**Project Description Document : *Knn Regressor,* , *Linear Regressor***

**Dataset Information:**

***Name:* California House price**

***Type:* Numerical Dataset**

**Dataset split:**

***-Training set:* 18000 row**

***-Testing set:*  1217 row**

***-Validation set:*  1216 row**

**Data Preprocessing:**

***Cleaning Data:***  Used get dummies on ocean proximity column because it contains string values so it converts string value into columns each column contains value (0, 1)

***Converting into metrices:*** Convert each of: X\_train, y\_train, X\_test, y\_test, X\_val, y\_val into metrices matrix

***Standard Scaler:*** by making preprocessor function it’s parameter is x metrices of matrix to make values of each feature column(from column 0 to column 7) scale the data by normalization and transform it under the bell curve(0~1)

**Another Implementation details:**

***Used Mean Squared Error and R-squared:*** to calculate training and validation error using linear regression and the result was so bad and caused the overfitting problem

But by using Knn the validation error reduced

***Choos best hyperparameter***:

It initializes a K-Nearest Neighbors regressor (knn) and defines a parameter grid for hyperparameter tuning and choose best hyperparameter and the way he will apply(Euclidean, Manhattan )

By using GridSearchCV the model is trained on the training data with k-fold cross-validation.

The training and validation errors are calculated using the mean squared error (MSE).

The R-squared score is computed for the test set, providing a measure of model performance.

**Results details:**

***Validation error:*** 60712

***Training error:*** 51965

***R-squared knn:*** 76%

***R-squared Linear:*** 69%

**Project Description Document : *K-Mean***

***Dataset Name :*** Flower Species Recognition

***Number of Clusters:2***

***Total Number of Samples:8189***

***Sample Size (in case of images):***1 -> [8189]

***The silhouette score :***

start the clusters from 2 to 102 to find best k clusters

***Implementation Details***

***Feature Extraction:***

***- Number of Features Extracted: [***2]

***- Feature Names:*** [Pixles and rgb colors]

***- Dimension of Resulted Features:*** [dimensions reduced to 32,32 and then normalized from 0 to1]

***Hyperparameters:***

used k clusters from 2 to 102 to find best sillhoute score

- Batch Size: [Specify the batch size](Not used as in kmeans minibatches changes result)

***Results Details***

***Model Evaluation on Testing Data:***

- sillhoute Curve: []

- sillhoute Accuracy: [0.15]

***Additional Notes***

used cv2 and sklearn clusters , PCA (Principal Component Analysis)

**Project Description Document : *Logistics Regressor***

***Dataset Information:***

***Name:*** Flower Species Recognition

***Number of Classes:*** 102

***Total Number of Images:*** 8189

***Selected Classes:*** I focused on a subset of 5 classes as required. The labels and the number of images for each flower species are as follows:

***Grape hyacinth:*** 41 images

***Hard-leaved pocket orchid:*** 60 images

***Spear thistle:*** 48 images

***English marigold:*** 65 images

***Canterbury bells:*** 40 images

***Dataset Split:***

***Training Set:*** 203 images

***Testing Set:*** 51 images

***Data Preprocessing:***

***Feature Extraction:*** Utilized Histogram of Oriented Gradients (HOG) features for image representation., features don't have human-readable names they represent information about the gradients and orientations of edges in the image.

***Feature Dimensionality:*** HOG features are represented as a 1D array,

Dimension of HOG features: (171396,)

***Variable Feature Count:*** The number of features extracted varies for each image, ranging from 291600 to 403380, indicating non-constant feature counts.

***Model Configuration:***

***Regularization Parameter (C):*** Set to 10.0.

***Maximum Iterations (max\_iter):*** Configured to 10,000 for solver convergence.

***Multi-class handling:*** ovr (one vs rest)

***Solver:*** Set to 'liblinear'. The solver parameter determines the algorithm used for optimization during training.

***regularization type(penalty):*** L1(Lasso)

***Model Type:***

***Problem Type:*** Multi-class classification.

***Evaluation Metrics:***

***Roc-Auc Graph***: Multiclass Roc curve (one vs rest).

***Performance Metrics:*** Confusion matrix, accuracy, and classification report

***Accuracy:*** Achieved an accuracy of 75%.