Advanced Machine Learning

ID	Name
20210588	عمار سعد محمد
20210411	سلمى ايمن عبد الفتاح
20210417	سلمى محمود فوزى
20210133	ادم محمد عطيه
20210572	على ابراهيم عاصم
20210103	احمد محمد عبد العزيز

كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

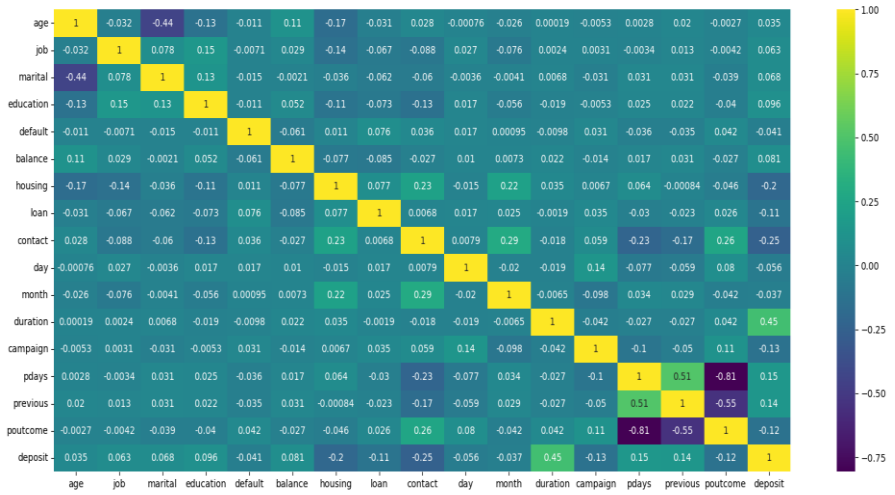
جامعة حلوان
HELWAN UNIVERSITY

classification dataset

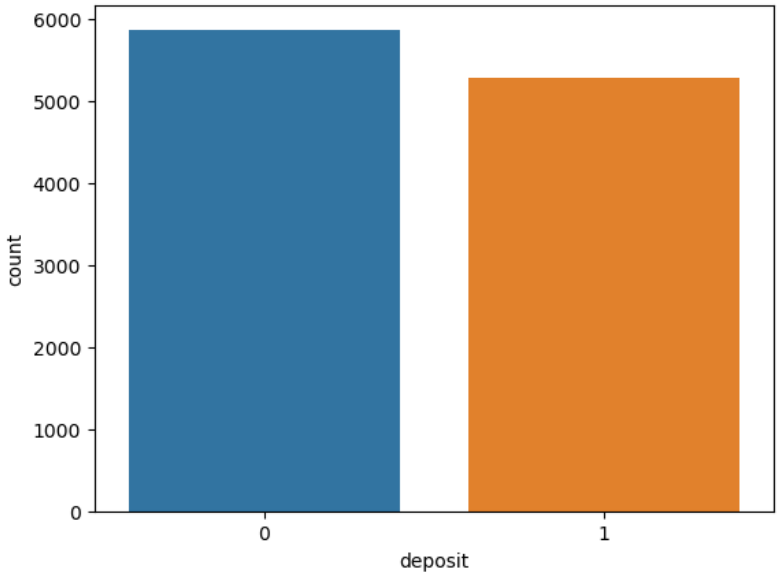
Name :	• **Banking marketing**
Number of Classes :	• 2 [Yes ,No]
Total Size of Dataset :	• 11,162
Training Data Size :	• 7813
Test Data Size :	• 3349
Features :	• 17

Data visualization:

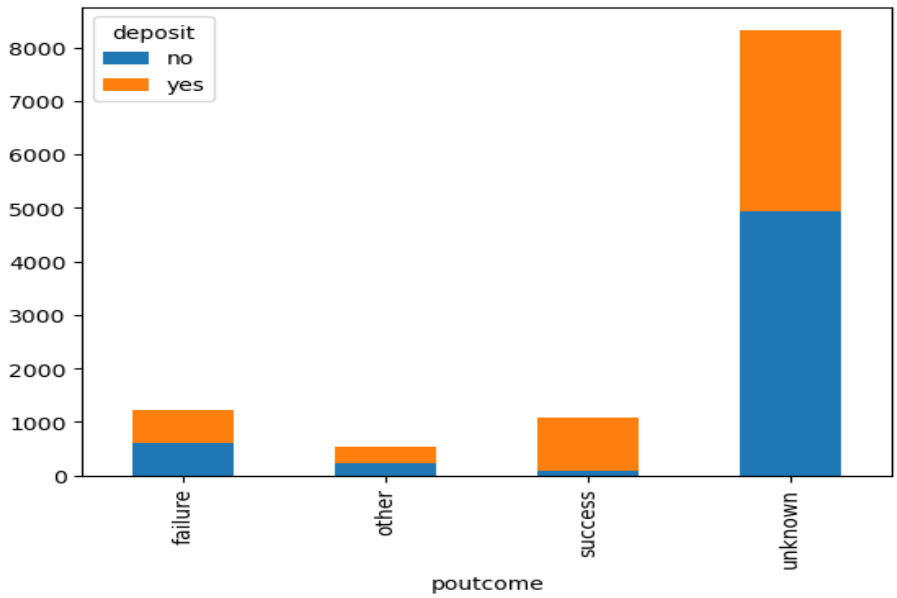
1) Heatmap :



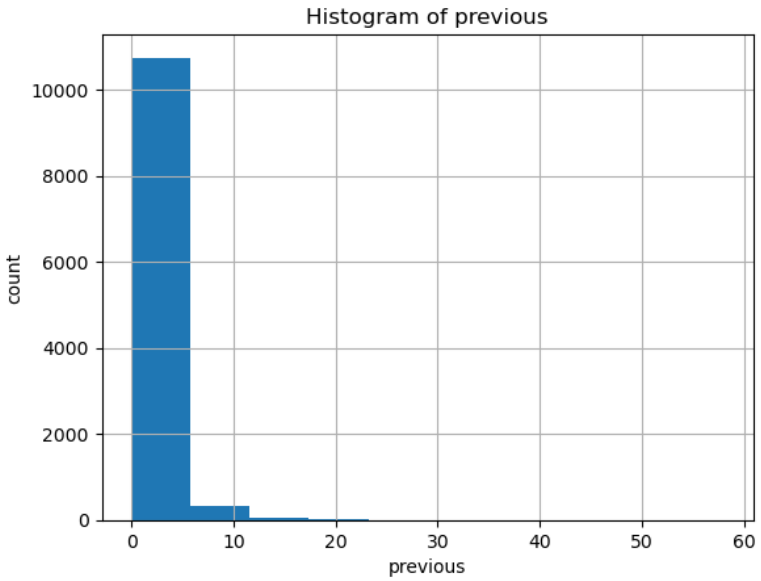
2) Count plot :



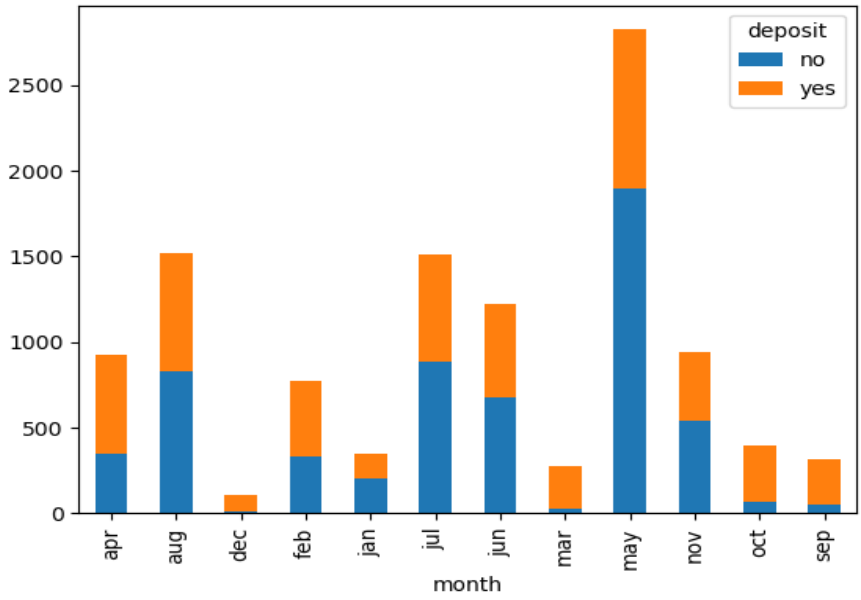
3) Poutcome :



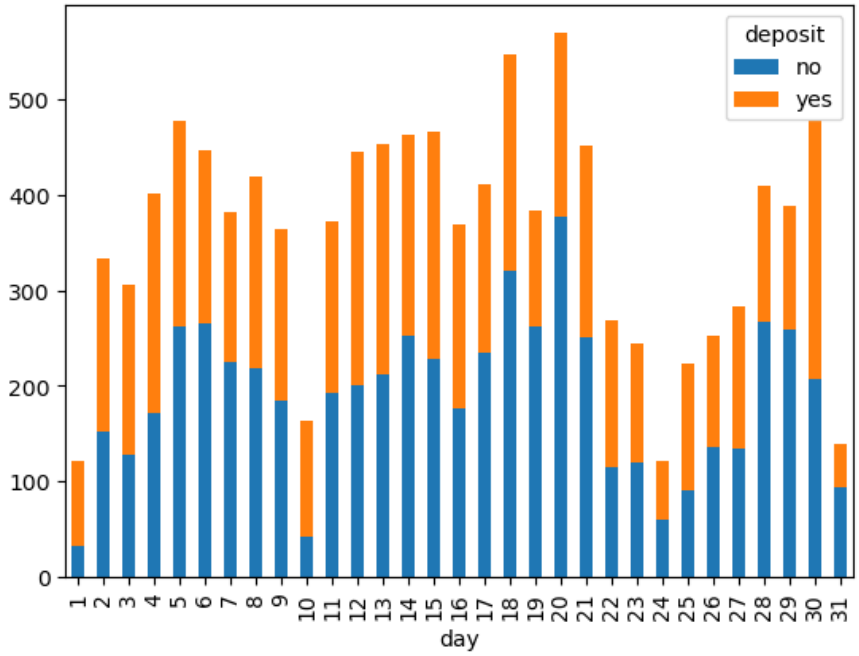
4) Histogram of "previous" :



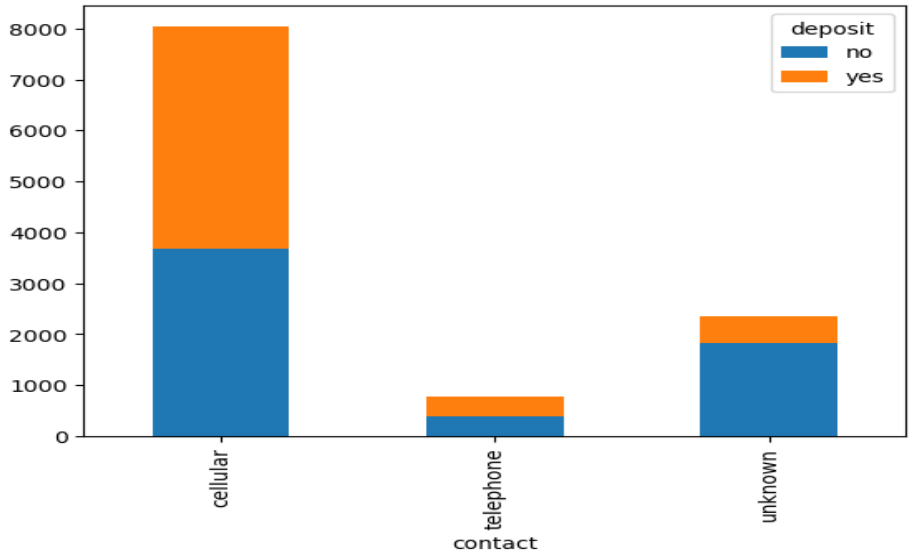
5) Bar chart of categorical variable "month" :



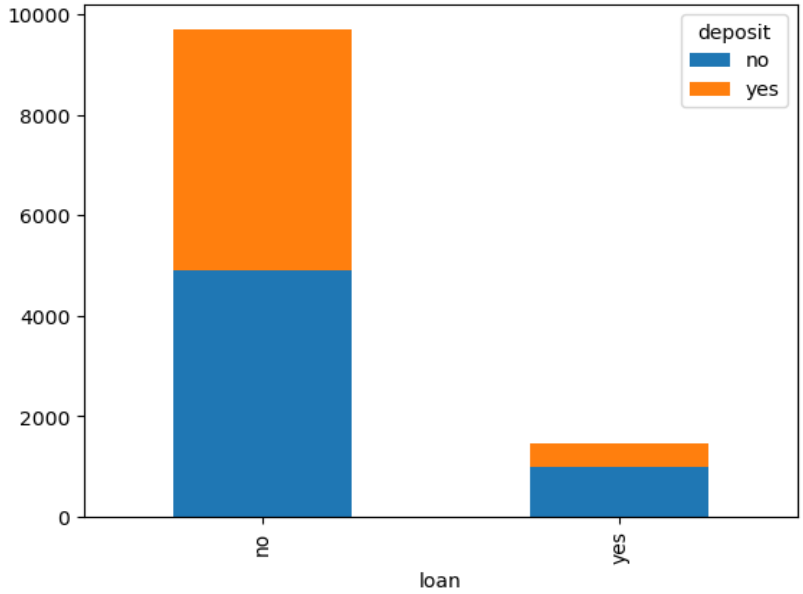
6) Bar chart of categorical variable ("day") :



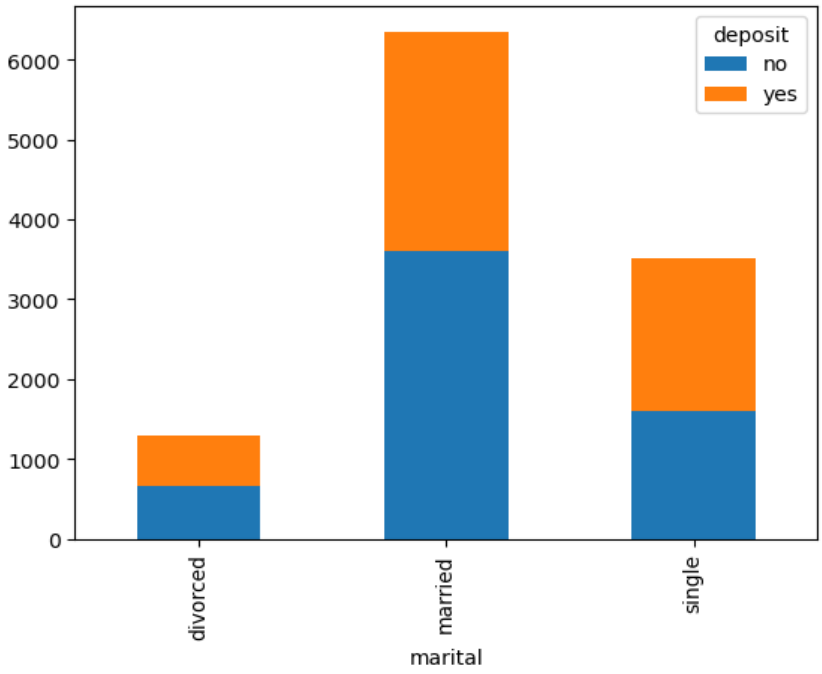
7) Bar chart of categorical variable ("contact") :



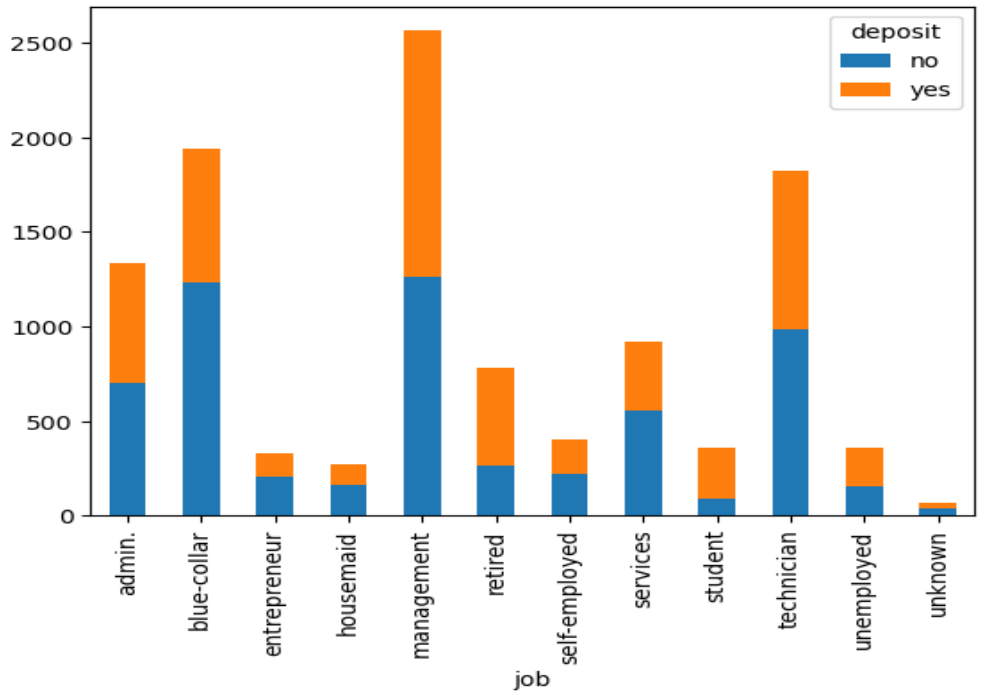
8) Bar chart of categorical variable ("loan") :



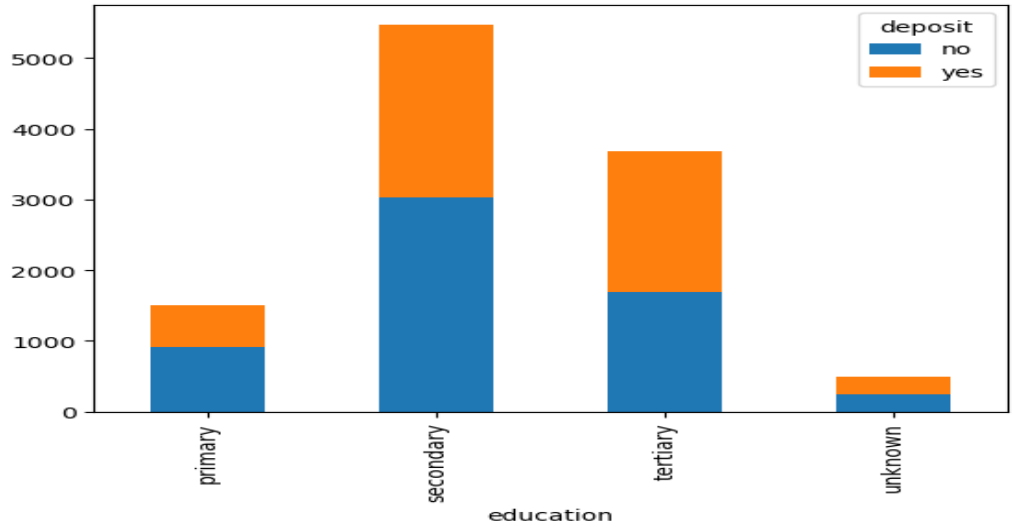
9) Bar chart of categorical variable ("marital") :



10) Bar chart of categorical variable ("job") :



11) Bar chart of categorical variable ("education") :



Decision Tree Model:

1) Implementation Details :

1) Model Hyperparameters:

Criterion	'gini'
max_depth	10
max_features	None
min_samples_split	30

2) Feature Extraction:

```
X = df.drop(columns=['deposit'])  # Features ['age','job','marital','education','default','balance',housing','loan','contact',
                                  #           'day','month','duration','campaign','pdays','previous','poutcome']

y = df['deposit']  #Target
```

The dimension of the resulted features is mentioned in the output: 16 features were extracted.

3) Regularization:

minimum samples per leaf and minimum samples per split, which are used to prevent overfitting.

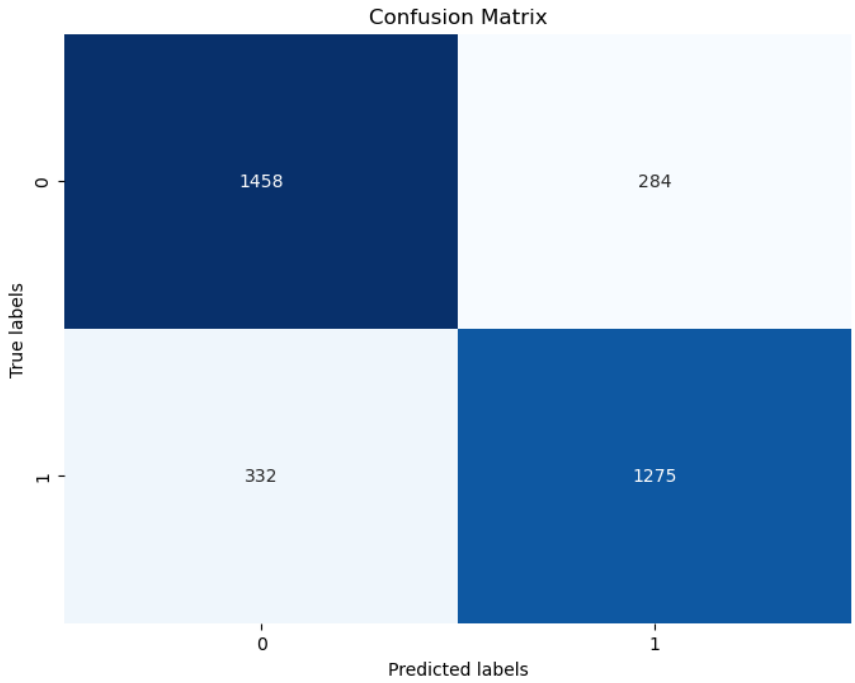
- Min Samples Leaf: Minimum number of samples required to be at a leaf node is set to 5.
- Min Samples Split: Minimum number of samples required to split an internal node is set to 30

2) Result Details:

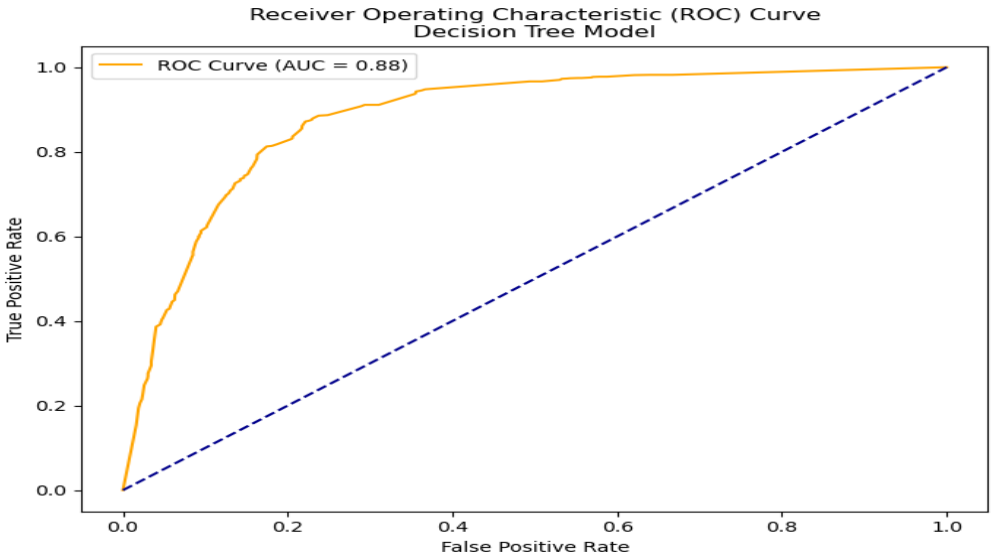
1) Model Accuracy :

- **Model accuracy**: 0.8160644968647357

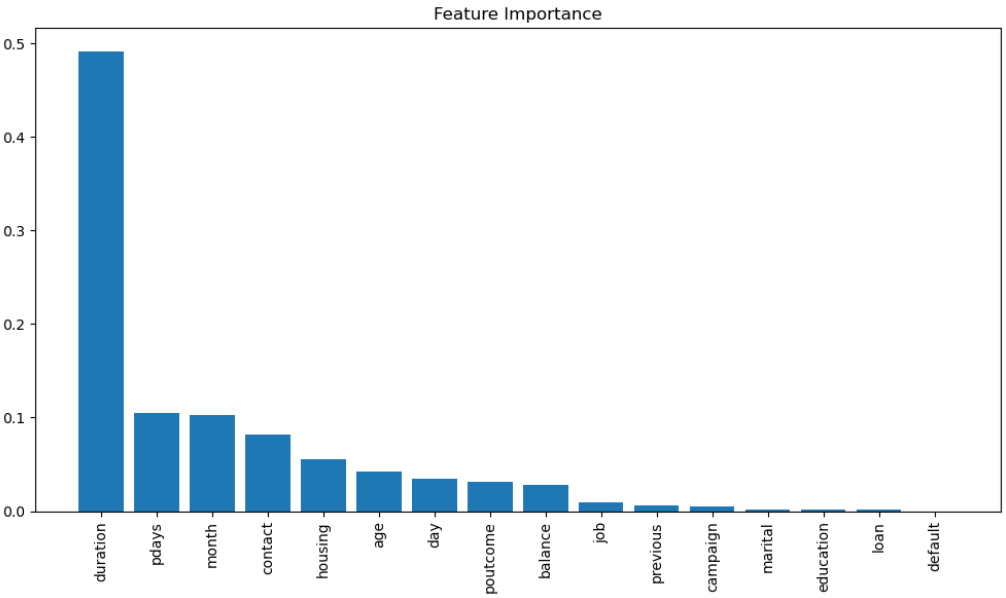
2) Confusion Matrix :



3) Receiver Operating Characteristic (ROC) Curve :



4) Feature Importance :



Neural Network Model:

1) Implementation Details :

1) Model Hyperparameters :

hidden layers	**3 hidden layers**
optmizer	**'adam'**
laearning_rate	**0.001**
epochs	**10**
batch_size	**25**
validation_split	**25%**

2) Feature Extraction:

```
X = df.drop(columns=['deposit'])  # Features ['age','job','marital','education','default','balance',housing','loan','contact',
                                  #           'day','month','duration','campaign','pdays','previous','poutcome']

y = df['deposit']  #Target
```

The dimension of the resulted features is mentioned in the output: 16 features were extracted.

3) Regularization :

Dropout regularization is used after the first hidden layer with a dropout rate of 0.3.

- Dropout is a technique used to prevent overfitting by randomly dropping a fraction of the input units (neurons) during training, which helps in generalization.

4) Batch Size:

The batch size is set to 25. This means that during each iteration of training, 25 samples are processed together before updating the weights of the neural network. Using mini-batch training helps in achieving a balance between computational efficiency and model convergence.

5) Number of Epochs:

The number of epochs is set to 10. An epoch refers to one complete pass through the entire training dataset.

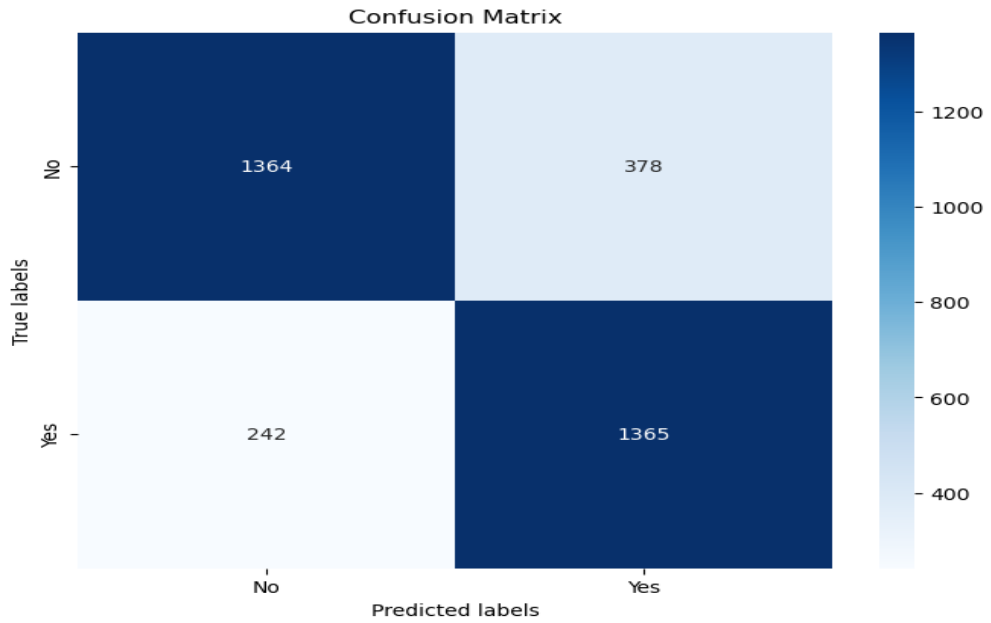
- Training for multiple epochs allows the model to see the entire dataset multiple times, enabling it to learn from the data and improve its performance over time.

2) Result Details:

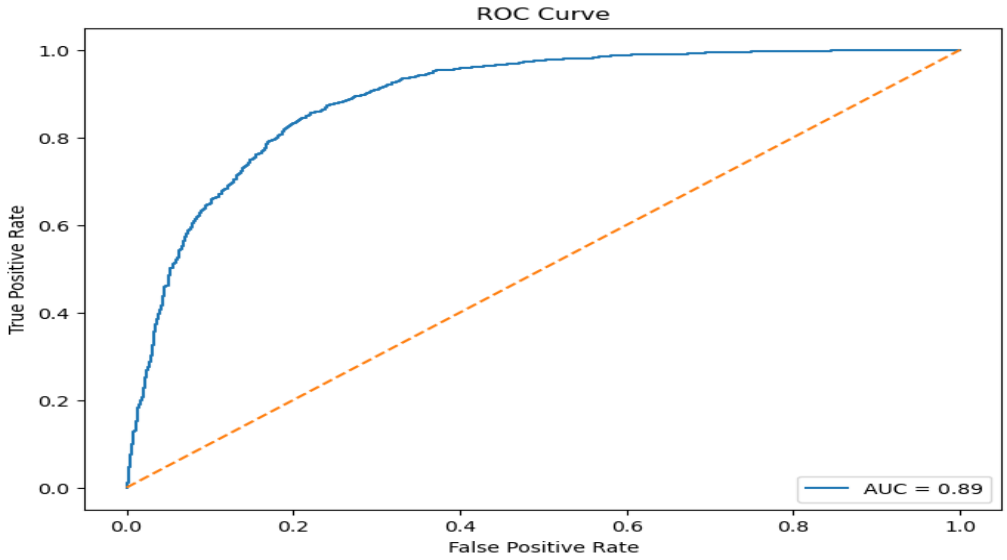
1) Model Accuracy :

- **Model accuracy**: 0.8109883666038513

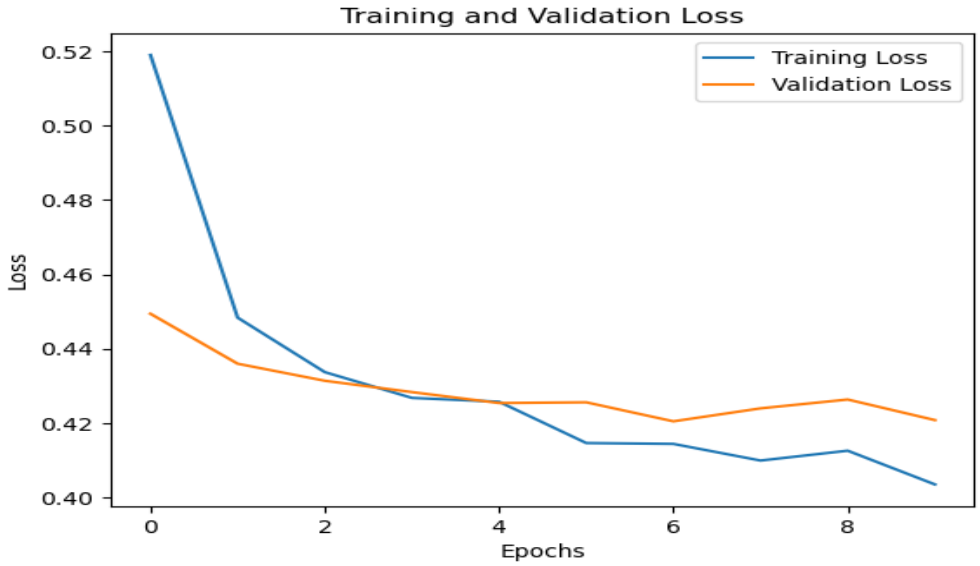
2) Confusion Matrix :



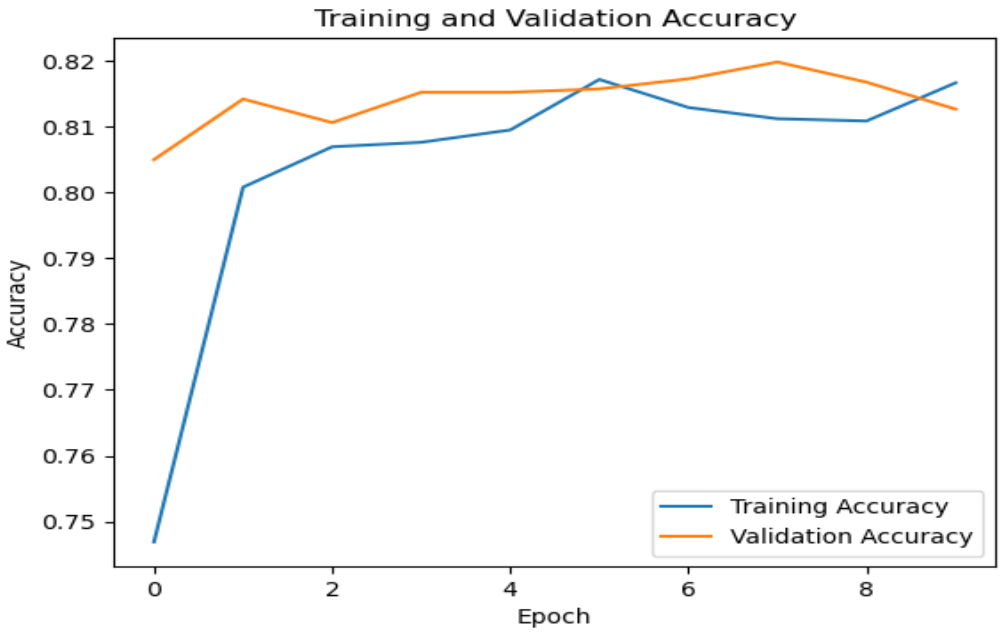
3) Receiver Operating Characteristic (ROC) Curve:



4) Training and Validation Loss :



5) Training and Validation Accuracy :



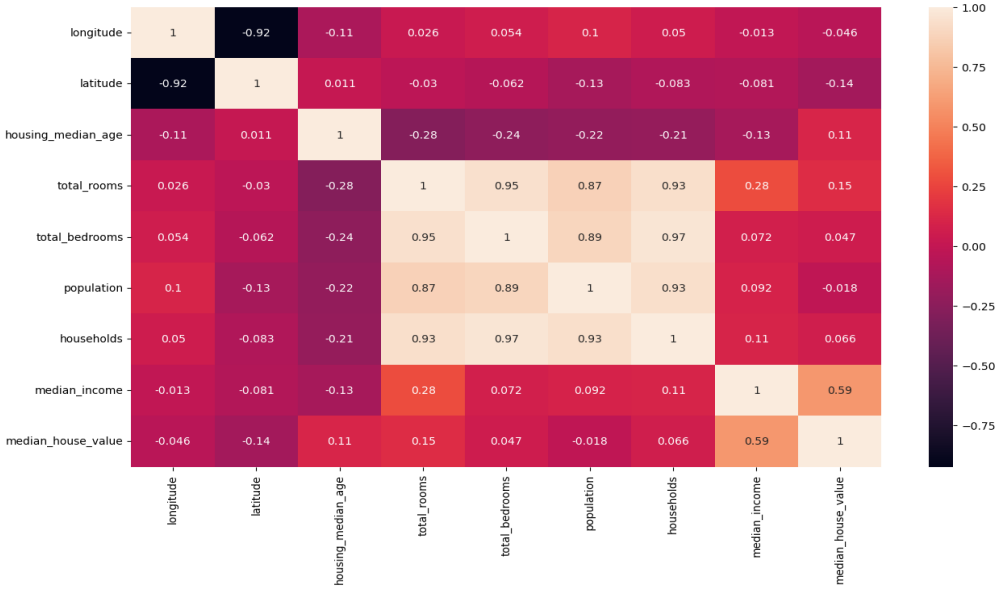
Regression dataset

Name :	• **California housing prices**
Total Size of Dataset :	• 20,640
Training Data Size :	• 16,512
Test Data Size :	• 4,128
Features :	• 9

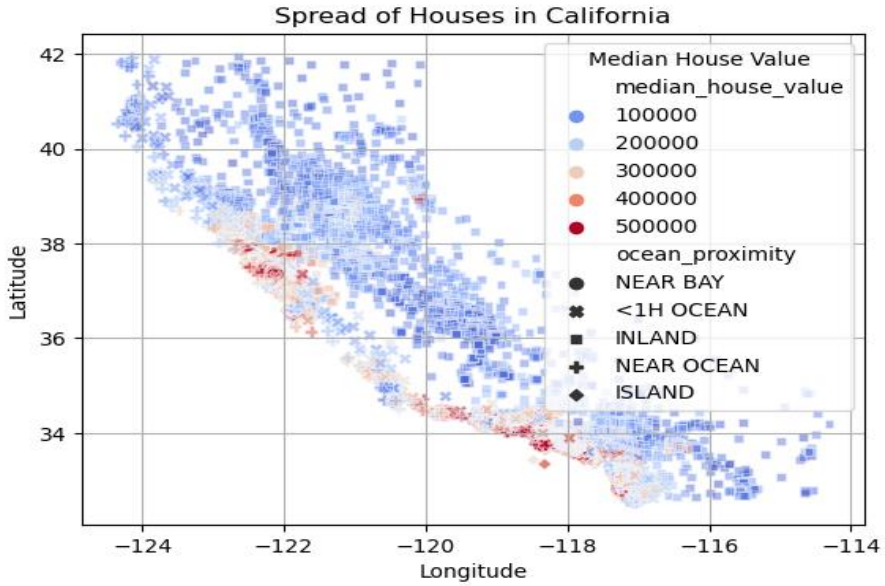
- after preprocessing I scaled the data and converted the categorical column by label encoder.

Data visualization:

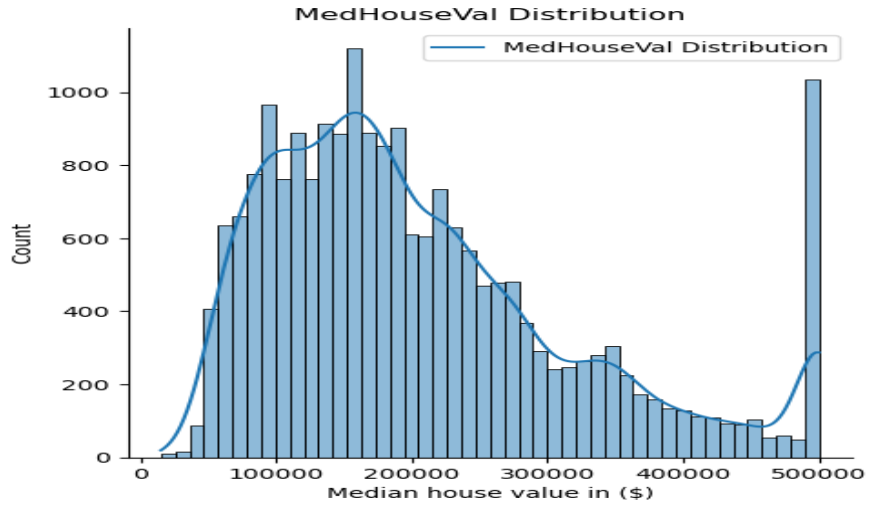
1) Heatmap :



2) Geographical Scatter Plot :

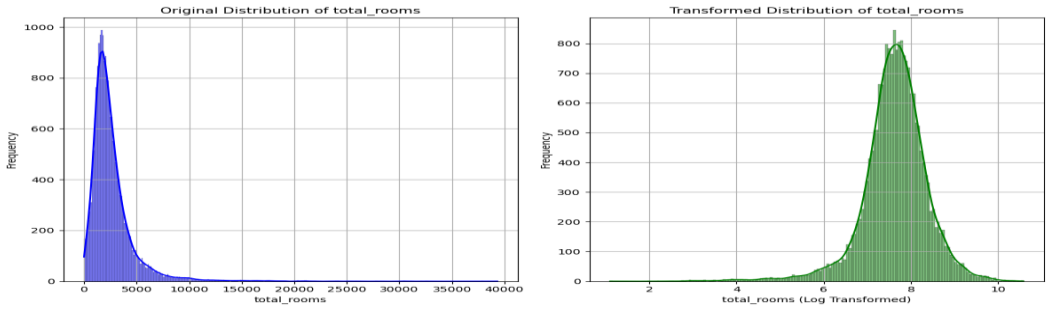


3) distribution plot represents the distribution of median house values Data :

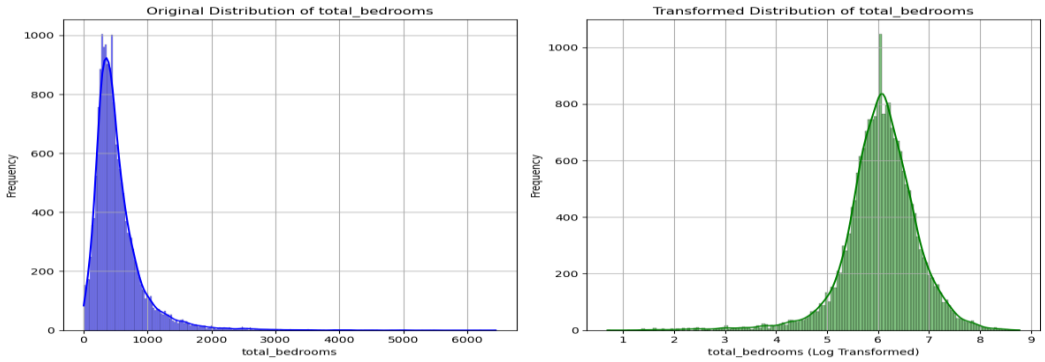


4) Log Transformation :

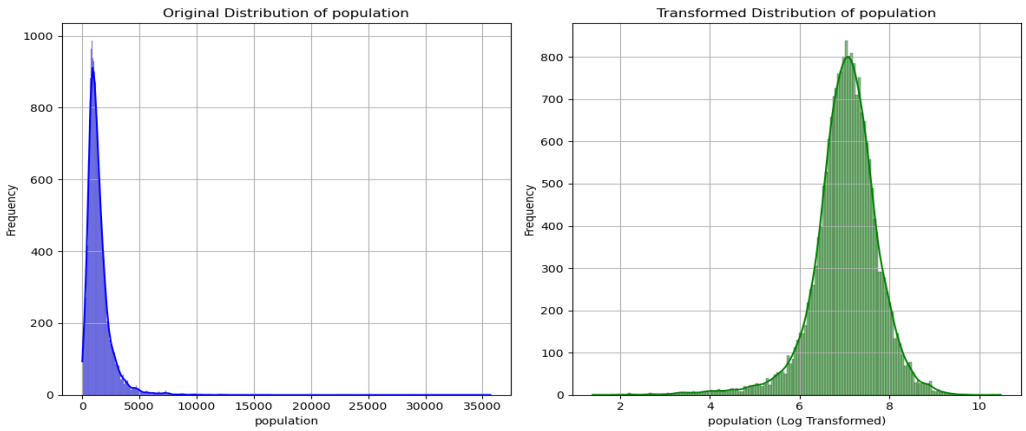
1)total_rooms:



2)total_bedrooms:

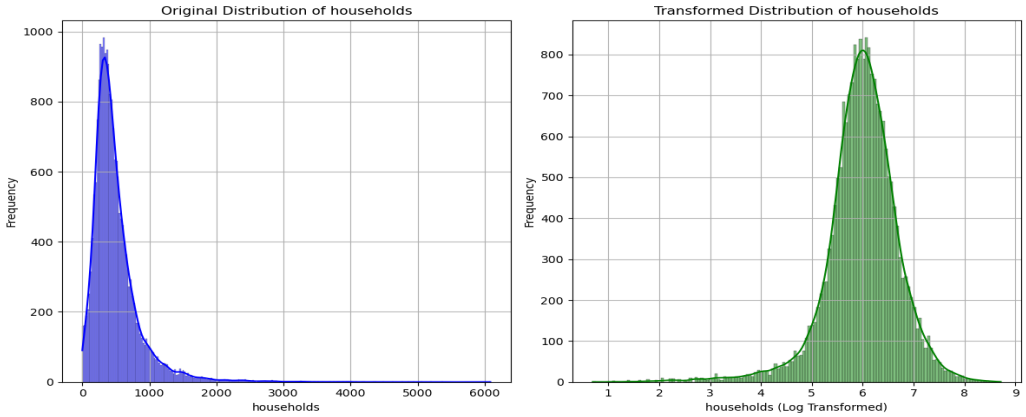


3)population:

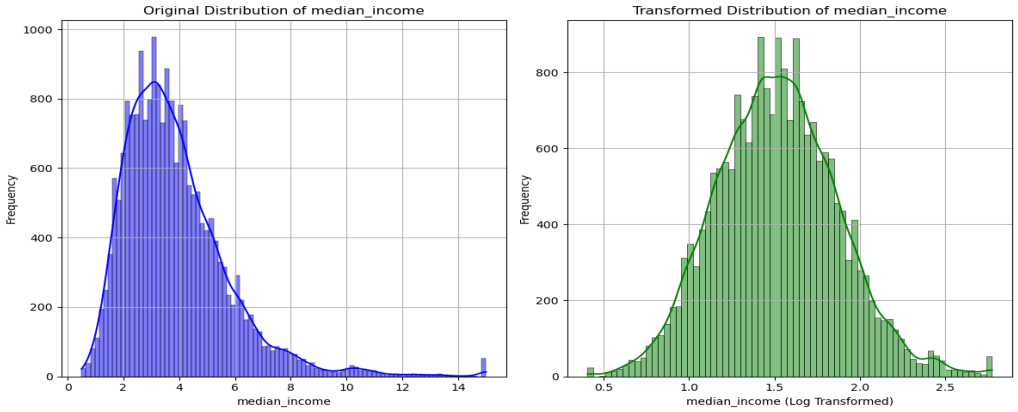


4) Log Transformation :

4) households:



5) median_income:



SVR Model:

1) Implementation Details :

1) Model Hyperparameters:

C	30
Gamma	'scale'
Kernel	'rbf'
Epsilon	0.2

2) Feature Extraction:

```
X_features = dataset.drop(['median_house_value'], axis = 1)  # Features ['longitude','latitude','housing_median_age','total_rooms','total_bedrooms'
                                                             #          'population','households','median_income','ocean_proximity']

y_target = dataset['median_house_value'] #Target
```

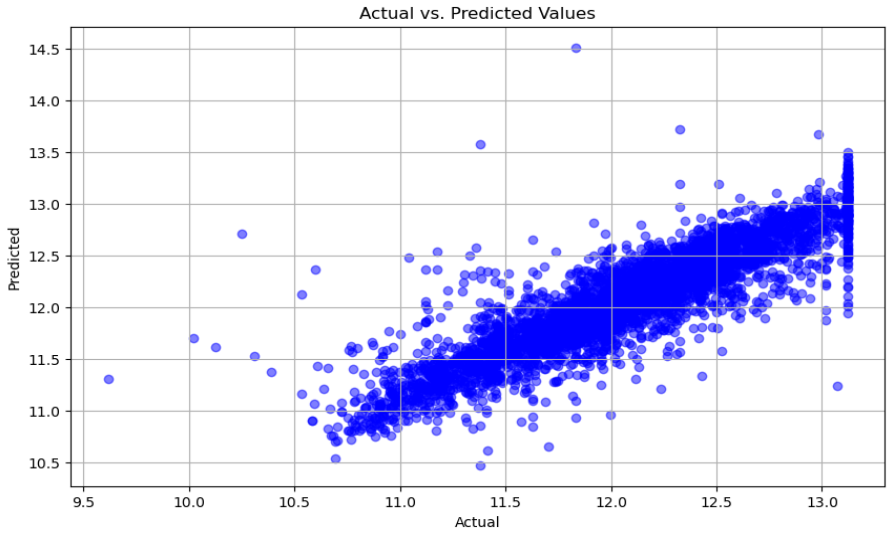
- Log transformation is applied to certain columns specified in log_columns.
- Label encoding is performed on the 'ocean_proximity' column using LabelEncoder from scikit-learn.

2) Result Details:

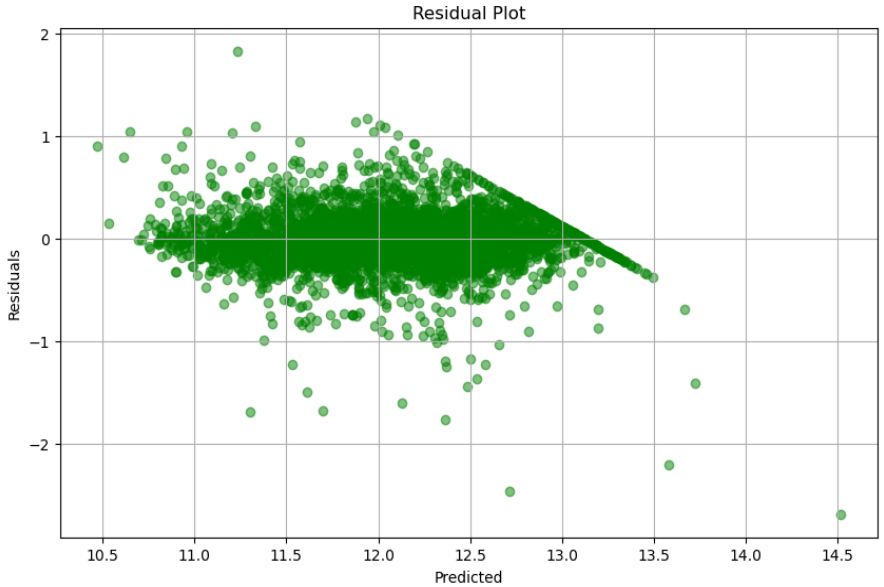
1) Model Accuracy :

- **train score**: 0.8588121584044552
- **test score:** 0.7800056178583221
- **Mean Squared Error:** 0.07180227341482881
- **R-squared:** 0.7800056178583221

2) Actual vs. Predicted Values:



3) Residual Plot:



4) Distribution of Residuals: