Documentation

SVM (classification):

General info on the dataset used:

- Name of the dataset: seed_dataset
- Number of classes: 2
- Labels are: Çerçevelik and Ürgüp Sivrisi.
- Number of samples: 2500
- Number of samples in training is 1750.
- Number of samples in validation is 500.
- Number of samples in testing is 750.

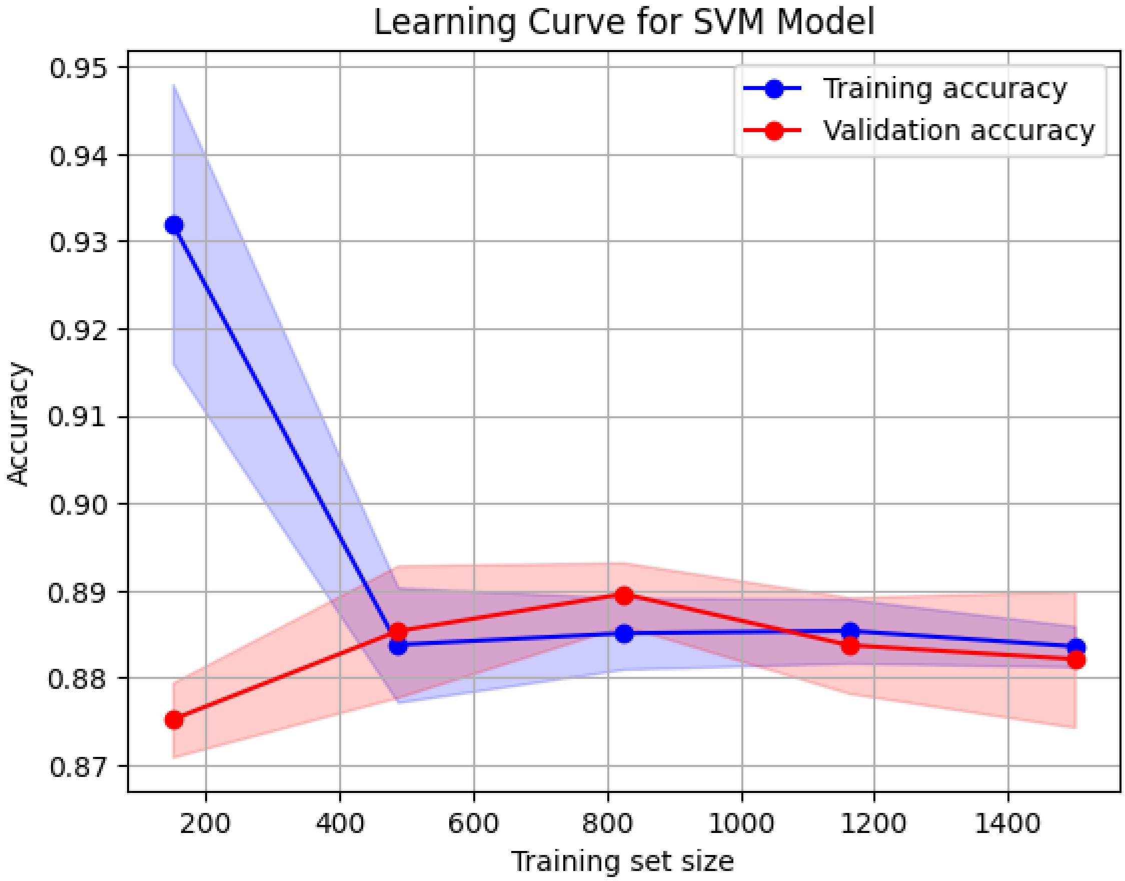
Implementation details:

- At feature extraction phase, 8 features are extracted, they are: Area, Perimeter, Major_Axis_Length, Minor_Axis_Length, Solidity, Extent, Roundness, Compactness.
- Cross-Validation **is** used in the model, and the number of fold is 5, and the ratio of training/validation is 4/1.
- Hyperparameters used in our model is:

- C Hyperparameter which controls the regularization strength
- Kernel which specifies the type of kernel used in the SVM model(linear).
- K which determines the number of subsets in which the data is divided for cross validation
- Shuffle which determines whether to shuffle the data before splitting it into folds. It's (true)→ data will be shuffled.
- Random state which is used to ensure that the same randomization is used each time you run the code.

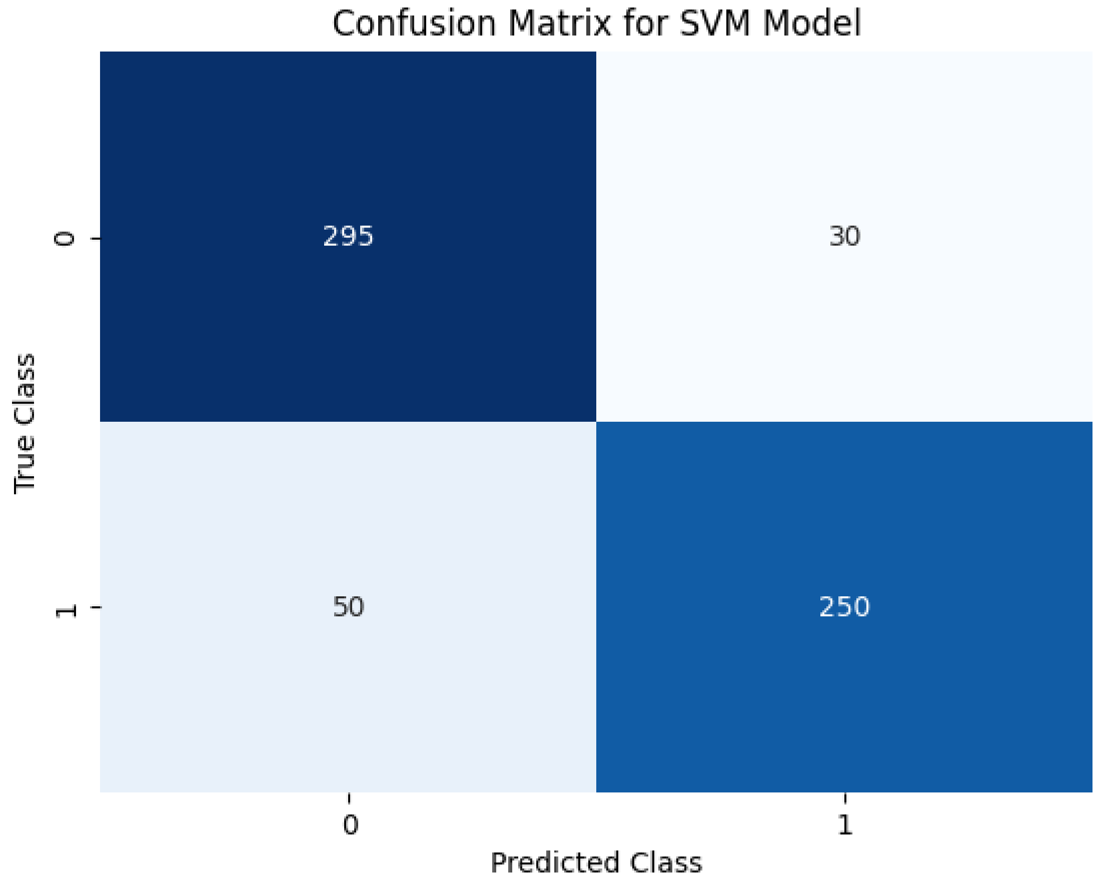
Results details:

Learning curve: It shows how error changes as the training set size increases

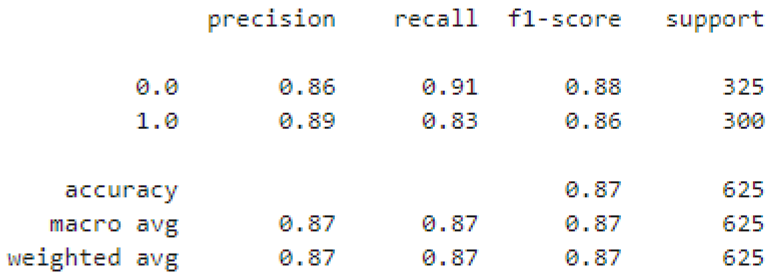


Accuracy: 0.872

Confusion Matrix: Visualizes the performance of the algorithm

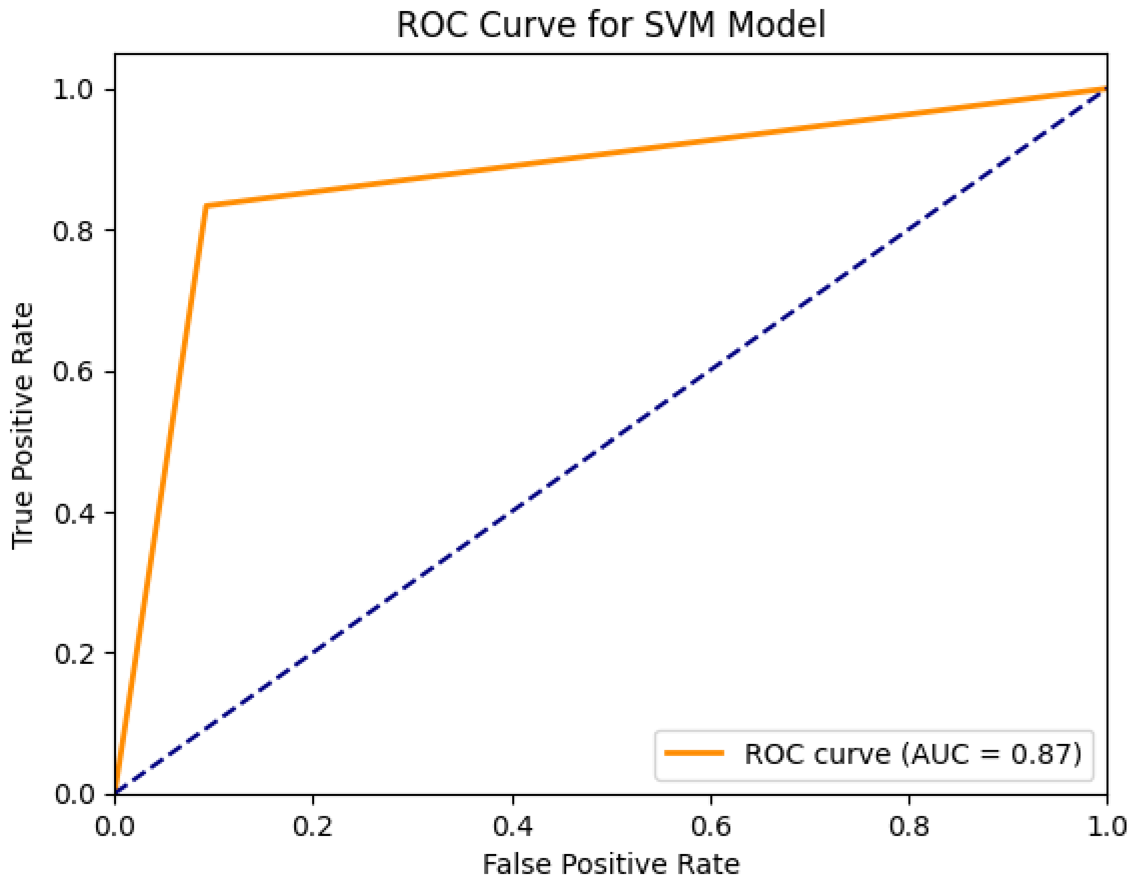


Classification Report: Represents the main metrics

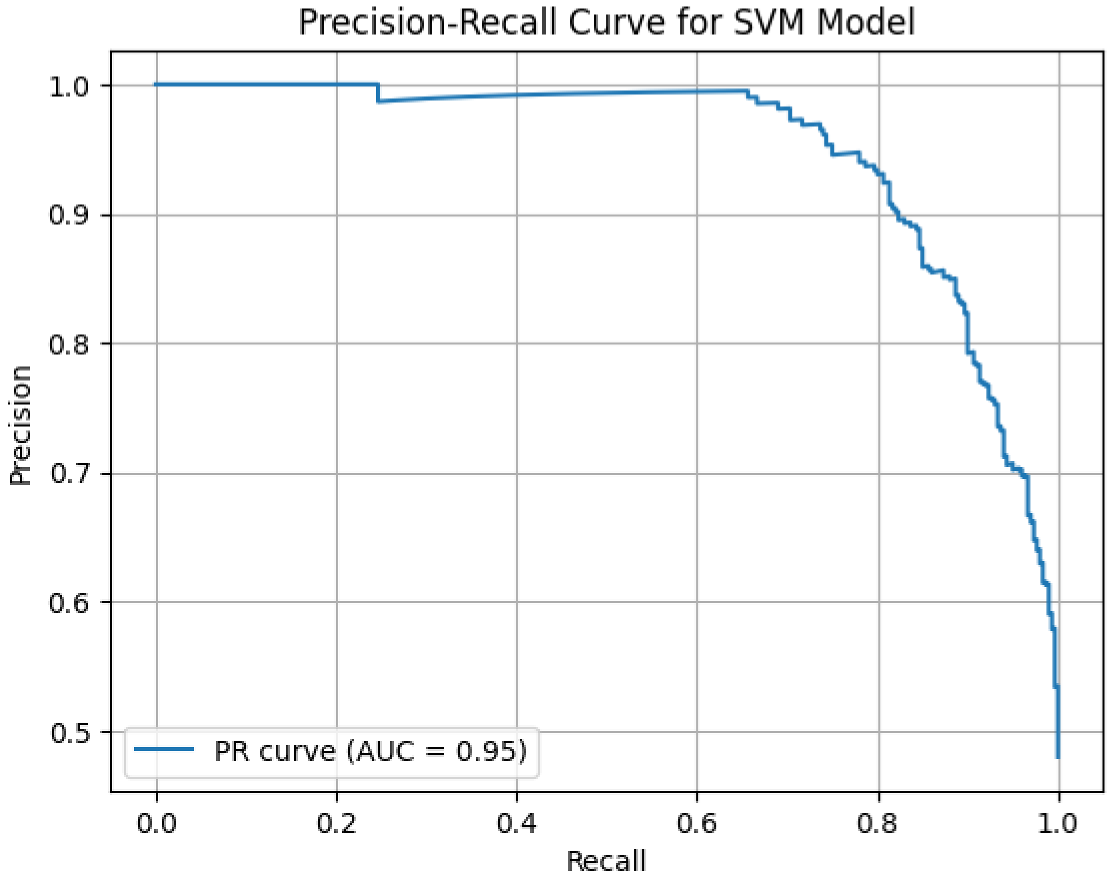


```
              precision    recall  f1-score   support

         0.0       0.86      0.91      0.88       325
         1.0       0.89      0.83      0.86       300

    accuracy                           0.87       625
   macro avg       0.87      0.87      0.87       625
weighted avg       0.87      0.87      0.87       625
```

Receiver Operating Characteristic (ROC) curve: It shows the connection between sensitivity and specificity, so it basically tells us how much the model is capable of distinguishing between classes



Precision-Recall curve: It shows the connection between precision and recall and, it provides insights into the model's ability to correctly classify the positive instances



Decision Tree (classification):

General info on the dataset used:

- Name of the dataset: seed_dataset
- Number of classes: 2
- Labels are: Çerçevelik and Ürgüp Sivrisi
- Number of samples: 2500
- Number of samples in training: 1750
- Number of samples in validation: 375
- Number of samples in testing: 375
- Number of features extracted: 9

Implementation details:

- Extracted features: ['Area', 'Perimeter', 'Major_Axis_Length', 'Minor_Axis_Length', 'Solidity', 'Extent', 'Roundness', 'Compactness', 'Class']
- Cross-Validation Scores: [0.81714286 0.82571429 0.84857143 0.82857143 0.82285714]
- Mean CV Score: 0.8285714285714285
- Number of Folds: 5
- Training Ratio (per fold): 0.8
- Validation Ratio (per fold): 0.2
- Hyperparameters used in our model is:

  Criterion: gini

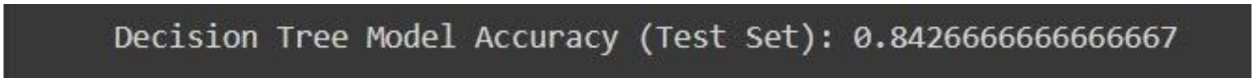
  Max Depth: None

Min Samples Split: 2

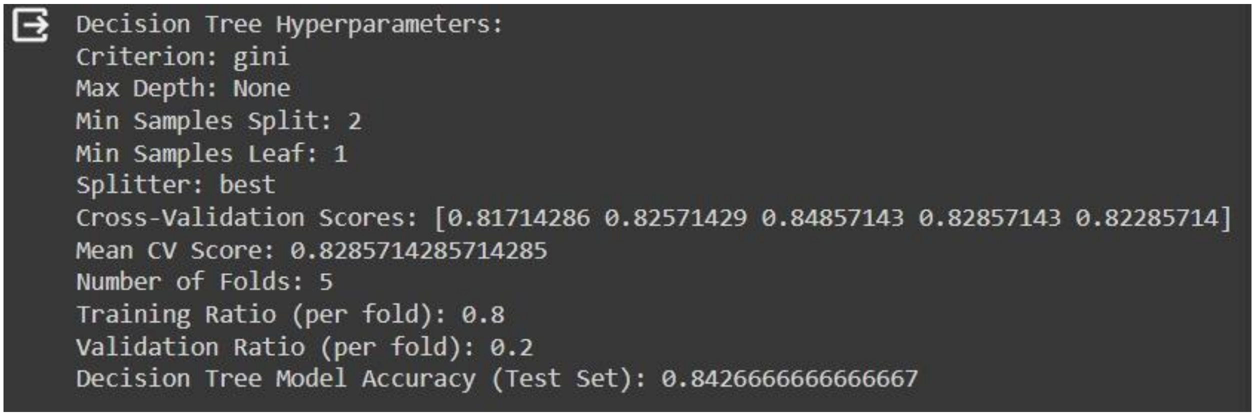
Min Samples Leaf: 1

Splitter: best

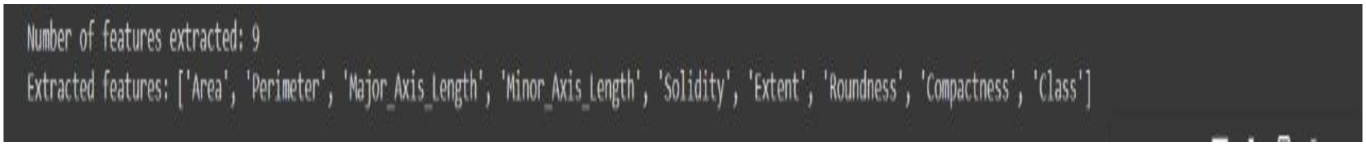
Results details:



```
Decision Tree Model Accuracy (Test Set): 0.8426666666666667
```



```
Decision Tree Hyperparameters:
Criterion: gini
Max Depth: None
Min Samples Split: 2
Min Samples Leaf: 1
Splitter: best
Cross-Validation Scores: [0.81714286 0.82571429 0.84857143 0.82857143 0.82285714]
Mean CV Score: 0.8285714285714285
Number of Folds: 5
Training Ratio (per fold): 0.8
Validation Ratio (per fold): 0.2
Decision Tree Model Accuracy (Test Set): 0.8426666666666667
```



```
Number of features extracted: 9
Extracted features: ['Area', 'Perimeter', 'Major_Axis_Length', 'Minor_Axis_Length', 'Solidity', 'Extent', 'Roundness', 'Compactness', 'Class']
```

Neural Network (Regression):

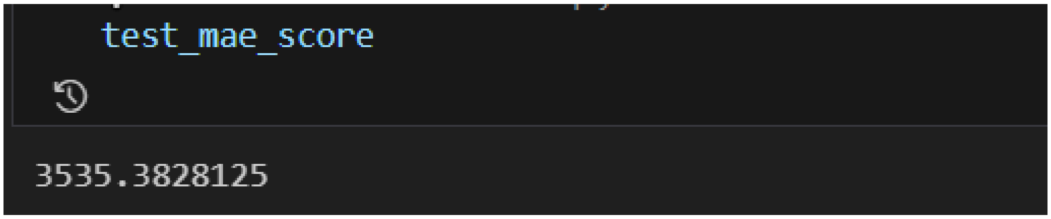
General info on the dataset used:

- Name of the dataset: vehicles
- Label: Price.
- Number of samples: 426,880
- Number of samples in training is 298,816.
- Number of samples in testing is 128,064.

Implementation details:

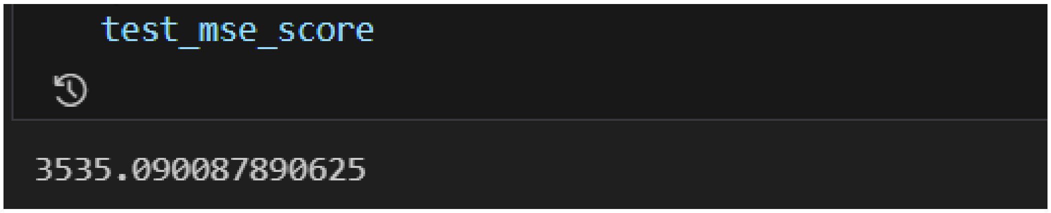
- At feature extraction phase: 12 features are extracted, they are: manufacturer, Model, condition, cylinders, fuel, odometer, title_status, transmission, drive, type, paint_color, and car_age.
- Model Architecture Hyperparameters:
    - Number of hidden layers: 2
    - Number of units in each hidden layer: 64
    - Activation function for hidden layers: ReLU (Rectified Linear Unit)
    - Output layer activation: Linear
- Training Hyperparameters:
    - Loss function: Mean Absolute Error (MAE)
    - Optimizer: RMSprop
    - Learning rate: 0.02
    - Number of epochs: 200
    - Batch size: 16

Results details:



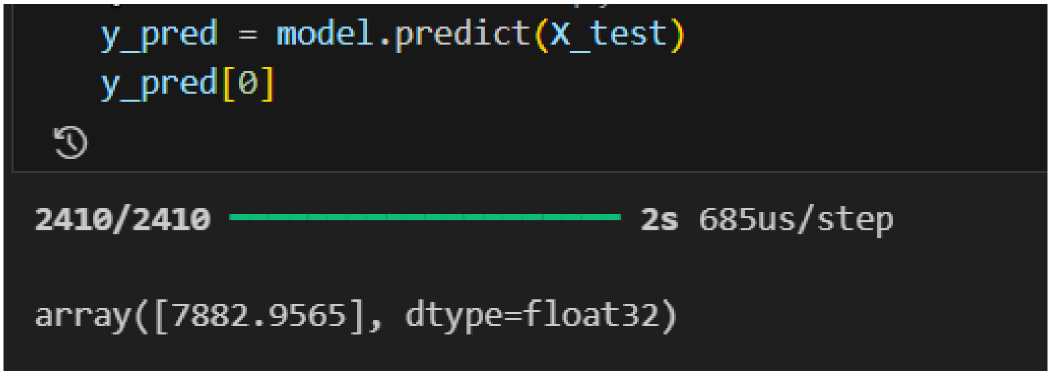
```
test_mae_score
```

```
3535.3828125
```



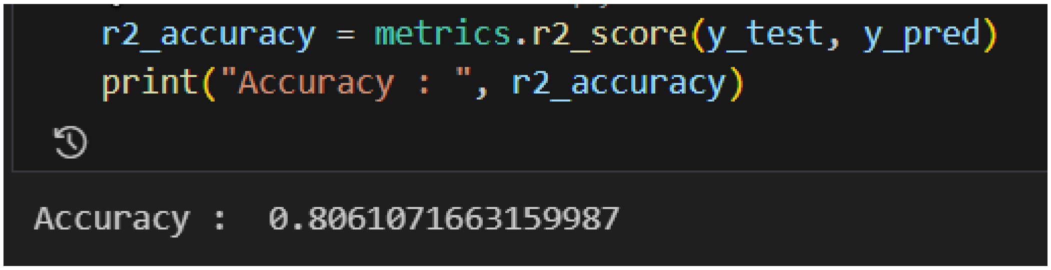
```
test_mse_score
```

```
3535.090087890625
```



```
y_pred = model.predict(X_test)
y_pred[0]
```

```
2410/2410 ━━━━━━━━━━━━━━━━━━━━ 2s 685us/step

array([7882.9565], dtype=float32)
```



```
r2_accuracy = metrics.r2_score(y_test, y_pred)
print("Accuracy : ", r2_accuracy)
```

```
Accuracy :  0.8061071663159987
```