# 

# INNOVISTA

Zaryaab Khan

Niral Devang Shah

Zohair Shakir

Rakesh Reddy Sriyapu Reddy

Vamsi Sai Tumu

# The Problem

We are looking to enhance the Innovista classifier, a tool intended for user-friendly classification of input data through UI-based menus and algorithms. The primary goal is to streamline the classification process, making it accessible even to individuals with limited proficiency in R code. The classifier's current design focuses on simplicity, but there is a need to further optimize its user interface and algorithmic functionality. Additionally, the classifier should be evaluated for its effectiveness in handling diverse datasets, ensuring robust performance across various input scenarios. The problem extends to accommodating users with minimal R code knowledge, necessitating improvements in instructional elements and guidance within the classifier interface. Furthermore, the classifier's usability and efficiency should be assessed through rigorous testing procedures, with a specific emphasis on user experience and error handling. Overall, the problem statement revolves around refining the Innovista classifier to offer an intuitive and effective solution for users with limited R code expertise, addressing usability, algorithmic performance, and adaptability to diverse datasets.

# Our Solution

The classifier Innovista has developed seeks to simplify the user experience by utilizing. UI based menus and algorithms to determine how to classify user input and present the results. It is designed to be used by individuals with minimal knowledge of R code. The classifier offers the user the option to input a vector containing the inputs or manually type the input in the UI prompt. In both cases, if the input contains NAs or strings, the function will stop and issue and error message, listing the index of the NAs or strings, enabling the user to easily identify and rectify the issue.

The classifier must be trained on a dataset each time it is run. The user is given the option to select from a list. This list is generated based on matching file names in the current working directory. If the user wishes to use a dataset in a different directory, or one that is not listed, the user can select the ‘other’ option and input the file name manually. The classifier will then search for the file in the current directory, and if it is not found, prompt the user to change the directory temporarily. This search only fires once, so if the user fails to provide the correct file. name or directory, the function will fail.

The classifier determines the type of classification algorithm to use based on the structure of the training dataset. For single feature datasets (b and c type datasets) the classifier will use KNN to predict the class of the input value(s). For 8 feature datasets (o type datasets), the classifier will use multinomial logistic regression to predict the class of the input based on the 8 feature values provided.

Due to the size of certain results exceeding max print in R, we opted to automatically assign the results to a variable in the global environment. This variable is named ‘results.\*’ where \* = a number. The classifier will always select the highest \* and increase it by 1 each time a new result is generated i.e., if the user manually creates a variable named ‘results.7’, and the classifier is run for the first time, the variable ‘results.8’ will be created in the global environment.

If KNN is used to predict classes, and the input is greater than 1, then the function will provide a graph with the count of the predicted classes. If the user also provides a vector of ‘true values’ (for the sake of testing accuracy), the function will provide 3 graphs which contain the true and false predictions for each class, and a pie chart with the total accuracy.

# **I.** Introduction

1. Brief overview of the final report

**Introduction**

The Innovista Classifier is a user-friendly tool designed for individuals with minimal knowledge of R code. It simplifies the classification process by employing UI-based menus and algorithms to classify user input and present results. The classifier requires training on a dataset each time it runs, allowing users to choose from a list of available datasets or input a custom file name.

**User Input Options**

* Users can input a vector or manually type input in the UI prompt.
* Error handling for NAs or strings, providing clear error messages with index information for easy rectification.

**Dataset Selection**

* Users can select a training dataset from a list generated based on the matching file names in the current working directory.
* Option to input a custom file name and search for it in the current directory, also prompting the directory change if required.

**Classification Algorithm**

* For single feature datasets (b and c types), KNN is used for predicting classes.
* For eight feature datasets (o type), multinomial logistic regression is employed for prediction.

**Results Handling**

* Results automatically assigned to a variable in the global environment (e.g: ‘results.\*’),with the \* incrementing for each new result.
* This automatically manages the large result sizes exceeding max print in R

**Graphical Outputs**

* When KNN is used and input is greater than 1, the function provides a graph displaying the count of predicted classes.
* If a vector of ‘TRUE’ values is provided, three graphs are generated showing TRUE and FALSE predictions for each class, along with a pie chart indicating total accuracy.

**Error Handling**

* The function fails gracefully with a single search for the specified file, prompting the user to correct input errors in case the specified file does not exist in the specified working directory.

1. Importance of the classification problem

The classification plays an important and vital role in various segments of the market to categorize and assign predefined labels or classes to the data. This is useful in the following segmentations or classifications,

* Customer Segmentation
* Spam Filtering
* Fraud Detection
* Credit Scoring Classification

# **II**. Description of Classification Problem

1. Nature of the proposed solution

* User Friendly Interface
  + Designed for users with minimal R code knowledge.
  + Flexibility in Input – Allows users to select vectors or manual inputs in the UI prompts.
* Error Handling
  + Detects NA or strings in the inputs and provides clear error outputs.
* Dynamic Dataset Selection
  + Users choose from current directory datasets.
  + Option to input custom files names for data sets.
* Adaptive Algorithm Selection
  + Chooses the algorithm to choose from between KNN for single feature datasets and logistic regression for 8-feature datasets.
* Efficient Result Handling
  + Manages large result sizes and assigns it to a global variable.

1. Key challenges and complexities

* Data Quality
  + The classifier’s performance relies heavily on the quality of the training data. Inaccurate or noisy data may lead to an impact on predictions.
* Handling Missing Values
* Effect of outliers on the predictions

# **III**. Proposed Solution Explained

1. Overview of the proposed solution

The classifier aims to simplify the user experience by providing a user-friendly interface that utilizes UI-based menus and algorithms for classifying user input. It is designed for individuals with minimal knowledge of R code.

Features:

* UI based prompts and menus are used to interact with the classifier. This makes it accessible to the user with limited R knowledge.
* The solution offers package management functionality, which offers to load and install the missing packages automatically if they are missing to execute the code.
* Users can manually select the training dataset from the current working directory or input the file path name.
* Users are prompted to change the working directory if the selected file does not exist.
* The classifier uses the KNN classification algorithm for single-feature datasets (b and c classifiers), and for eight-featured datasets, the classifier makes use of logistic regression.
* Error messages are provided in case users are facing issues.

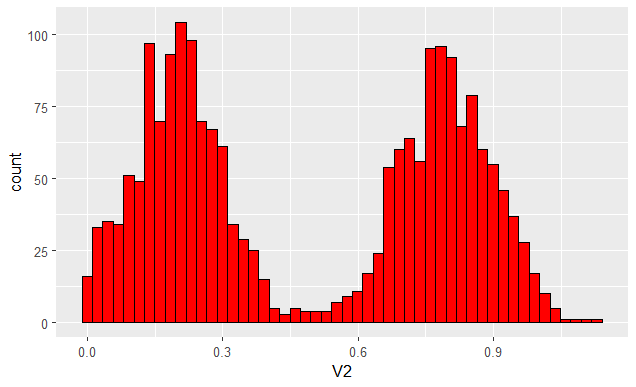
1. Methodology and algorithms used.

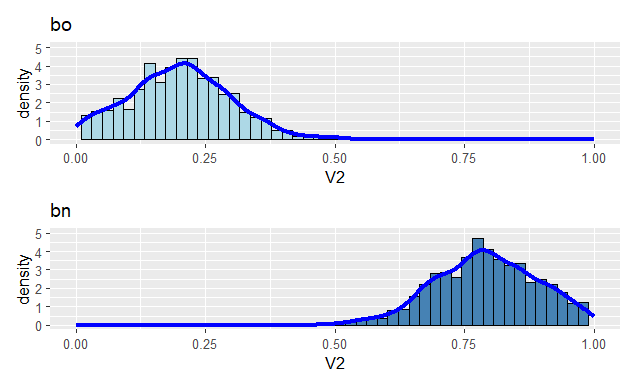
* KNN (K Nearest Neighbours) algorithm is used to classify single-feature datasets as in B and C classifiers.
* Logistic Regression is used to classify eight-segment datasets like the O classifier.

# **IV**. Results of Proposed Solution

## Innovista’s B Classifier

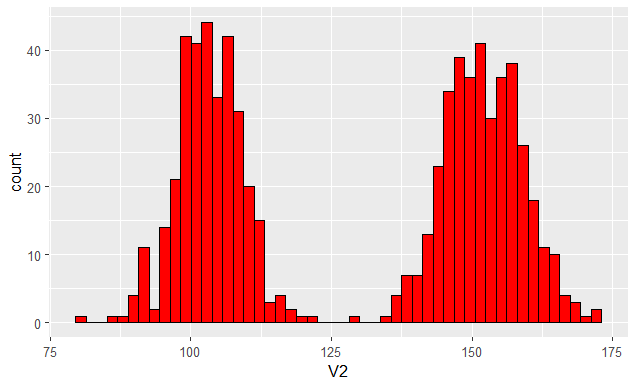
B1 Dataset Frequency Graph



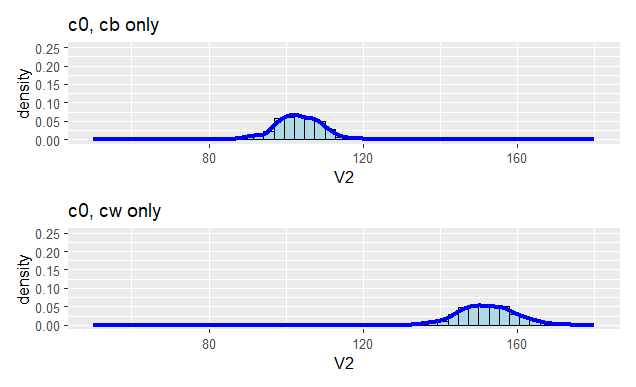
B1 Dataset Density graph separated by various classes.  
  


## Innovista’s C Classifier

C1 Dataset Frequency Graph



C1 Dataset Density graph separated by various classes.



# Accuracy Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| Company/Accuracy | B Classifier (%) | C Classifier (%) | O Classifier (%) |
| McKintel | 85 | 90 | 100 |
| Nebula | 99.8 | 99.7 | 100 |
| Project\_Classification | 87 | 98.5 | 97 |
| A3MOS | 80 | 100 | 100 |
| Warrier | 78 | 96 | 99 |

Looking into various companies offering the classification models, Nebula is most efficient in terms of usage and complexity of the program. Testing the accuracy for various classifiers, it was determined as below for Nebula,

# Nebula

## B Classifier

This classifier appears to use a method based on calculating the mean and difference between the input value and the mean of each plot. It then calculates the Euclidean distance and weight for each plot, deciding based on these values. The classification is determined by comparing values and means, and it includes a mechanism for handling overlapping values.

1. Data Preparation:

* Read the dataset ('dataset\_b2.csv').
* Calculate the mean for classes 'bo' and 'bn' separately.

1. Classification Algorithm:

* Compute the Euclidean distance for each class.
* Assign weights to each class based on distance.
* Decide on the predicted class using a threshold mechanism for handling overlapping values.

1. Testing:

* Illustrative test with a specific input value.

1. Accuracy

* Testing with inputs values resulted in a accuracy of 99.8%

## C Classifier

The C\_Classifier calculates the mean for two different classes (cw and cb) and then classifies new values based on their difference from these means. It iterates through a vector of values, comparing the differences, and assigns a class (cw or cb) to each value.

1. Data Preparation:

* Read the dataset ('c0.csv').
* Calculate the mean for classes 'cw' and 'cb' separately.

1. Classification Algorithm:

* Iterate through a vector of values, comparing differences with class means.
* Assign a class ('cw' or 'cb') based on the minimum difference.

1. Testing:

* Illustrative test with a vector of values.

1. Accuracy

* Testing the classifier with input vectors resulted an accuracy of 99.78%

## O\_Classifier

The O\_Classifier reads a dataset, renames columns, and defines a function (`classify\_c`) that calculates the Euclidean distance between an input vector and each row in the dataset. It then assigns the class of the row with the minimum distance as the predicted class.

1. Data Preparation:

* Read the dataset ('dataset\_o2.csv').
* Rename columns for clarity.

1. Classification Algorithm:

* Define a function (`classify\_c`) to calculate Euclidean distance.
* Iterate through rows to find the minimum distance and assign the corresponding class.

1. Testing:

* Illustrative test with a sample input vector.

1. Accuracy

* Testing the classifier with a sample input vector resulted an accuracy of 100%

## Nebula’s Classification Code



# Innovista vs. Other Companies

External solutions are designed to classify only a specific dataset, meaning the user may be required modify the code to ensure it works correctly.

However, Innovista solution differs in that it can adapt to any dataset, regardless of its name or directory. This feature makes it easier for the user to avoid errors and ensures the solution is more flexible and user-friendly. A single function determines the correct algorithm to use based on the structure of the training dataset chosen.

# Ranking Order for Other Companies

Below are the ranking orders for various companies based on overall customer experience, resolution, and promptness.