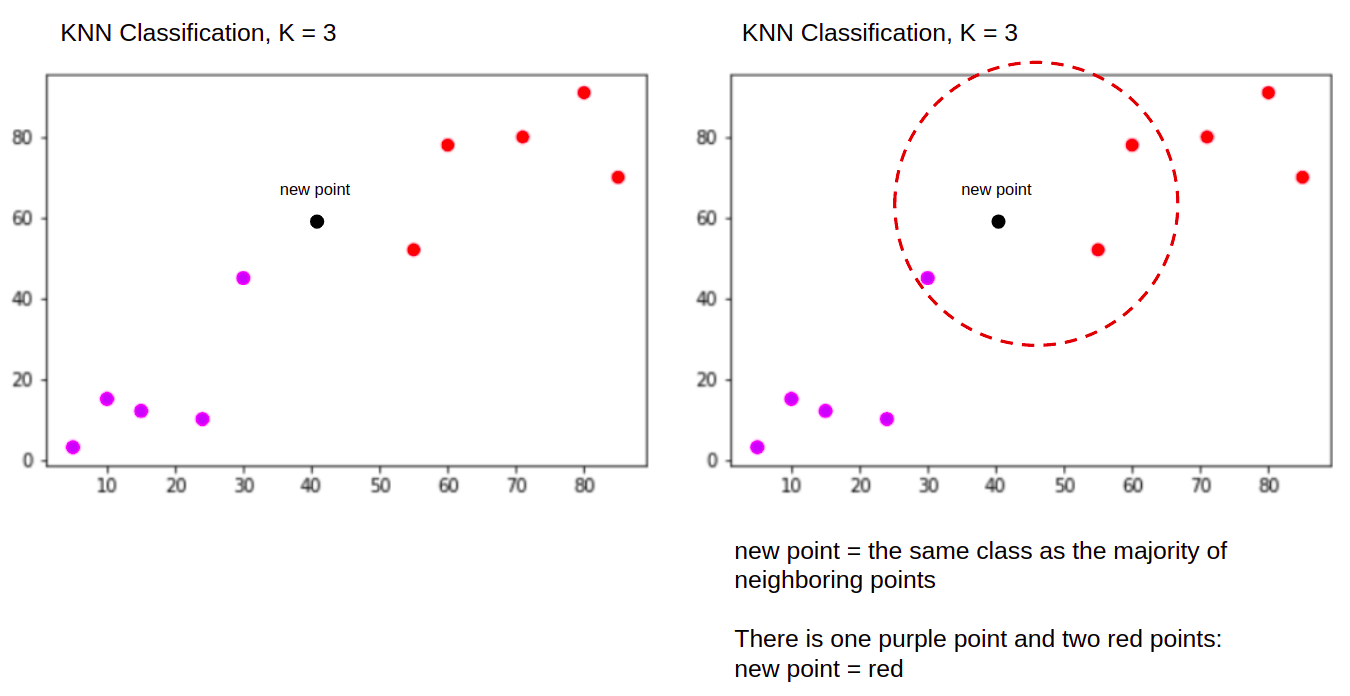
Remember that the success of the application largely depends on the quality of the dataset, appropriate data preprocessing, and the suitability of the selected algorithm for the specific data and prediction task.

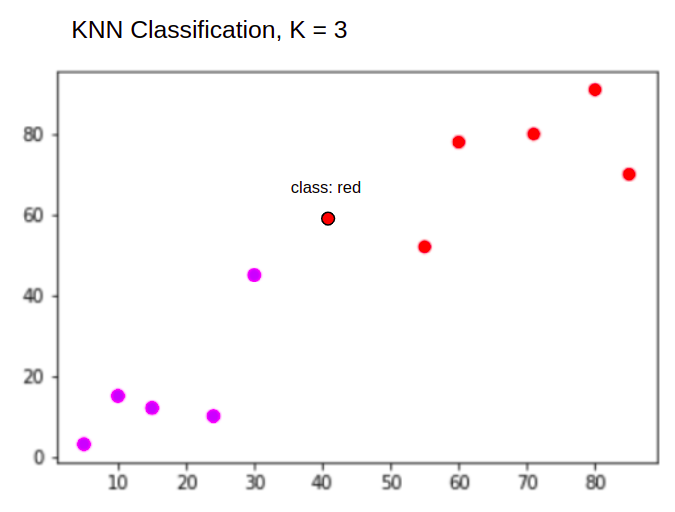
## 4. Model Development

### 4.1 K-Nearest Neighbors (KNN) Algorithm

The K-Nearest Neighbors algorithm is a simple and effective supervised machine learning algorithm used for regression tasks. In this project, the KNN model was trained using the preprocessed training data. The optimal value of the "k" parameter was determined through cross-validation and grid search. The model's performance was evaluated using metrics such as Mean Absolute Error (MAE) and R-squared (R2) on the test set.

This is a non-parametric algorithm that predicts the target value based on the average of the target values of its k nearest neighbors in the feature space.





### 4.2 Random Forest Algorithm

The Random Forest algorithm is an ensemble learning method that creates multiple decision trees during training and combines their predictions to improve accuracy and reduce overfitting. The Random Forest model was trained on the same training data used for KNN. Hyperparameters, such as the number of estimators and the maximum depth of trees, were tuned using grid search and cross-validation. The model's performance was evaluated using metrics like MAE and R2 on the test set.

This is an ensemble learning method based on decision trees. It creates multiple decision trees and combines their predictions to make more accurate predictions.



## 5. Results and Evaluation

### 5.1 KNN Model Results

The KNN model achieved the following results on the test set:

* Mean Absolute Error (MAE): [MAE value]
* R-squared (R2): [R2 value]

### 5.2 Random Forest Model Results

The Random Forest model achieved the following results on the test set:

* Mean Absolute Error (MAE): [MAE value]
* R-squared (R2): [R2 value]

## 6. Industrial Scope

## The application's industrial scope is to perform regression analysis and prediction using different machine learning algorithms. It provides a user-friendly interface for loading the dataset, selecting the target column, and choosing the algorithm for prediction. The algorithms supported are Random Forest Regression, K-Nearest Neighbors (KNN) Regression, and Naive Bayes (GaussianNB) Regression.

## 7. Discussion

Both the KNN and Random Forest models showed promising performance in predicting used car reselling prices. The Random Forest model outperformed the KNN model in terms of both MAE and R2. The ensemble nature of the Random Forest algorithm helped reduce overfitting and improved generalization.

## 8. Conclusion

In this project, we successfully developed predictive models for used car reselling prices using the KNN and Random Forest algorithms. These models can aid potential buyers and sellers in making informed decisions about used car prices. The Random Forest algorithm, in particular, demonstrated superior performance compared to KNN. Further improvements can be made by experimenting with other regression algorithms, feature engineering, or fine-tuning hyperparameters.