FP Draft

Group 11

B classifier

Description of the classification problem:

The classification problem revolves around categorizing data points into two classes, 'bn' and 'bo', based on certain features.

Description of proposed solution and its performance:

The proposed solution involves creating a simple classification function `classify_b(value)` that assigns a class label ('bn' or 'bo') to a given input value based on a predefined threshold (0.5 in this case). The function classifies values greater than 0.5 as 'bo' and values less than or equal to 0.5 as 'bn'. The performance of this solution depends on how well this threshold-based classification aligns with the actual distribution of data points in the dataset.

Comparative analysis between the proposed solution and other existing solutions in the market:

The provided solution is a basic threshold-based classifier, which may not be as sophisticated or accurate as other machine learning algorithms like logistic regression, decision trees, or neural networks. These algorithms can learn complex patterns in the data and potentially achieve higher classification accuracy than a simple threshold-based approach. After feedback, we adopted a more sophisticated model as well using Bayesian Classification (b classifier), Logistic Regression (C classifier with 500 files), O classifier (SVM).

Factors used in making the selection accuracy, customer service/tech support, and customer satisfaction:

The selection of a classification solution depends on various factors including accuracy, computational efficiency, interpretability, scalability, and ease of deployment. Customer service/tech support and customer satisfaction are crucial for ensuring that users can effectively utilize the solution and address any issues or concerns they encounter during its implementation.

Reflections on the experience of working with other groups:

Working with other groups can provide valuable insights and perspectives, but it can also present challenges, particularly when resolving issues or bugs in their tools. Effective communication and collaboration are essential for navigating these challenges and finding mutually beneficial solutions.

Insights into handling requests from other groups to enhance and debug your own code/tool:

We received feedback from other groups that we had to make our code more dynamic so that it could fit in other datasets. So, we worked accordingly.

Personal reflections on the classification problem:

The classification problem addressed in this code snippet is a simplified example, but similar challenges are encountered in various professional settings, such as fraud detection, sentiment analysis, and medical diagnosis. Insights gained from this project, such as the importance of feature selection, model evaluation, and understanding domain-specific nuances, can be applied in practical work scenarios to develop more robust and effective classification solutions.

C Classifier

Description of the classification problem:

The classification problem in this code involves categorizing data points into two classes, 'cb' and 'cw', based on the values provided. The threshold for classification seems to be set at 200, where values less than or equal to 200 are labeled as 'cb', and values greater than 200 are labeled as 'cw'.

Description of proposed solution and its performance:

The proposed solution consists of a simple classification function `classify_c(values_vector)` that takes a vector of values as input and classifies each value into either 'cb' or 'cw' based on the defined threshold. The function loops through the input values, applies the classification rule, and returns a vector containing the assigned class labels. The performance of this solution depends on how well the threshold of 200 discriminates between the two classes and whether it captures the underlying patterns in the data accurately.

Comparative analysis between the proposed solution and other existing solutions in the market:

The provided solution is a basic threshold-based classifier, which may lack the sophistication and predictive power of more advanced machine learning algorithms like logistic regression, decision trees, or support vector machines. These algorithms can learn complex patterns in the data and potentially achieve higher classification accuracy than a simple threshold-based approach. After feedback, we adopted a more sophisticated model as well using Bayesian Classification (b classifier), Logistic Regression (C classifier with 500 files), O classifier (SVM).

Factors used in making the selection accuracy, customer service/tech support, and customer satisfaction:

When selecting a classification solution, factors such as accuracy, computational efficiency, interpretability, scalability, and ease of deployment are considered. Additionally, customer service/tech support and customer satisfaction play vital roles in ensuring that users can effectively utilize the solution and address any issues or concerns they encounter during its implementation.

Reflections on the experience of working with other groups:

Collaborating with other groups can offer valuable insights and perspectives, but it can also present challenges, particularly when resolving issues or bugs in their tools. Effective communication, collaboration, and problem-solving skills are essential for navigating these challenges and finding mutually beneficial solutions.

Insights into handling requests from other groups to enhance and debug your own code/tool:

We received feedback from other groups that we had to make our code more dynamic so that it could fit in other datasets. So, we worked accordingly.

Personal reflections on the classification problem:

The classification problem addressed in this code snippet is straightforward, but similar challenges are encountered in various professional settings, such as fault detection, quality control, and anomaly detection. Insights gained from this project, such as the importance of threshold selection and understanding the underlying data distribution, can be applied in practical work scenarios to develop more robust and effective classification solutions.

O Classifier

Description of the classification problem:

The provided code addresses a classification problem where data points are categorized into different classes ('o1' to 'o9') based on the values of eight features.

Description of proposed solution and its performance:

The proposed solution involves a function `classify_c(values_vector)` that assigns a class label to a vector of input values based on predefined thresholds. Each feature has its own threshold range, and if the input values fall within a certain range for a particular feature, the corresponding class label is assigned. The performance of this solution depends on how well the defined threshold ranges capture the underlying patterns in the data and whether they accurately classify the input vectors.

Comparative analysis between the proposed solution and other existing solutions in the market:

The provided solution is a rule-based classifier that assigns class labels based on predefined thresholds for each feature. While this approach is simple and interpretable, it may not capture complex patterns in the data as effectively as more advanced machine learning algorithms like logistic regression, decision trees, or neural networks. These algorithms can learn from the data and potentially achieve higher classification accuracy. After feedback, we adopted a more sophisticated model as well using Bayesian Classification (b classifier), Logistic Regression (C classifier with 500 files), O classifier (SVM).

Factors used in making the selection accuracy, customer service/tech support, and customer satisfaction:

When selecting a classification solution, factors such as accuracy, computational efficiency, interpretability, scalability, and ease of deployment are considered. Additionally, customer service/tech support and customer satisfaction play vital roles in ensuring that users can effectively utilize the solution and address any issues or concerns they encounter during its implementation.

Reflections on the experience of working with other groups:

Collaborating with other groups can offer valuable insights and perspectives, but it can also present challenges, particularly when resolving issues or bugs in their tools. Effective communication, collaboration, and problem-solving skills are essential for navigating these challenges and finding mutually beneficial solutions.

Insights into handling requests from other groups to enhance and debug your own code/tool:

We received feedback from other groups that we had to make our code more dynamic so that it could fit in other datasets. So, we worked accordingly.

Personal reflections on the classification problem:

The classification problem addressed in this code snippet is rule-based and relies on predefined thresholds for classification. While this approach may be suitable for certain scenarios, it may not generalize well to complex datasets with nonlinear relationships between features and classes. Insights gained from this project can be applied in practical work scenarios to develop more robust and effective classification solutions, especially when dealing with structured data and predefined rules for categorization.