|  |  |
| --- | --- |
| Name | ID |
| حازم احمد عبدالعال شاذلي | 20210265 |
| حامد وليد فتحي | 20210273 |
| حافظ عادل حافظ | 20210272 |
| عبدالرحمن عبد العزيز ابراهيم عبدالعزيز | 20210514 |
| محمود عبدالدايم ابو المجد حسين | 20210867 |
| جمال الدين ايمن عبدالرحمن | 20210250 |
| عبدالله محمود جمال الدين | 20210558 |

GitHub Repository Link

Numerical Dataset:

[Hazem-Ahmed1/House\_Prices\_Advanced\_Regression\_Techniques (github.com)](https://github.com/Hazem-Ahmed1/House_Prices_Advanced_Regression_Techniques)

Image Dataset:

[gimmeursocks/UTKFaceClassifier (github.com)](https://github.com/gimmeursocks/UTKFaceClassifier)

**Numerical Data Set**

1. **General Information**

**1-Name:** House Prices Advanced Regression Techniques

**2-Total Number of Samples:** 1460

**3-Number of Training:** 80% -> 1168

**4-Number of Testing:** 20% -> 292

**(b) Implementation details**

**Features Extracted Number:** 4

[“HouseAge”,”GarageAge”,”TotalBathRoom”,”OutDoorArea”]

**The dimension of resulted features :**

A screenshot of a computer screen

Description automatically generated

**Is Cross Validation is used?** Yes

**Number of Folds**: 5



**Ratio of Training/Validation:**

**A screen shot of a computer

Description automatically generated**

**Hyperparameters used:**

**Number of Neighbors (n\_neighbors):** Defines the number of neighbors used for prediction.

**Weights (weights):** Specifies the weight function used in prediction.

**’P’:** the 'p' hyperparameter is related to the choice of the Minkowski distance metric.

**A black screen with white text

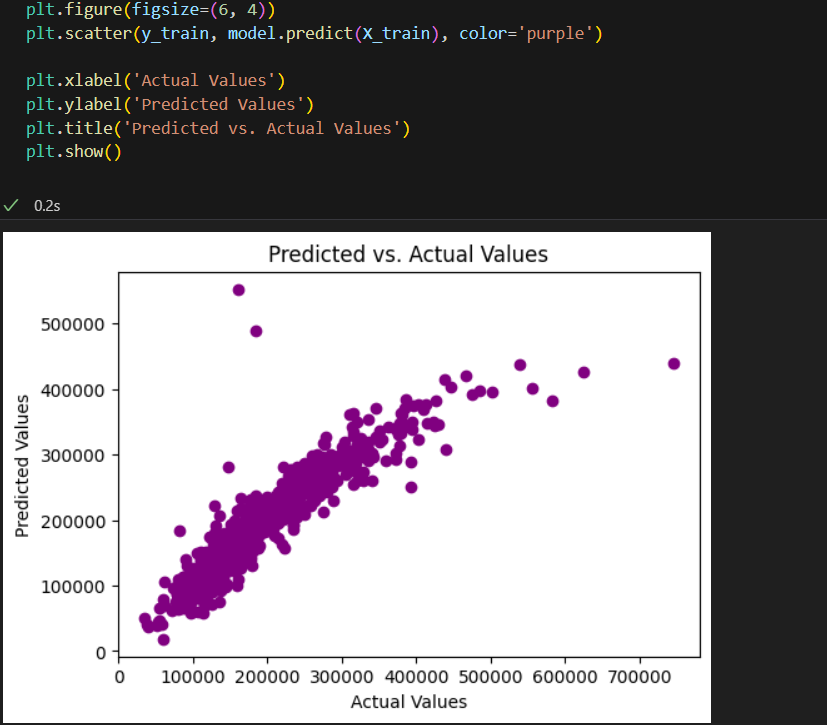
Description automatically generated**

**(c) Result details**

**Accuracy In Linear Regression Model:**

**A screen shot of a computer code

Description automatically generated**

****

**Accuracy In KNN Model:**

**A screen shot of a computer code

Description automatically generated**

**A screen shot of a computer screen

Description automatically generated**

* In terms of overall accuracy, the KNN model performs slightly better (85.62%) than the linear regression model (85.10%).

**Image Data Set**

**(a) General Information**

**Explanation of data :** **A close up of a baby

Description automatically generated**

**Age**

**Gender (0:M,1:F)**

**Race**

**1-Name:** age detection

**2-Total Number of Samples:** 14223 sample   
(dynamic number) original number is 23k

**3-Number of Classes:** 4

-“Kid” 0-17

-”Adult” 18-49

-”old” 50-79

-”senior” 80+

•**Their Labels:** [1,2,3,4]

**4-Number of Training:** 56% -> 7964

**5-Number of validation:** 14% -> 1991

**5-Number of Testing:** 30% -> 4267

**6-Images Size:** 200 x 200 x 3 pixels (Three for RGB channels)

**(b) Implementation details**

**The dimension of resulted features :**

**A black and white text

Description automatically generated**

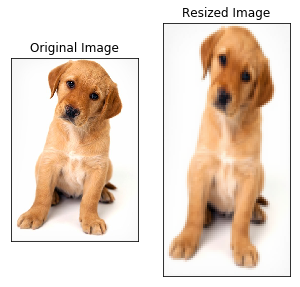
**Features Extracted Using Histogram of Oriented Gradients (HOG):**

HOG is a feature descriptor that captures information about the gradients or edges in an image:

**A computer screen with text

Description automatically generated**

* The function initializes an empty list **hog\_features\_list** to store the HOG features for each image.
* It then iterates through each image path in the input **image\_paths**.
* For each image, it uses **io.imread** to read the image.
* setting **as\_gray=True** to convert the image to grayscale. Grayscale is often used for HOG feature extraction.
* The image is then resized to a fixed size of (64, 64) using **transform.resize**. This step is important because HOG works better with images of a consistent size.



* The **hog** function is applied to the resized image, specifying **orientations=9** to divide the gradient angles into 9 bins. The parameter **feature\_vector=True** indicates that the output should be a 1D array (feature vector) instead of a 2D array.
* The computed HOG features for the image are appended to the **hog\_features\_list**.
* The function repeats this process for all images in the input list.
* Finally, the function returns the list of HOG features for all the input images.

****

**Hyperparameters used in LogisticRegression:**

****

**C (C):** 0.1 ‘Regularization is a technique used to prevent overfitting in a model by penalizing the complexity of the model’

**Solver(‘saga’) :** Stands for Stochastic Average Gradient, which is a variant of gradient descent that uses a stochastic average of past gradients. This solver is often suitable for large datasets.

**Tolerance (tol):** 0.1 ‘refers to the stopping criterion based on the convergence of the optimization algorithm used to find the coefficients that best fit the data

**A graph with blue squares

Description automatically generated**

**Solver(‘saga’) :** Stands for Stochastic Average Gradient, which is a variant of gradient descent that uses a stochastic average of past gradients. This solver is often suitable for large datasets.

**A graph with a line

Description automatically generated**

**Tolerance (tol):** 0.1 ‘refers to the stopping criterion based on the convergence of the optimization algorithm used to find the coefficients that best fit the data

**A graph with a line going up

Description automatically generated**

**C (C):** 0.1 ‘Regularization is a technique used to prevent overfitting in a model by penalizing the complexity of the model’

**(c) LogisticRegression Result details**

**1. Accuracy score in Logistic Regression:**

**A screen shot of a computer program

Description automatically generated**

**2. Accuracy score for each set (train,test,validation) in Logistic Regression:**

**A screenshot of a computer program

Description automatically generated**

**3. Confusion Matrix in Logistic Regression:**

**A blue squares with numbers and a blue square

Description automatically generated**

The elements of the confusion matrix are as follows:

* **Row 1 (Age Group 0):**
  + 572 instances were correctly predicted as Age Group 0.
  + 240 instances from Age Group 0 were misclassified as other groups. And so on

**4. ROC Curve in Logistic Regression:**

**A computer screen shot of text

Description automatically generated**

**A graph of different colored lines

Description automatically generated**

**True Positive Rate (TPR):** The proportion of actual positive instances that are correctly identified by the model.

**False Positive Rate (FPR):** The proportion of actual negative instances that are incorrectly classified as positive.

* **Class 1 (AUC = 0.93):** A high AUC indicates good discrimination for class 1. The closer to 1, the better.
* **Class 2 (AUC = 0.87):** A good AUC but slightly lower than class 1. Still, it indicates reasonable discrimination.
* **Class 3 (AUC = 0.88):** Similar to class 2, indicating good discrimination.
* **Class 4 (AUC = 0.96):** The highest AUC among the classes, suggesting excellent discrimination for class 4.
* Senior(80+) is the easiest, Adult(18-49) is the hardest

**K Means model**

**(b) Implementation details**

**Image resized :**

**A screen shot of a computer code

Description automatically generated**

**The dimension of resulted features :**

**A black rectangular object with a black border

Description automatically generated**

**Hyperparameters used in KMeans Clustering:**

**A graph with a line

Description automatically generated**

**A graph with a blue line

Description automatically generated**

****

**Number of Clusters (n\_clusters):** 3 ‘Determines the number of clusters’

**Maximum Number of Iterations (max\_iter):** 400 ‘The maximum number of iterations the K-Means algorithm will run.’

**Initialization Method (init):** 'k-means++' ‘Specifies the method used to initialize the centroids of the clusters.’

**(c) Result details**

**A graph of a graph

Description automatically generated with medium confidence**Confusion matrices, ROC curves, and loss curves are concepts more commonly associated with supervised learning, particularly in classification tasks. K-means clustering, on the other hand, is an unsupervised learning algorithm used for clustering data.

**Show the frequency of ages in each cluster**