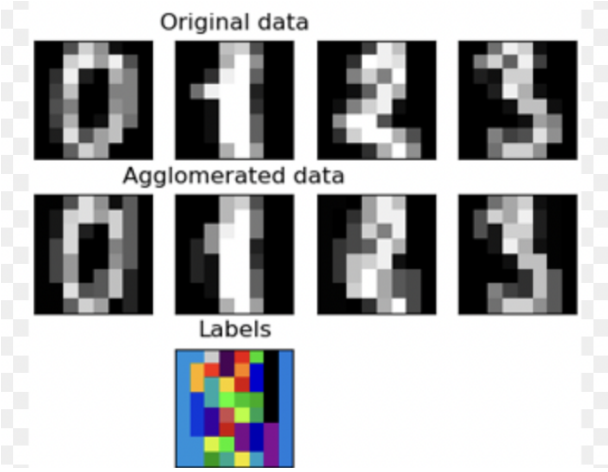
CMPUT 466 Mini-Project Report

Introduction:

The purpose of this report is to compare the results of four different models on the Digits dataset, and to provide a justification for each model's performance.

Methods:

Dataset: Here various machine learning models are implemented on the digits dataset. The digits dataset is a set of 8x8 images of handwritten digits ranging from 0 to 9. The objective of the models is to correctly classify the digit in each image.



Each datapoint is a 8x8 image of a digit.

Classes	10
Samples per class	~180
Samples total	1797
Dimensionality	64
Features	integers 0-16

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html

Steps:

- Four different models were tested on the Digits dataset, including Logistic Regression (multiclass classifier), SVM, KNN, and Linear Regression.

- The data was split into training, validation and test sets. 60% of the data used for training and then 20% of the data used for validation. At last, 20% of the data used for testing.

- Next, the data is scaled to the range [0, 1] using division by 16.0. This is done to ensure that all features have similar ranges, which can help improve model performance.

- The code then trains three models, Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), on the training set using the default hyperparameters. After training the models, they are hyperparameter tuned using a grid search approach, which tries different combinations of hyperparameters from a predefined list of values. The best hyperparameters for each model are chosen based on the accuracy score on the training set. The hyperparameters tuning process is done using the *GridSearchCV* function from scikit-learn's *model_selection* module, and the accuracy metric is used for scoring.

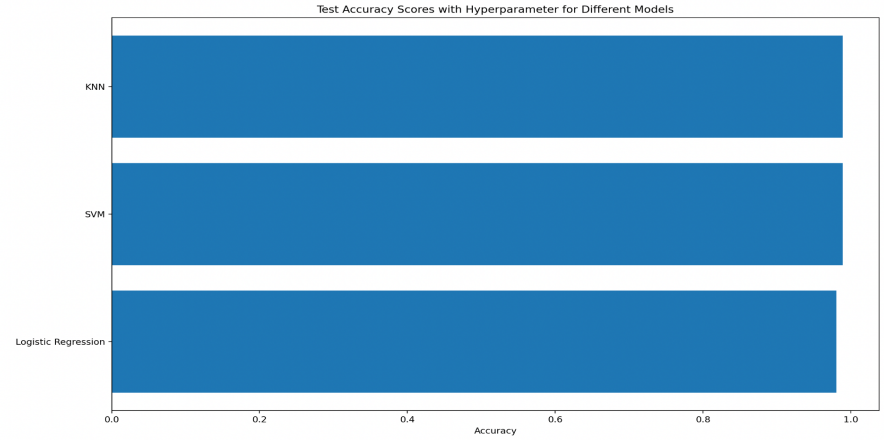
- Then, the code evaluates the models' performance on the test set using the best hyperparameters obtained from the grid search. The accuracy score is computed for each model using scikit-learn's *accuracy_score* function. A bar graph is plotted using Matplotlib to compare the models' performance. The same process is repeated for the models without hyperparameter tuning, and the results are plotted to compare the performance with and without hyperparameter tuning.

- The code also evaluates the models' performance on the validation set. This is done to see how the models generalize to unseen data. The accuracy score is computed using scikit-learn's *accuracy_score* function, and the results are plotted and compared using Matplotlib.

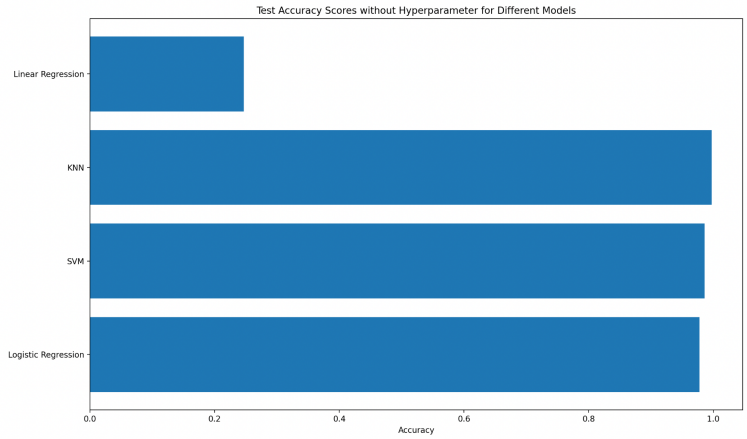
- Finally, the code calculates the training loss for the logistic regression (*log_loss*) and linear regression models (*mean_squared_error*). The results are compared by using plots

Results:

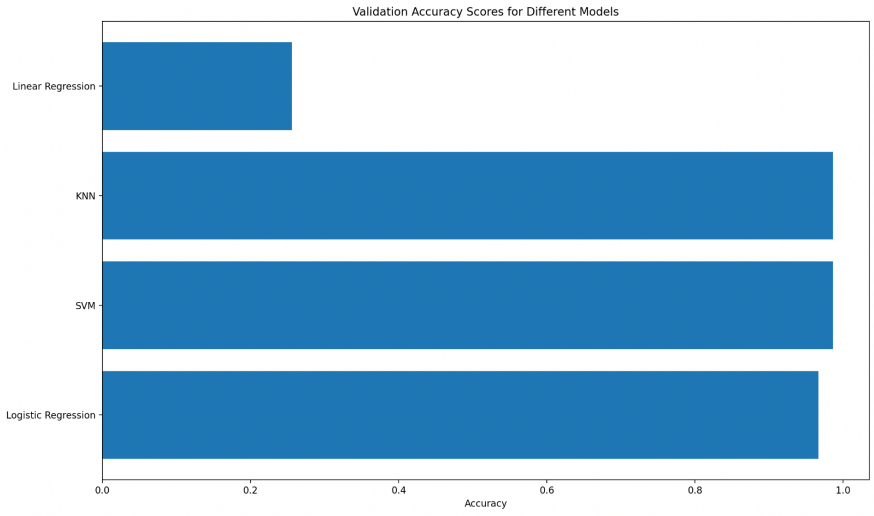
Test Accuracy Scores with hyperparameters for Logistic Regression, SVM, and KNN:



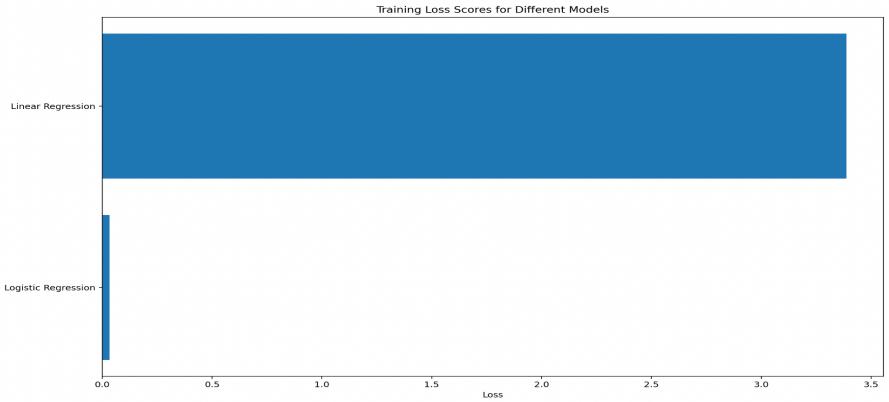
Test Accuracy Scores without hyperparameters for Linear Regression, Logistic Regression, SVM, and KNN:



The Validation Accuracy Scores for Different Models:



Training Loss Scores for Different Models:



The results for the four models are as follows:

- Best logistic regression hyperparameters: {'C': 10}
- Best SVM hyperparameters: {'C': 10, 'gamma': 'scale'}
- Best KNN hyperparameters: {'n_neighbors': 3}

- Logistic Regression Test Accuracy with Hyperparameter: 0.9805555555555555
- SVM Test Accuracy with Hyperparameter: 0.9888888888888889
- KNN Test Accuracy with Hyperparameter: 0.9888888888888889

- Logistic Regression Test Accuracy without Hyperparameter: 0.9777777777777777
- SVM Test Accuracy without Hyperparameter: 0.9861111111111112
- KNN Test Accuracy without Hyperparameter: 0.9972222222222222
- Linear Regression Test Accuracy without Hyperparameter: 0.24722222222222223

- Logistic Regression Validation Accuracy: 0.9665738161559888
- SVM Validation Accuracy: 0.9860724233983287
- KNN Validation Accuracy: 0.9860724233983287
- Linear Regression Validation Accuracy: 0.2562674094707521

- Logistic Regression Training Loss: 0.033629023620477025
- Linear Regression Training Loss: 3.3869448364152532

Reasoning:

The highest validation and testing accuracy score was achieved by SVM and KNN models. The SVM model is a non-linear classification model that tries to find the best decision boundary that can separate the data into different classes. The SVM model uses a kernel function to map the data into a higher-dimensional space to find a linear decision boundary. The KNN model is a non-parametric classification model that uses the distance between data points to classify the data. It finds the k-nearest neighbors to a data point and assigns the most common class label among these neighbors to the data point. Also, here KNN model's test accuracy without hyperparameter is better than with hyperparameter which is shocking. One reason could be that the hyperparameter tuning process may have caused overfitting in this case.

The validation and test accuracies achieved by the logistic regression model is also good but slightly less than SVM and KNN. The logistic regression model is a linear classification model that can be used for binary or multiclass classification problems. It tries to find the best decision boundary that can separate the data into different classes.

Linear Regression model performed poorly because the linear regression model is a regression model used for predicting continuous values. In this code, linear regression is used for classification, which is not its primary purpose. Training loss of linear regression is also much higher. On the other hand, training loss for logistic regression is much less which indicates that it is a better model.

Conclusion:

In conclusion, KNN and SVM models were the most accurate models for the Digits dataset. Linear Regression performed poorly, as it is not suitable for classification tasks.