

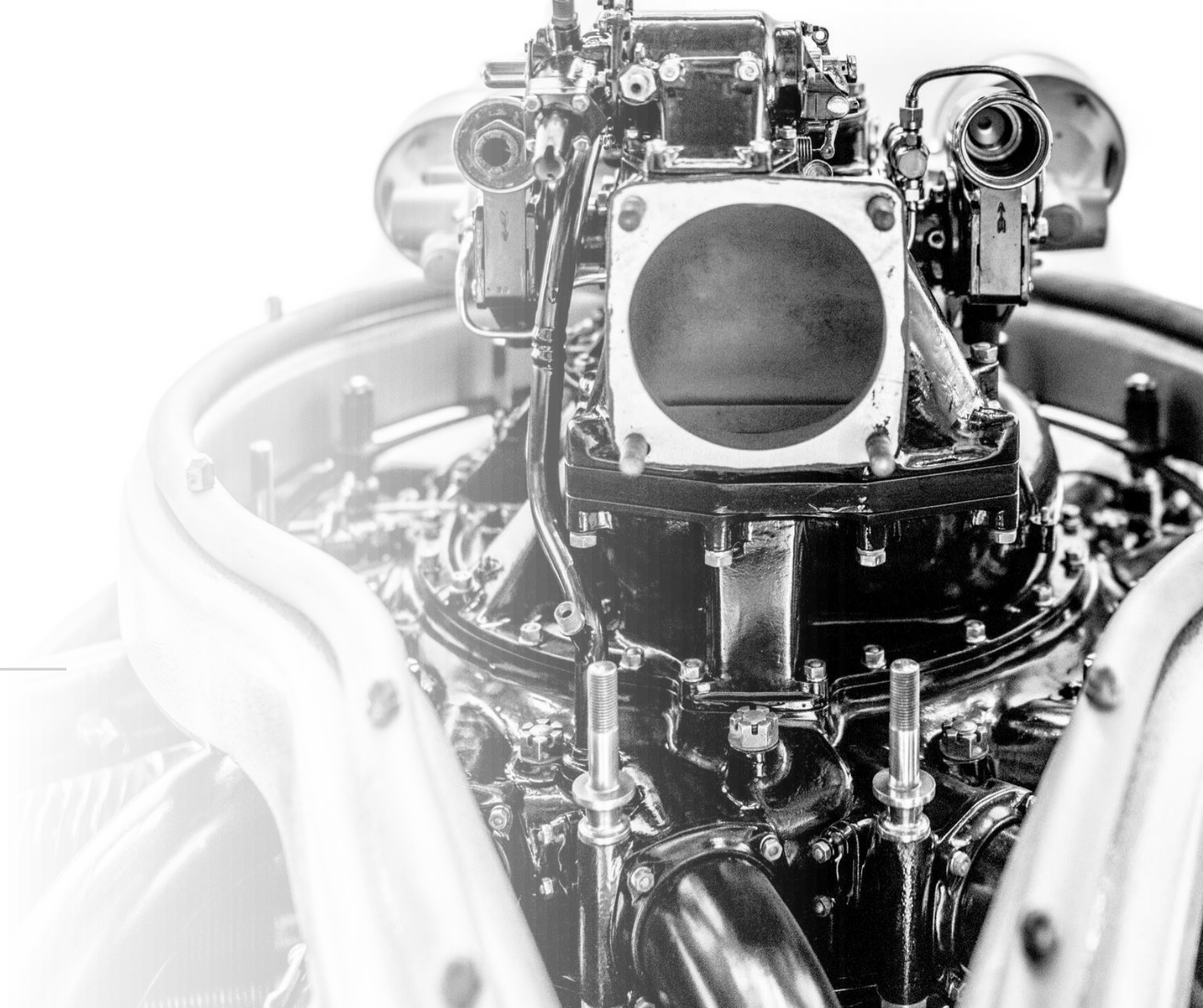


# INNOVISTA

---

Performance Analysis and  
Comparative Study

Group-2



# The Classification Problem

---

- There are 3 types of datasets that are given and the task is to predict classes based on the values we input
- B-type and C-type datasets have 2 classes and O-type has 10 classes
- For the B-type we should be able to input a number and get the correct class
- For the c-type we should be able to input a vector of any length and get the correct class for each element in the vector
- For O-type we need to input a vector that contains 8 numbers and get the correct class for the vector



# Parameters considered to find the Perfect solution

---

Our first phase of testing was based just upon the "Accuracy of the model"

---

Based on Accuracy alone we narrowed our option to Nebula and A3mos

---

We took into account parameters like ease of use, errors and editing required to use the code

---

Finally, after in-depth testing we found Nebula to be the most optimal solution, from the options available in the market.



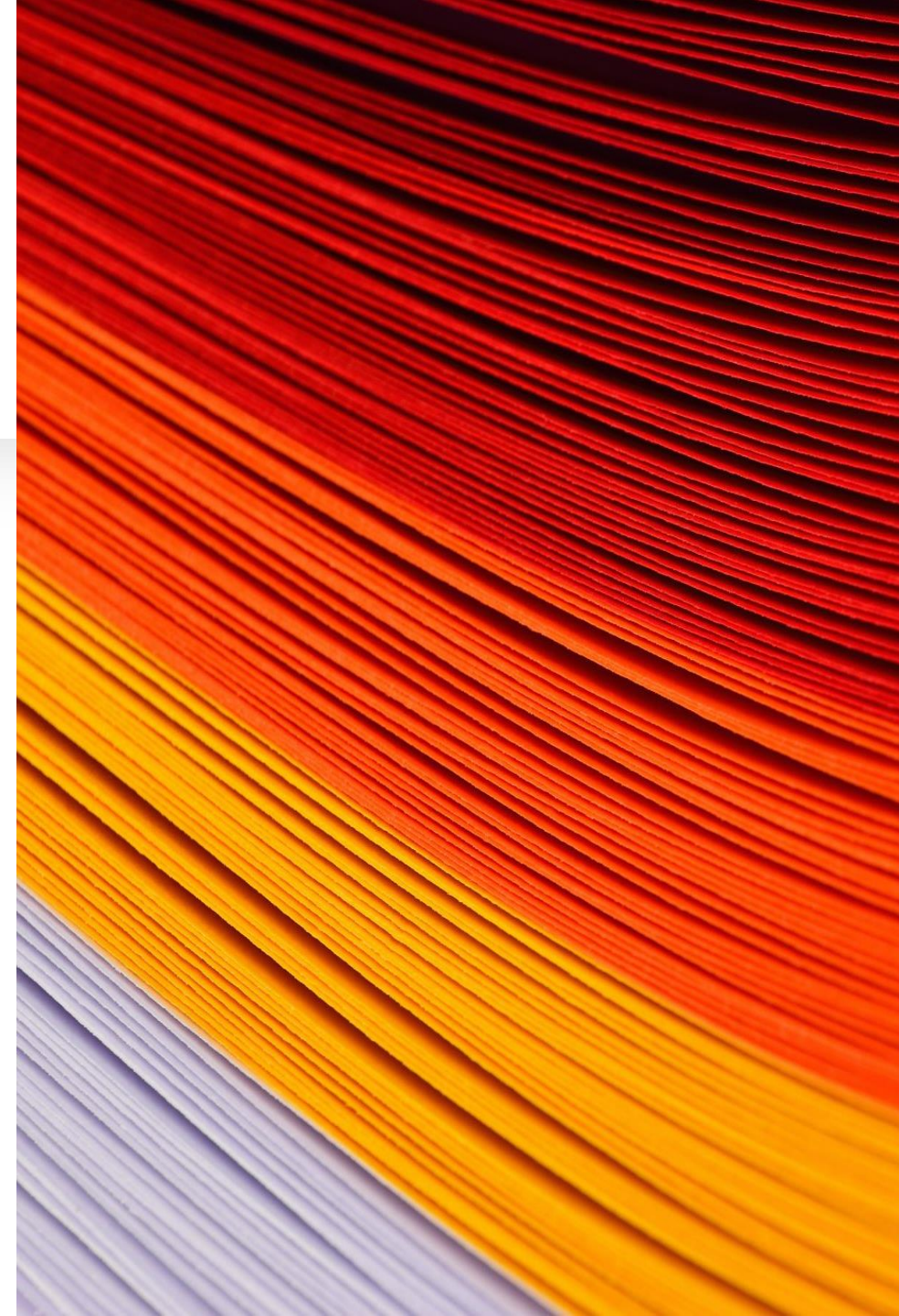
# The Nebula Classifier

## For B and C Type Classifiers

- The Nebula classifier works by using the Euclidean Distance between the mean of two groups and input values

## For O Classifier

- The classifier use a FOR loop to iterate over the rows and then they find the difference between the input value and current row. It keeps going until the difference is 0 and classifies it according the row it stops at



# Strengths

- High Accuracy of the model
- The Code was cleanly written
- Good Responsiveness

# Weaknesses

- The code is not user-friendly
- It requires user to have a strong understanding of R to operate
- The code didn't have any built-in features to test accuracy
- For O classifier the run time is too high





# Overview and Key Features of Innovista

- The Innovista classifier is a user-friendly tool that simplifies the process of classifying user input using UI-based menus and algorithms. It is designed for individuals with minimal R code knowledge.

01

UI-based menus for easy navigation and input

02

Automatic algorithm selection based on dataset structure

03

Error handling for NAs and strings in input

04

Automatic results storage in the global environment

# Setup and selection

The function is called without parameters or variable assignment. It checks for required packages and prompts for installation if necessary.

The classifier prompts the user to select a dataset from a list of matching files in the current directory. If the desired dataset is not listed, the user can select "other" and input the file name manually.

# Classifiers:

## B Type Classification

For B type datasets, the classifier uses KNN to predict the class of the input value(s). It provides a message with the results and stores them in a data frame in the global environment.

## C Type Classification

For C type datasets, the classifier uses KNN to predict the class of the input value(s). It provides a message with the results, stores them in a data frame in the global environment, and creates graphs.

## O Type Classification

For O type datasets, the classifier uses multinomial logistic regression to predict the class of the input based on the 8 feature values provided. It provides a message with the results and stores them in the global environment.

## Error Handling

The classifier checks for NAs and strings in the input and issues an error message if any are found, listing the index of the error. This allows the user to easily identify and rectify the issue.



## K-Nearest Neighbors (KNN) in Innovista Classifier

KNN is like your classifier asking similar inputs for advice on classifying a new input.



### Content:

R- code uses KNN to predict a class based on the majority class of its nearest neighbors.

If most similar inputs have a certain class, your classifier predicts that class for the new input.



### Real-Life Analogy:

Imagine you're deciding what movie to watch, and you ask your friends who have similar tastes. If most of them like action movies, you'd likely enjoy an action movie too.



### Application in Innovista:

KNN is utilized when deciding the class of an input based on its similarity to others in the dataset.

# Multinomial Logistic Regression in Innovista Classifier

Multinomial Logistic Regression is like assigning weights to different factors (features) to predict the class of an input.



## Content:

code applies Multinomial Logistic Regression for datasets with eight features to predict the class based on these features.



## Real-Life Analogy:

Consider predicting the genre of a movie based on various factors like its rating, actors, and director. If highly-rated movies often fall into the drama genre, a new highly-rated movie is likely a drama.



## Application in Innovista:

Multinomial Logistic Regression is used when predicting the class of an input based on its features in the eight-feature datasets.

# Comparison Among different companies:

B-classifier

## Team 4 (Nebula):

- Utilizes a weighted approach with Euclidean distance, providing a nuanced classification.
- Clarity and readability are good.
- Addresses overlap cases.

## Team 1 (Mckintel):

- Straightforward threshold-based classification.
- Clarity and readability are decent.
- Lacks complexity compared to Team 2.

## Team 5 (Warriors):

- Uses a simple for loop for classification.
- Clarity is moderate.

## Teams 6 (a3mos) and 3 (Project\_Classification):

- Similar approach to Team 3.
- Basic threshold-based classification.
- Clarity is decent.



# C-classifier

## Team 1 (Mckintel):

Utilizes two files for classification based on different criteria.

Clarity and readability are good.

## Teams 4 (Nebula), 5 (Warriors)

Both teams use a straightforward mean-based classification.

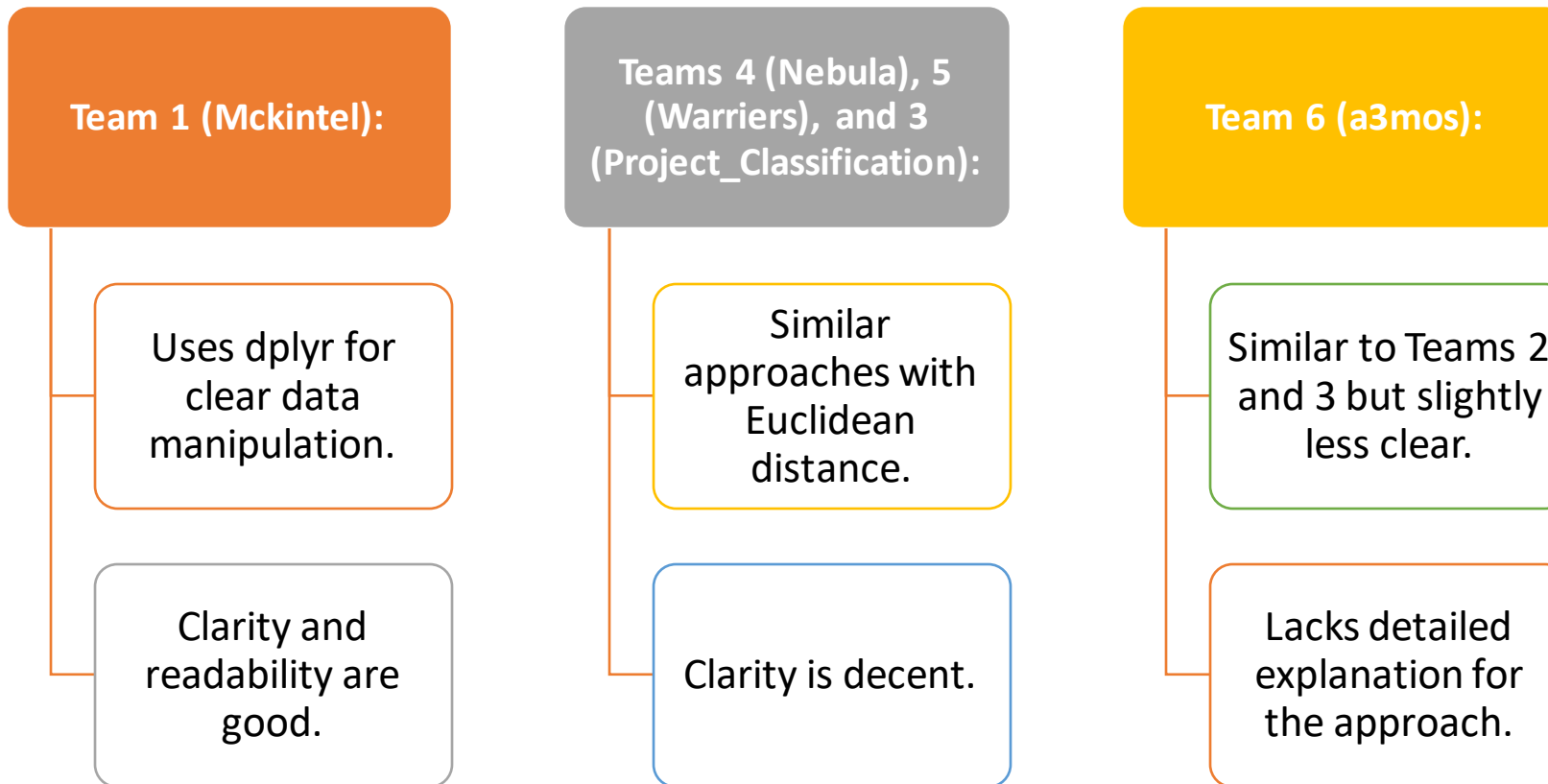
Clarity is decent.

## Teams 6 (a3mos) and 3 (Project\_Classification):

Similar to Teams 2 and 3 but slightly less clear.

Lacks detailed explanation for the approach.

# O-classifier



# Accuracy comparison for different companies

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

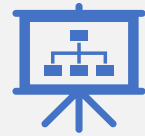


# Overall Rankings



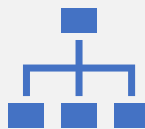
## **Team 4 (Nebula):**

Innovative approach in the B Classifier.  
Clarity and efficiency are commendable.



## **Team 1 (Mckintel):**

Good overall performance in all classifiers.  
Clarity and readability are strong.



## **Teams 5 (Warriors), 6 (a3mos), and 3 (Project\_Classification):**

Similar performance with minor variations.  
Adequate clarity but lacking some detail.

# Recommendations:

- Teams might consider error handling and informative messages for better usability.
- Consideration of edge cases, such as when a class cannot be determined, may enhance robustness.
- Teams may want to optimize their code for efficiency and clarity.





# Thank you

---