|  |  |
| --- | --- |
|  | Supervised & Unsupervised ML Projects |
|  | under supervision of: |
|  | Dr. Ali Hamza  Eng. Ahmed Nousir  Fall 2023  Made by :   1. Ahmed Nehad Mohamed Shehata 42010704 2. Seif Farid Shawky Mondy Elnemr 42010551 3. Ali Hussien Sayed Ahmed 42010308 4. Abdelrhaman Ayman Ibrahem Elmoamely 42010392 5. Ola Rashad Abo Elmagd Mostafa 42010314 6. Ebada Hamdy Badawy Attiya  42010494 |
|  |  |

|  |
| --- |
| Supervised Learning  Regression |

1. **Definition of the problem:**

- The dataset contains information about cars that is distributed into 13 columns as the following:

Engine: this column represents the car engine capacity value represented in numerical value of float type.

Kilometers\_Driven: this column represents the number of kilometers the car is driven represented in numerical value of float type.

Mileage: this column represents car mileage value as it represents (Total Distance Travelled / Total Fuel Consumed) represented in numerical value of float type.

New\_Price: this column represents the price of a new car represented in numerical value of integer type.

Power: this column represents car engine horsepower value represented in numerical value of float type.

S.No: this column represents the id for each car in dataset represented in numerical value of float type.

Seats: this column represents the number of car seats represented in numerical value of integer type.

Year: this column represents car production year represented in numerical value of float type.

Owner\_Type: this column represents the number of car owners represented in object type.

Fuel\_Type: this column represents car fuel type (petrol / diesel / CNG / LPG / electric) represented in object type.

Transmission: this column represents car gear transmission (Automatic/Manual) represented in object type.

Name: this column represents car engine type represented in object type.

Location: this column represents car location relative to the city represented in object type.

Price: this column represents the price of a new car represented in numerical value of integer type.

- The target is to predict the car price value (Price Column) using this information (Columns).

1. **Methods used:**

We used up to 7 regression models to predict car price (Price column):

1. Linear Regression.

2. Ridge Regression.

3. Lasso Regression.

4. Decision Tree Regression.

5. Random Forest Regression.

6. Support Vector Regression.

7. Gradient Boosting Regression.

1. **Experiment:**

We calculated the score and used multiple ways to calculate error, we will show one of the most important ways to calculate error for some models.

For Linear Regression Model:

is Mean Absolute Error: 11.610493261035199

Explained Variance Score: 0.8991641487487579

------

For Random Forest Model:

Mean Absolute Error: 3.289936266312899

Explained Variance Score: 0.9606662085724429

------

For SVR Model:

Mean Absolute Error: 15.402647383850033

Explained Variance Score: 0.6201433050900348

------

For Decision Tree Model:

Mean Absolute Error: 2.743103527860859

Explained Variance Score: 0.9627801184399765

------

For Gradient Boosting Model:

Mean Absolute Error: 8.097706385954533

Explained Variance Score: 0.937186168356765

1. **References:**

Dataset: https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho?select=car+data.csv

|  |
| --- |
| Supervised Learning Classification |

## Definition of the problem:

- The dataset contains information about Diabetes that is distributed into 9 columns as the following:

Gender: this column represents the biological sex of the individual represented in object type.

Age: this column represents the age of individuals represented in numerical value of float type.

Hypertension: this column represents if the individual suffers from hypertension or not represented in numerical value of integer type.

Heart\_diseases: this column represents if the individual suffers from heart disease or not represented in numerical value of integer type.

smoking\_history: this column represents the smoking status for each individual represented in object type.

bmi: this column represents the measure of body fat based on weight and height. represented in numerical value of float type.

HbA1c\_level: this column represents the measure of a person's average blood sugar level over the past 2-3 months in numerical value of float type.

blood\_glucose\_level: this column represents the amount of glucose in the bloodstream at a given time represented in numerical value of integer type.

diabetes: this column represents the target variable being classified represented in numerical value of integer type.

- The target is to classify if an individual has diabetes or not (diabetes Column) using this information (Columns).

## Methods used:

We used up to 6 regression models to classify if an individual has diabetes or not (diabetes Column):

1. Random Forest.

2. Logistic Regression.

3. Decision Tree.

4. SVM.

5. Gaussian Naive Bayes.

6. KNN.

## Experiment:

We calculated the accuracy for all models which are:

Random Forest Model accuracy: 0.97215

Logistic Regression Model accuracy: 0.94425

Decision Tree Model accuracy: 0.97215

SVM Model accuracy: 0.9567

Gaussian Naive Bayes Model accuracy: 0.90595

KNN Model accuracy: 0.96115

## References:

Dataset: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset

|  |
| --- |
| unSupervised Learning Clustering |

1. **Definition of the problem:**

- The dataset contains information about data scientist job salary that is distributed into 11 columns as the following:

work\_year: this column represents the year the salary was paid represented in numerical value of integer type.

experience\_level: this column represents the experience level in the job during the year represented in object type.

employment\_type: this column represents the type of employment for the role represented in object type.

job\_title: this column represents the role worked in during the year represented in object type.

salary: this column represents the total gross salary amount paid represented in numerical value of integer type.

salary\_currency: this column represents the currency of the salary paid represented in object type.

salary\_in\_usd: this column represents the salary in USD represented in numerical value of integer type.

employee\_residence: this column represents Employee's primary country of residence during the work year represented in object type.

remote\_ratio: this column represents the overall amount of work done remotely represented in numerical value of integer type.

company\_location: this column represents the country of the employer's main office or contracting branch represented in object type.

company\_size: this column represents the average number of people that worked for the company during the year represented in object type.

1. **Methods used:**

We used K-Mean as the main and only clustering algorithm.

1. **Experiment:**

We calculated Silhouette Score the models which is: 0.9757739568499175

1. **References:**

Dataset: https://www.kaggle.com/datasets/ruchi798/data-science-job-salaries/data

|  |
| --- |
| Image Classification |

## Definition of the problem:

- The dataset contains X-Ray chest images that is distributed into 3 classes as the following:

COVID19: this class represents the images that diagnose the individual as covid-19 patient.

PNEUMONIA: this class represents the images that diagnose the individual as pneumonia patient.

NORMAL: this class represents the images that diagnose the individual is not a covid-19 or pneumonia patient.

- The three classes are divided into training images and testing images.

- The target is to classify if the individual is normal, pneumonia patient or covid-19 patient using the available images.

## Methods used:

We used CNN and Pre-trained ViT Model to classify if the individual is normal, pneumonia patient or covid-19 patient as the following:

We used Vision Transformer (ViT) model for being instantiated using the vit.vit\_b16 function. This function likely creates a Vision Transformer model with specific parameters.

The parameters include:

1- the image size: This parameter defines the size of the input images that the model expects. As we set it to be the variable ”IMG\_SIZE” that is predefined by the value (224, 224) so it means the model expects input images with dimensions 224x224 pixels.

2- activation function: This parameter specifies the activation function used in the final output layer of the model. We set it to 'SoftMax', which is commonly used for multi-class classification problems.

3- pretrained: This is a Boolean parameter that determines whether to use pre-trained weights for the ViT model. We set it to `True` so the model will use weights learned on a large dataset during pre-training. This is often useful as the model has already learned useful features from a diverse set of images.

4. include\_top: Another Boolean parameter that decides whether to include the top layers (classification head) in the model. We set it to `False`, as the feature extraction part of the ViT model only will be used, and we will add our classification head.

5. pretrained\_top: This Boolean parameter specifies whether to use pre-trained weights for the top layers (classification head). We set it to `False` as the model won’t use pre-trained weights for the classification head.

6. `classes`: The number of classes in the classification task. This parameter is used to define the number of units in the final Dense layer of the classification head. In this case, we set to `3`, indicating a multi-class classification task with three classes (normal, pneumonia or covid-19).

These hyperparameters allow us to configure the ViT model according to our specific task and dataset. Choosing appropriate values for these parameters is essential for the model to perform well on our problem.

After that we added Keras Input layer. The shape of the input is determined by “IMG\_SIZE” variable that is predefined and has 3 channels.

This line applies the previously created ViT model to the input “inp”. It essentially connects the input to the ViT model, making it part of the overall neural network.

The output from the ViT model is then flattened using the Flatten layer. This is typically done to transform the 3D output into a 1D vector so that it can be connected to densely connected layers.

These lines add fully connected (Dense) layers to the model. Each layer has a specified number of units (256, 64, 32) and is followed by the GELU activation function. GELU stands for Gaussian Error Linear Unit, and it's a non-linear activation function.

The final layer is another Dense layer with 3 units (assuming 3 classes), and it uses the “softmax” activation function. This layer is responsible for the final classification output of the neural network.

In summary, we used a pre-trained Vision Transformer (ViT) as a feature extractor, followed by several densely connected layers for further processing and classification. The network is designed for a multi-class classification task with three classes (normal, pneumonia or covid-19).

## Experiment:

On training:

We calculated the accuracy for the model during training which is: 0.9778%

We calculated the loss for the model during training which is: 0.0660%

ViT model only on testing:

We calculated the accuracy for the ViT model only on testing dataset which is: 89.829%

We calculated the loss for the ViT model only on testing dataset which is: 0.2761%

Main model on testing:

We calculated weighted f1 score for the model on testing dataset which is: 0.902%

We calculated mean squared error for the model on testing dataset which is: 0.38354%

We calculated accuracy score for the model on testing dataset which is: 89.829 %

## References:

Dataset: <https://www.kaggle.com/datasets/alsaniipe/chest-x-ray-image>