Machine Learning

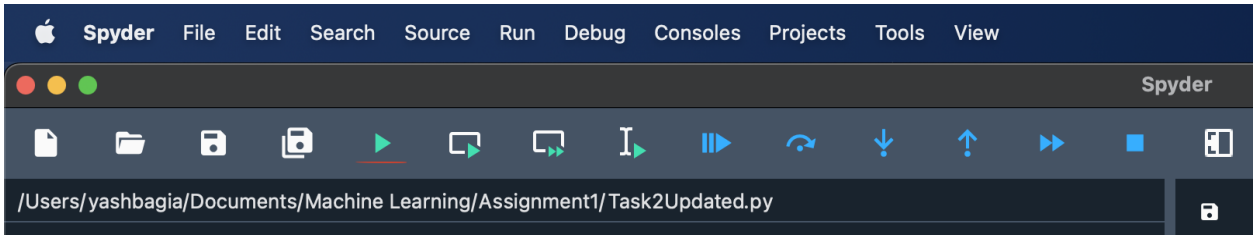
Assignment - 1

Yash Bagia
20395995

Running the code

Running the code is pretty simple:

1. Open Spyder IDE
2. Load the folder
3. Select among the 2 python files “Task1.py” or “Task2.py”.
4. Simply click the Cyan Play button or F5.



Solution Logic

Task 1:

This code performs several steps for regression analysis on a dataset containing blood pressure data:

1. Data Loading:
- The code loads the blood pressure dataset from a CSV file (`bloodpressure-23.csv`) into a Pandas DataFrame.
- It selects the feature variable 'SERUM-CHOL' as the predictor (`X`) and the target variable 'SYSTOLIC' as the response variable (`y`).

2. Polynomial Regression with Cross-Validation**:
- The code defines a function `calculate_rmse` that calculates the Root Mean Squared Error (RMSE) for polynomial regression of a specified degree using cross-validation.
- It iterates through polynomial degrees from 1 to 14 and computes the RMSE for each degree using 10-fold cross-validation.
- It identifies the best degree that minimises RMSE and stores it in the variable 'best_degree'.
- The RMSE values for different polynomial degrees 1-14 are printed and visualised in a plot.

3. Polynomial Regression Model Fitting:
- Using the best degree determined earlier, it fits a polynomial regression model to the data.
- The coefficients of the fitted model, including the intercept and polynomial terms, are printed.

4. Multiple Linear Regression:

- It performs linear regression on a subset of features ('AGE', 'ED-LEVEL', 'SMOKING STATUS', 'EXERCISE', 'WEIGHT', 'SERUM-CHOL', 'IQ', 'SODIUM') along with cross-validation to calculate RMSE.
- The coefficients of the linear regression model are printed, including the intercept and coefficients for each feature.

5. Ridge Regression:

- Ridge regression is performed on the same subset of features with a specified regularisation parameter (`alpha = 0.1`) using cross-validation to calculate RMSE.
- The coefficients of the Ridge regression model are printed, including the intercept and coefficients for each feature.

Task 2:

This code uses 2 datasets: the MNIST dataset and a sample text dataset (For language prediction).

1. MNIST Dataset Handling:

- The code begins by loading the MNIST dataset using TensorFlow and preprocesses it. It reshapes the image data, scales it, and converts the labels to binary values (1 for digit 6, 0 for other digits).

2. PCA for Dimensionality Reduction:

- The ‘perform_pca’ function is defined to perform Principal Component Analysis (PCA) on the MNIST dataset, reducing its dimensionality to achieve a 88% variance ratio.
- PCA is applied to both the training and testing data.

3. Logistic Regression on MNIST Dataset:
- Logistic regression is used as a classification model to predict whether a digit is 6 or not.
- The ‘train_and_evaluate_logistic_regression’ function is used to train and evaluate the logistic regression model.
- Metrics such as training and testing accuracy, classification report, confusion matrix, and misclassified samples are calculated and printed for the MNIST dataset.

4. Sample Text Data Handling:
- A sample text dataset containing multilingual texts and corresponding labels is defined.

5. Text Vectorization:
- The text data is transformed into numerical features using TF-IDF vectorization (`TfidfVectorizer`) to prepare it for machine learning.

6. Logistic Regression on Text Data:
- Logistic regression is applied to the TF-IDF transformed text data.
- The model is trained, and its accuracy and classification report are printed.

7. Misclassified Labels for Text Data:
- The code correctly extracts and prints the labels of misclassified samples in the text data, addressing a previous issue.

8. Predicted Labels for Text Data:
- The predicted labels for all test samples in the text data are printed.

9. Ridge Regression on MNIST Data:
- Ridge regression(a form of linear regression with L2 regularisation) is applied to the MNIST dataset.
- Metrics such as coefficients, intercept, and mean RMSE (Root Mean Squared Error) are calculated and printed for the Ridge regression model.

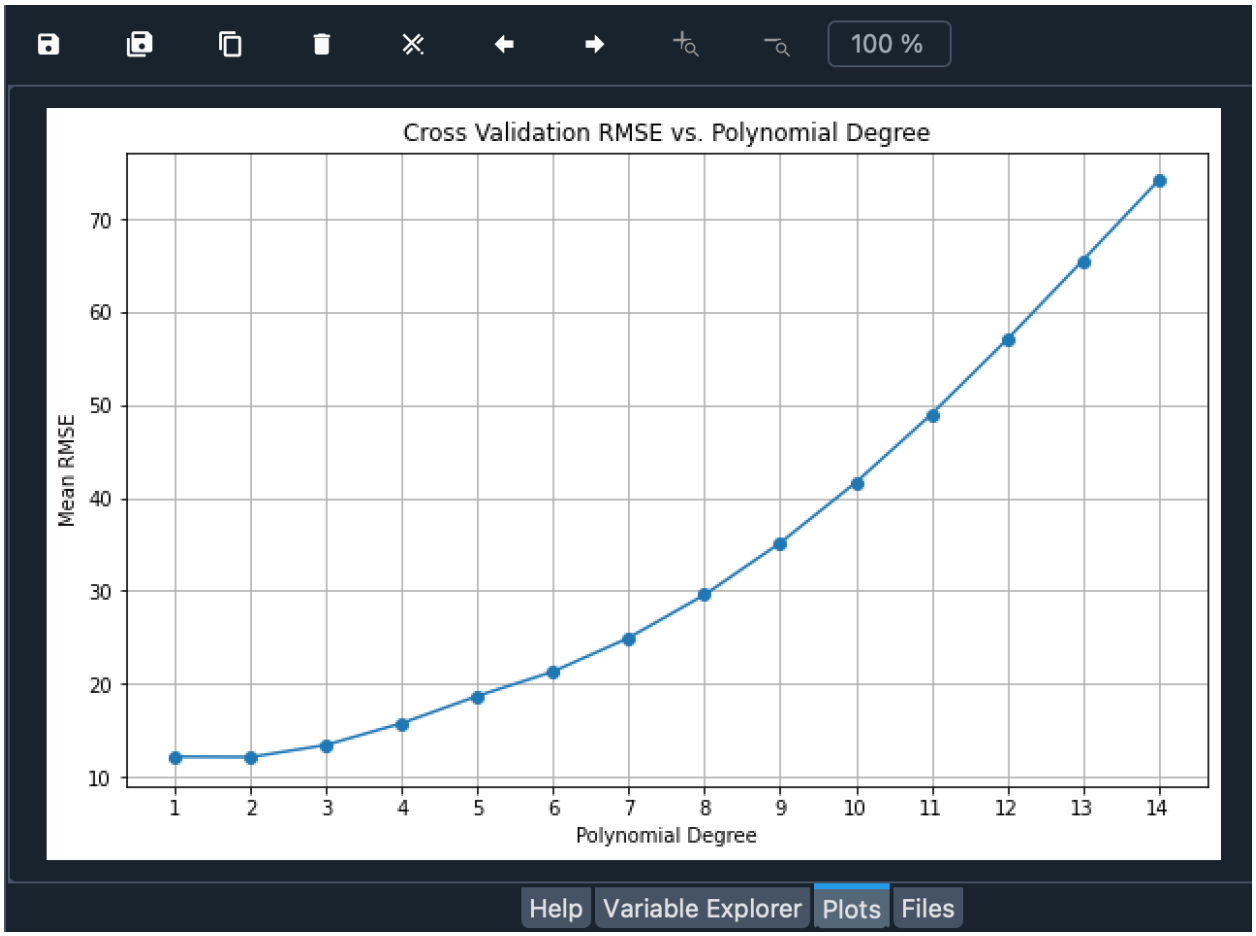
10. Printing Misclassified Digits for PCA Model:
- For the PCA-transformed MNIST dataset, misclassified digits are printed.
- The code correctly extracts and prints the misclassified digits.

In summary, this code demonstrates machine learning tasks such as logistic regression, PCA for dimensionality reduction, Ridge regression, and TF-IDF vectorization on two different datasets: the MNIST handwritten digits dataset and a sample text dataset. It calculates and prints relevant evaluation metrics for each task.

Test Run:

Task 1:

Here's the plot generated for task 1:



The plot uses polynomial degrees as X and Mean RMSE as Y for the plot.

Output:

runfile('/Users/yashbagia/Documents/Machine Learning/Assignment1/Task1.py',
wdir='/Users/yashbagia/Documents/Machine Learning/Assignment1')

Polynomial Regression:

Degree 1: RMSE = 12.165560461823437

Degree 2: RMSE = 12.127162327432321

Degree 3: RMSE = 13.436364879445975

Degree 4: RMSE = 15.77481212245977

Degree 5: RMSE = 18.70140826558649

Degree 6: RMSE = 21.322371318303823

Degree 7: RMSE = 24.95768182571313

Degree 8: RMSE = 29.571791569235657

Degree 9: RMSE = 35.14699897379459

Degree 10: RMSE = 41.644007470982544

Degree 11: RMSE = 48.98197727499172

Degree 12: RMSE = 57.01324530887654

Degree 13: RMSE = 65.49878218120813

Degree 14: RMSE = 74.08879984265866

Best Degree: 2

Intercept: 278.560362416353

Coefficients:

Coefficient 0: 0.0

Coefficient 1: -1.5484716054214105

Coefficient 2: 0.0039494790668519065

Coefficients (Multiple Linear Regression):

AGE: 0.44231020209792044

ED-LEVEL: -1.1325292450724316

SMOKING STATUS: -0.06743506009109418

EXERCISE: -1.0187941825920237

WEIGHT: 0.13386439288374738

SERUM-CHOL: 0.004394913289680476

IQ: -0.0018497953671015328

SODIUM: 0.20381035489416255

Intercept(Multiple Linear Regression): 66.68546356952064

Mean RMSE(Multiple Linear Regression): 8.509707572421089

Coefficients (Ridge Regression):

AGE: 0.4423841623235467

ED-LEVEL: -1.128232306368453

SMOKING STATUS: -0.06760809665768469

EXERCISE: -1.0175469004339361

WEIGHT: 0.13384909202360626

SERUM-CHOL: 0.004434042864640476

IQ: -0.002077799560400274

SODIUM: 0.20372362672252464

Intercept (Ridge Regression): 66.70686764934332

Mean RMSE (Ridge Regression): 8.50873659229708

Task 2:

Here's the output for task 2:

runfile('/Users/yashbagia/Documents/Machine Learning/Assignment1/Task2Updated.py', wdir='/Users/yashbagia/Documents/Machine Learning/Assignment1')

MNIST Results:

Number of Principal Components Preserved: 784

Training Accuracy: 0.99

Testing Accuracy: 0.99

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	9042
1	0.93	0.91	0.92	958
accuracy			0.99	10000
macro avg	0.96	0.95	0.96	10000
weighted avg	0.99	0.99	0.99	10000

Confusion Matrix:

[[8979 63]

[82 876]]

Number of Misclassified Samples: 145

Misclassified Digits: [0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 1 1 1

1 1 1 1 1 0 0 1 1 1 0 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1 1]

Language Prediction Results:

Number of Principal Components Preserved: 40

Training Accuracy: 1.00

Testing Accuracy: 0.00

Classification Report:

	precision	recall	f1-score	support
English	1.00	0.00	0.00	1.0
French	1.00	0.00	0.00	1.0
German	0.00	1.00	0.00	0.0
Spanish	1.00	0.00	0.00	1.0
accuracy			1.00	3.0
macro avg	0.75	0.25	0.00	3.0
weighted avg	1.00	0.00	0.00	3.0

Number of Misclassified Samples: 3

Misclassified Labels: ['French' 'English' 'Spanish']

Predicted Labels: ['German' 'German' 'German']

PCA Results:

Number of Principal Components Preserved: 73

Training Accuracy: 0.99

Testing Accuracy: 0.81

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.90	0.89	12604
1	0.00	0.00	1.00	1396
accuracy			0.81	14000
macro avg	0.45	0.45	0.95	14000
weighted avg	0.80	0.81	0.90	14000

Confusion Matrix:

[[11296 1308]

[1396 0]]

Number of Misclassified Samples: 2704

Misclassified Digits: [0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0]

Answers

Task 1:

- **3.** The best degree would be 2 as it has the lowest RMSE: 12.127 and as the plot shows, the RMSE keeps increasing exponentially from degree 3 onwards making degree 2 the “elbow” of the curve.
  The intercept would be: 278.560 and the as it is a quadratic polynomial second degree equation, it has 3 coefficients:
  Coefficient 0: This is the intercept term.
  Coefficient 1: This is the coefficient of the linear term (x).
  Coefficient 2: This is the coefficient of the squared term (x^2).

  Coefficient 0: 0.0
  Coefficient 1: -1.5484716054214105
  Coefficient 2: 0.0039494790668519065

- 6. The best model is the Ridge Regression model as it has the lowest RMSE which indicates better predictive performance. Though it is very close to Multiple Linear Regression, it does have slightly lower RMSE which makes it the best model among the 3.

  Polynomial Regression RMSE: 12.127162327432321

  Multiple Linear Regression RMSE: 8.509707572421089

  Ridge Regression RMSE: 8.50873659229708

Task 2:

Let's analyse the model on three different datasets and their results:

1. MNIST Results:
- Training Accuracy: 0.99
- Testing Accuracy: 0.99

This model has a high training and testing accuracy, both at 0.99. This suggests that the model is performing very well on the MNIST dataset. The precision, recall, and F1-score for both classes (0 and 1) are also high, indicating good performance. The confusion matrix shows very few misclassified samples (145 out of 10,000), which is a small error rate.
Overall, this model appears to be a good fit for the MNIST dataset.

2. Language Prediction Results**:
- Training Accuracy: 1.00
- Testing Accuracy: 0.00

This model exhibits a training accuracy of 1.00, indicating that it has perfectly learned the training data. However, the testing accuracy is 0.00, which suggests that the model completely fails to generalise to unseen data. The precision, recall, and F1-score for all classes are also problematic, and the model misclassified all testing samples. This is a clear case of **overfitting**, where the model has memorised the training data but cannot generalise to new data.

3. PCA Results:
- Training Accuracy: 0.99
- Testing Accuracy: 0.81

This model has a high training accuracy of 0.99 but a lower testing accuracy of 0.81. While the training accuracy is high, the testing accuracy suggests that the model may not generalise well to unseen data. The precision, recall, and F1-score for the second class (1) are particularly problematic. The confusion matrix shows a significant number of misclassified samples (2704 out of 14,000), indicating that the model is struggling to correctly classify the second class. This could be a sign of **overfitting** or a model that is not well-suited to the dataset.

To summarise:

- The MNIST model appears to be a **good** fit for the dataset, with high training and testing accuracy.
- The Language Prediction model is **overfitting** the training data and performs poorly on testing data.
- The PCA model may be **overfitting** as well, as indicated by the significant difference between training and testing accuracy and the poor performance on the testing set.

Limitations:

There are a few limitations of the model:

1. The model performs bad in language prediction, especially German. From the sample dataset, the model predicted 3 samples as German while the true label was English, French and Spanish.
   It was perfect in training but it needs to be fine-tuned for better test results. The sample data also needs to be of better quality to avoid over-fitting.