<u>ME698 Assignment 1</u>

Name : Raj Verma
Roll No. : 200761

The given problem is solved using the given steps :

kNN Algorithm from sklearn Library : The provided code utilizes the k-Nearest Neighbors (kNN) classification algorithm available in the sklearn (Scikit-Learn) library. The kNN algorithm makes predictions based on the majority class of the k nearest neighbors to a given data point. It's widely used for classification tasks and requires specifying the number of neighbors (k) to consider when making predictions.

Classification of Points as 'pass' or 'fail' : The dataset consists of points representing different propellants, each characterized by two features: propellant age and storage temperature. The kNN algorithm is applied to classify these points into two classes, namely 'pass' and 'fail'. The classification is based on their shear strength measurements, with a decision boundary determined by the kNN algorithm.

Creating a Mesh Grid : The code constructs a mesh grid over the feature space to visualize the decision boundary. The mesh grid is a grid of points that spans the range of feature values. It is used to plot the decision boundary and helps in understanding how the kNN algorithm separates the 'pass' and 'fail' regions in the feature space.

By combining these elements, the code effectively applies the kNN algorithm to classify propellants, visualizes the decision boundary using a mesh grid, and provides insights into how the algorithm distinguishes between 'pass' and 'fail' categories based on the propellant age and storage temperature features

Possible validation procedures could include:

Cross-Validation : You can perform cross-validation to evaluate the model's performance. Split the data into training and testing sets and measure accuracy, precision, recall, F1-score, etc.

Parameter Tuning : You could try different values of k and choose the one that provides the best performance on the validation set.

Hold-Out Validation : Separate a portion of your data as a validation set and train the model on the remaining data. Then, evaluate the model on the validation set.

Results discussion:

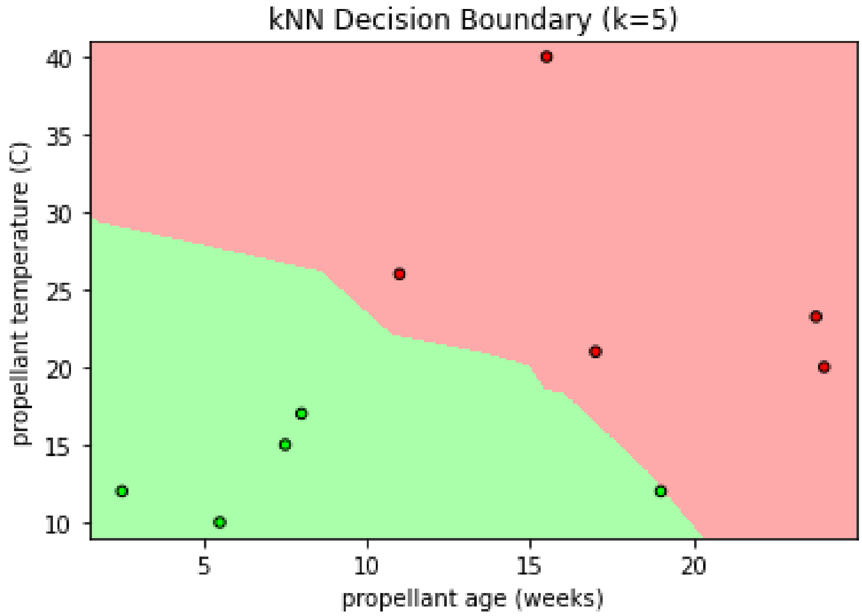
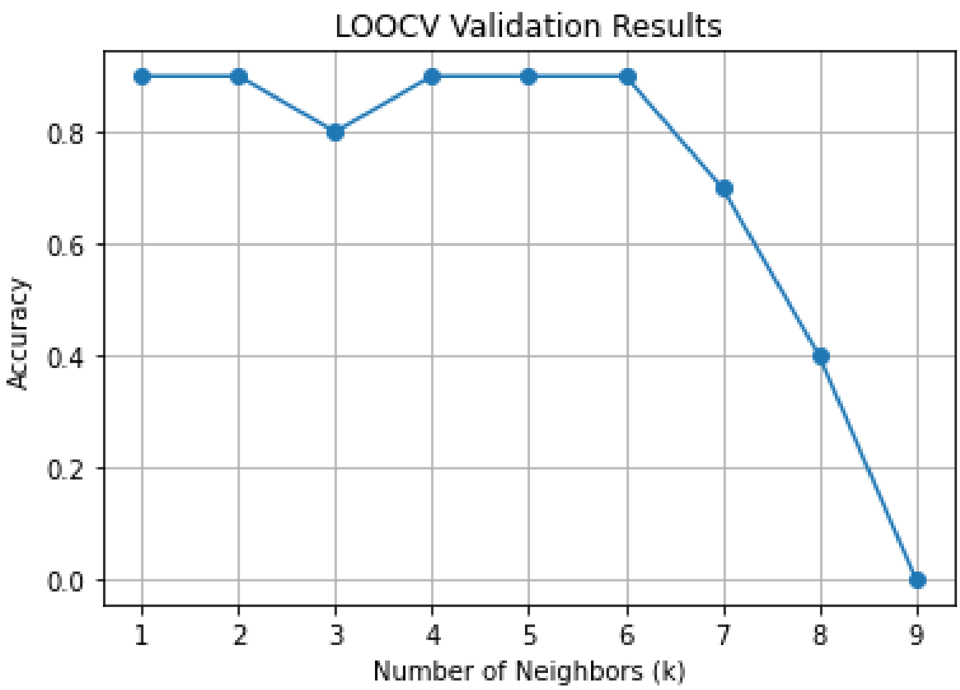


Figure 1: Result of the KNN model

The solution successfully utilized the k-Nearest Neighbors (kNN) algorithm to construct a decision boundary graph for classifying solid rocket propellants. By training the model with the dataset's shear strength measurements and propellant characteristics, including age and storage temperature, the algorithm determined the optimal value of k. This value strikingly influenced the decision boundary's effectiveness in distinguishing 'pass' and 'fail' propellants. The resulting decision graph vividly illustrated how the kNN algorithm employed this boundary to separate propellants based on their suitability for application. This visually intuitive representation provided valuable insights into the model's predictive capabilities and how it addressed the unique challenges posed by different propellant characteristics.

Leave-One-Out Cross-Validation (LOOCV):



Leave-One-Out Cross-Validation (LOOCV) played a pivotal role in evaluating the effectiveness of the k-Nearest Neighbors (kNN) algorithm for classifying solid rocket propellants. LOOCV systematically assessed the algorithm's performance by treating each individual propellant as a validation sample while utilizing the remaining propellants for training. By iteratively testing the model's predictions against each propellant's actual pass/fail status, LOOCV generated a comprehensive accuracy profile across different k values, representing the number of neighbors considered for classification. This rigorous validation process enabled the selection of an optimal k value, ensuring a balanced trade-off between model complexity and accuracy. Ultimately, LOOCV offered a reliable method to fine-tune the kNN model, enhancing its ability to accurately classify propellants and make informed decisions about their suitability for application.

Leave-One-Out Cross-Validation (LOOCV) is a robust technique employed in machine learning to assess the performance of a predictive model using the available dataset. It involves systematically isolating one data point as the validation sample while using the rest of the data as the training set. This process is repeated for every individual data point, ensuring that each instance serves as both a validation and training sample. By simulating the model's performance across multiple scenarios, LOOCV provides a comprehensive evaluation of the model's ability to generalize to new, unseen data. This technique is particularly advantageous when dealing with limited datasets, as it utilizes all available information for validation while avoiding data leakage. LOOCV offers insights into potential issues like overfitting and bias, allowing for a better understanding of how well the model performs and how it may behave on unseen data. In essence, LOOCV is a valuable tool that aids in optimizing model parameters and enhancing the overall reliability of predictive models.