**Category Classification**

Brieuc Lunven, Yuyu Chebon Ndiema, Albin Poulose,

Hiral Yadav and Ling Zhang

University of Texas at Dallas

BUAN 6333: Foundations of Programming for Business Analytics

Professor Waseem G. Shadid

Project- Group 3

Table of Contents

[Problem Statement 2](#_Toc151982126)

[Classification Algorithms 2](#_Toc151982127)

[Category Identification 2](#_Toc151982128)

[Data Analysis 3](#_Toc151982129)

[B Classifier 3](#_Toc151982130)

[C Classifier 3](#_Toc151982131)

[O Classifier 4](#_Toc151982132)

[Cross Validation 5](#_Toc151982133)

[Model Selection 7](#_Toc151982134)

[Project Evaluation 9](#_Toc151982135)

[B Classifier 9](#_Toc151982136)

[C Classifier 9](#_Toc151982137)

[O Classifier 9](#_Toc151982138)

[Other Team Evaluation 9](#_Toc151982139)

[Groupe 1 - Mckintel 9](#_Toc151982140)

[Groupe 2 – Innovista 10](#_Toc151982141)

[Groupe 4 - Nebula 11](#_Toc151982142)

[Groupe 5 - Warriers 11](#_Toc151982143)

[Groupe 6 - A3MOS 12](#_Toc151982144)

***Standout Performers…………..……………………………………………………………………………………………….13***

[Learning Experience 13](#_Toc151982145)

[Conclusion 13](#_Toc151982146)

[References 13](#_Toc151982147)

# Statement

The project entails developing a dynamic simulation that replicates a real-world scenario where a client solicits proposals from various companies. These proposals aim to create tools for data analysis. The classification tasks involve assigning appropriate labels to both individual numeric values and vectors of numeric data

1. B\_Classifier:

The objective of this function was to take a single numeric input and then output the class type of that value.

* Classification problem: should decides whether a single numeric value is more like a generic number ('bn') or something else ('bo').
* input: The numeric value is sourced from the file dataset\_bx.csv.
* **Output:** A string representing the class type, where the value can be either 'bn' or 'bo' based on the characteristics of the input numeric value.

1. C\_Classifier:

This function accepted a vector of numeric values as input and returned a vector of strings corresponding to the class type of the respective input value.

* Classification problem : The goal is to categorize each numeric value in a given vector into one of two classes, denoted as 'cb' or 'cw'.
* Input: This classifier operates on a single vector containing numeric values sourced from the file cx.csv.
* Output: The result is a vector of strings, mirroring the size of the input vector, where each string signifies the class type ('cb' or 'cw') corresponding to the respective numeric value.

1. O\_Classifier:

This function was supposed to take 8 numeric value vectors as input and outputs a single string representing the class type.

* Classification problem : With a set of eight numeric values representing features, this function should determine an overall class type ('o0' to 'o9') for the entire set.
* Input: The input vector is derived from the file dataset\_ox.csv, and it encompasses eight numeric values that represent distinct features associated with various classes.
* Output: The output is a single string, indicating the class type. Possible output values range from 'o0' to 'o9', each representing specific characteristics inferred from the input vector.

# Classification Algorithms

Classification is a Machine Learning supervised model that predicts outcome based on different features seeded in the model. Classification models are useful to predict against set of predefined categories unlike regression models that predict continuous dependent variables. As we must classify a category from a predefined vector, this supervised ML algorithm suits best for our problem statement.

Going further we will deep dive into different classification model evaluation statics and conclusion to draw for the same.

There are a plenty of classification models to choose between based on the problem statement. Some well-known data models that fit well in most of the data nature are –

* Linear Regression
* Logistic Regression
* Ridge & Lasso Regression
* SVM
* Random Forest
* Xgplot
* Naïve Bayes

# Category Identification

1. Linear Regression or Ridge & Lasso Regression

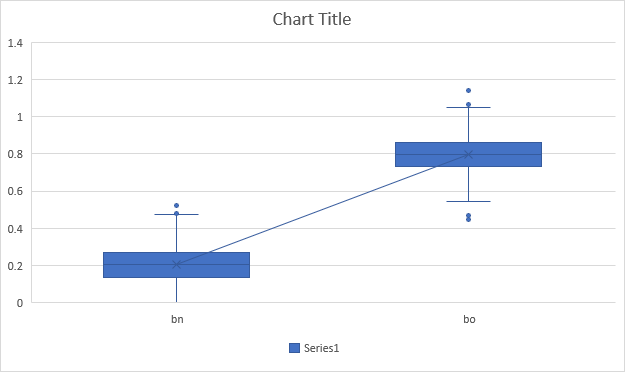
To conclude a best-fit data model, data analysis and high-level prediction of result is a crucial task in any classification problem. For this purpose, below are our findings and study to predict the best-suited data model of interest.

# Data Analysis

Data analysis is an important step in the process of predicting data model. It gives a visualization of data overlapping between different features and based on the linearity of the plot a model can be concluded. To understand the data characteristics, we have used different visualization mechanisms like scatter plot, box plot, histograms in Excel and python.

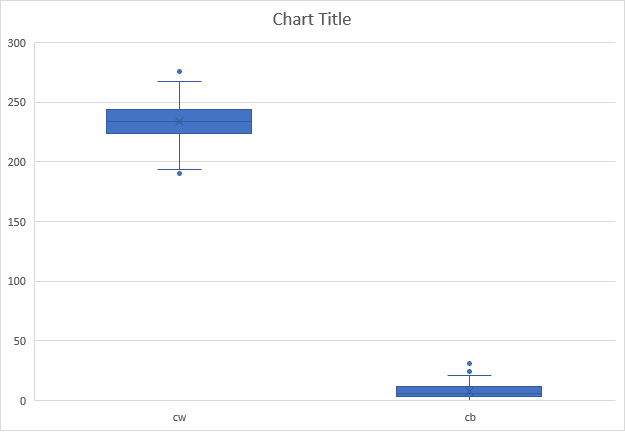
## B Classifier

Based on the dataset analyzed, the mean for both categories falls in different ranges hence there is no data overlapping and data can be distinguished using decision boundary. Hence the problem can be applied to any of the category classification algorithms like logistic regression, KNN (Kth Nearest Neighbor), or SVM.



## C Classifier

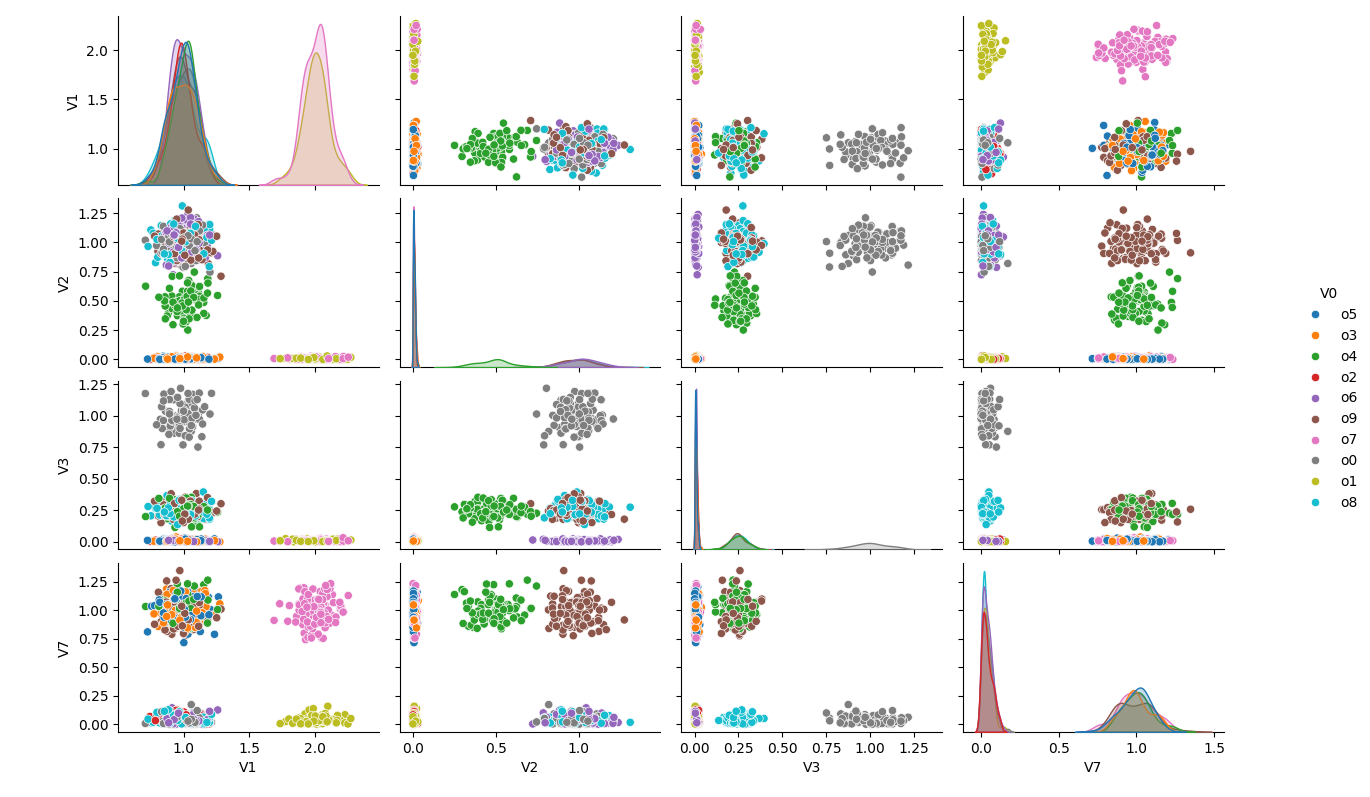
Like the B-dataset, there is a clear distinction between both categories (*cb* and *cw*) and a decision boundary can be drawn to distinguish both. Hence any category classification models like gradient decent / KNN / SVM are good to use.



## O Classifier

Based on the criticality of the feature, here we are finding the correlation between (V1, V2, V3, V7) features and are distinguished based on category.

From graphs, it is clearly visible that there is a lot of overlap between data points and data cannot be bifurcated in a regression line, hence linear regression model will not fit well. Also, linear regression does not suit for large number of features, as we are predicting category based on 8 features.



Sample code script – Below is the sample script used to come up with pairplot between different features.

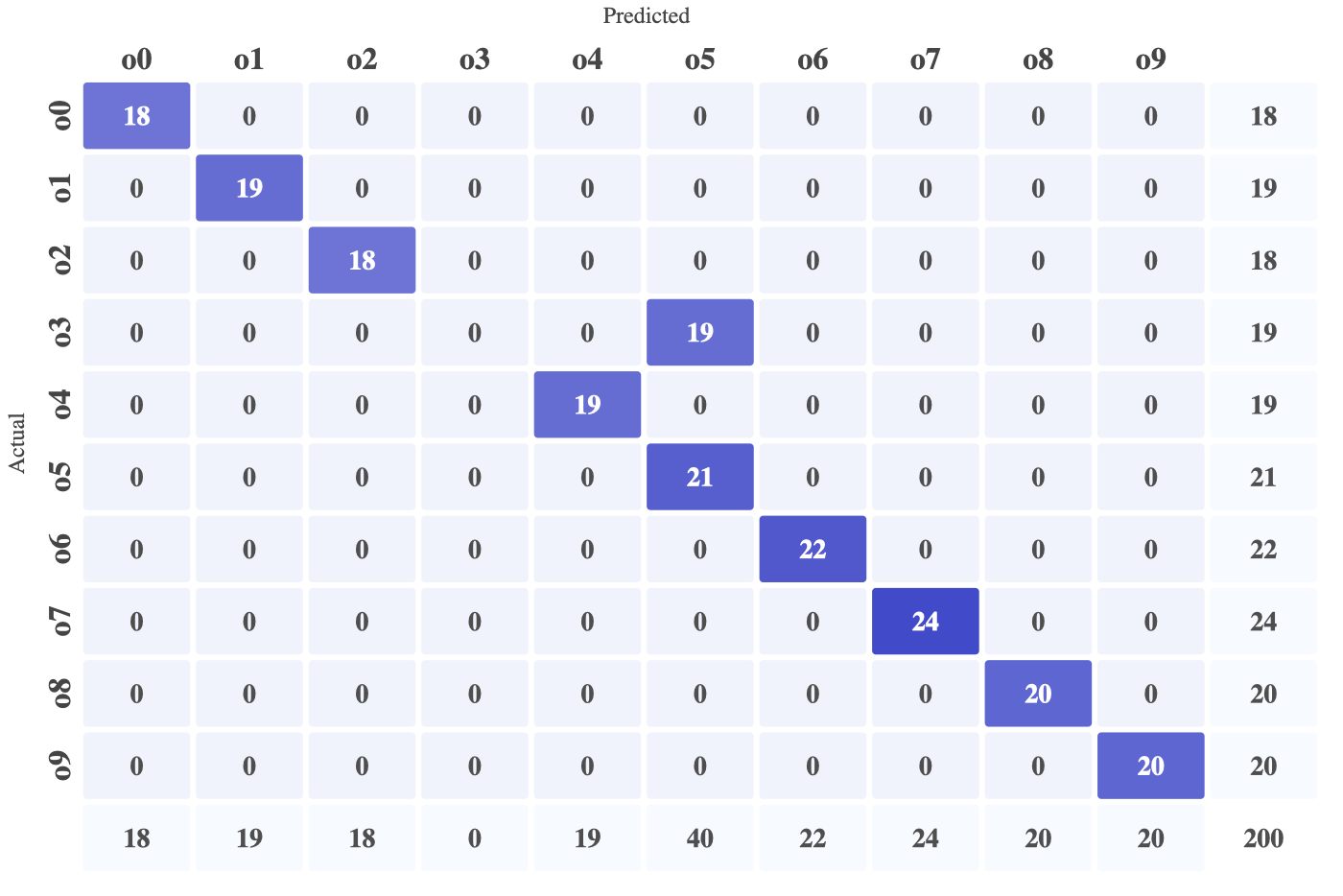
|  |
| --- |
| import seaborn as seaborn import pandas as pd import matplotlib.pyplot as plt  dataset\_o = pd.read\_csv('dataset\_o6.csv') dataset\_o = dataset\_o[['V0', 'V1', 'V2', 'V3', 'V7']] seaborn.pairplot(data = dataset\_o, hue='V0', height=1.75)  plt.show() |

### Cross Validation

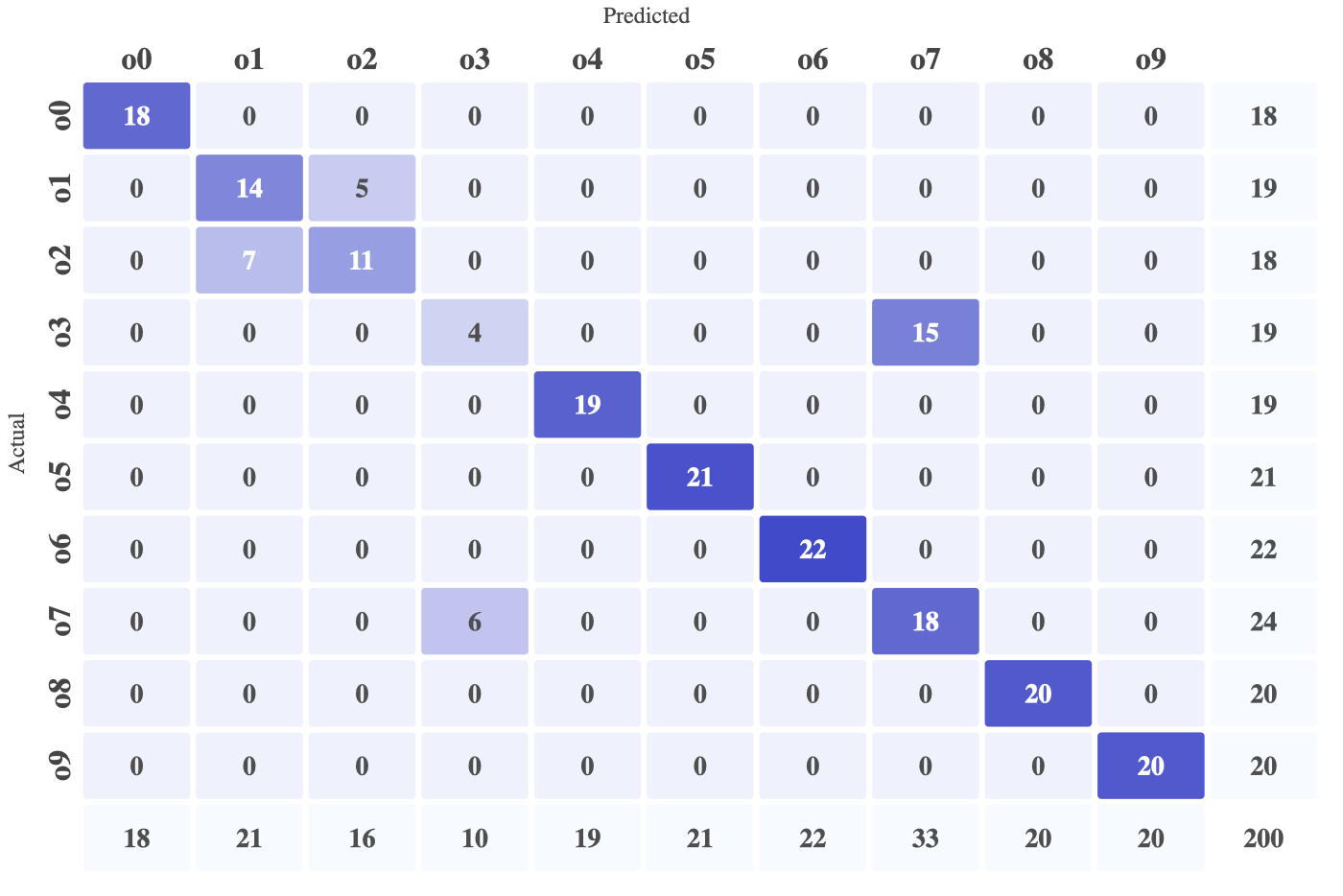
Cross Validation is a technique that compares output of different models, it gives average probability of getting a dependent variable based on different features in training and test datasets. This is an important step to get a rough idea about which model will work best on given dataset and will be most accurate.

In out classification problem, based on the data characteristic, we have compared Ridge and Lasso and Random Forest Classification models to get rough probability of the model to go. Find below statistics for reference –

Ridge and Lasso Classification Probability Model

**

*Random Forest Classification Probability Model*



### Model Selection

Based on the statistics drawn above, we have come up with an ***Elastic Net Classifier (Ridge & Lasso) Classification Model*** *for O-Classifier*. Based on our study, this model fits well due to –

* It supports many features well and we have 8 features to support.
* It works on L1 and L2 regulation strategy that reduces data overfitting and helps in predicting the most accurate result.

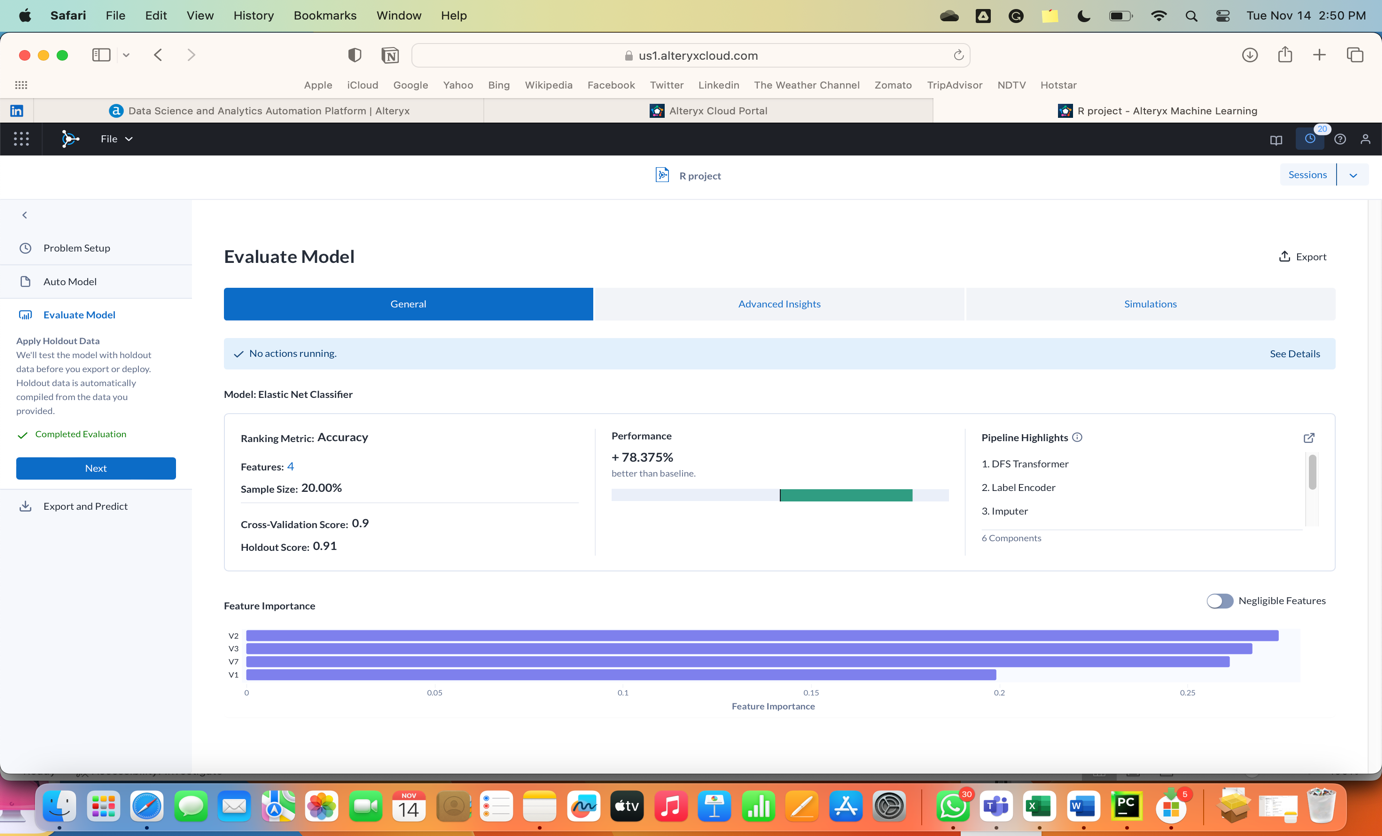
|  |
| --- |
| Models: Ridge, Lasso, ElasticNet  Number of resamples: 50  Accuracy  Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  Ridge 0.98058252427184466882 0.99021987435751002149 1 0.99702103231183070164 1 1 0  Lasso 0.98979591836734692745 1.00000000000000000000 1 0.99900182013399418945 1 1 0  ElasticNet 0.98989898989898994497 1.00000000000000000000 1 0.99880780313325445707 1 1 0  Kappa  Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  Ridge 0.97841123454202472143 0.98911932937920576947 1 0.99668643626680430980 1 1 0  Lasso 0.98865083960625366188 1.00000000000000000000 1 0.99888992496001882682 1 1 0  ElasticNet 0.98876532001815709005 1.00000000000000000000 1 0.99867393027453088550 1 1 0 |

Optimum value of alpha and lambda for Ridge Algorithm -

A graph with numbers and lines

Description automatically generated

Below is the accuracy of the model for test data evaluation on Alteryx -

 ==

# Project Evaluation

## B Classifier

|  |  |  |
| --- | --- | --- |
| Test Data | Expected Category | Actual Category |
| 0.1731428710029030 | bn | bn |
| 0.637474603861165 | b0 | b0 |
| 0.609800736246051 | b0 | b0 |
| 0.290342113234898 | bn | bn |

## C Classifier

|  |  |  |  |
| --- | --- | --- | --- |
| Test Data | File | Expected Category | Actual Category |
| (239.56798589879700, 3.22170221106961200,  212.76518053622200) | C0.csv | (cw,cb,cb) | (cw,cb,cb) |
| (149.95345769864, 225.594876834637) | C28.csv | (cb,cw) | (cb,cw) |

## O Classifier

|  |  |  |
| --- | --- | --- |
| Test Data | Expected Category | Actual Category |
| (0.96649628,2.013170766,0.383186652,0.102118084,1.013104189,0.112836605,1.098008553,0.043559417) | o9 | o9 |
| (0.16649628,1.013170766,0.783186652,0.34158084,1.913104189,0.452836605,1.118008553,0.657559417) | o3 | o3 |

# Other Teams Evaluation

## Groupe 1 **- Mckintel**

|  |  |
| --- | --- |
| ***In the Classify\_b:***  Suggesting the code needs to be more optimized . When we enter Specific value from dataset it shows 'bn' instead of 'bo'.  Please check on that. For eg:   |  | | --- | | 0.5036308189240626 |   ***In the Classify\_c:***  Function is working. But seems like it takes only data file and may show error with reading other datasets.  ***In the Classify\_o:***  Function is not working, showing error  **Error in `tbl\_at\_vars()`:**! Can't subset columns that don't exist.  ✖ Column `V3` doesn't exist.  Run `rlang::last\_trace()` to see where the error occurred.  Please help as to what can be done with this? |

## Groupe 2 **– Innovista**

|  |
| --- |
| Compilation Issues –   * Code was not compiling on first instance. * Classifier was failing for evaluation.   ***In the Classify\_b & c :***  Getting incorrect classification for below values  A screenshot of a computer  Description automatically generated  A screenshot of a computer  Description automatically generated  ***In the Classify\_o:*** I did not find any incorrect predictions yet as training data is distributed like that. But you opted for logarithmic linear regression, which is more suitable for binomial classification and has fewer independent variables, however here we are predicting against 8 different factors (V1 – V8). It may cause incorrect results if training data changes, however, when I compared the model in the tool, logarithmic classification is giving good accuracy for the training data we seed. |

## Groupe 4 **- Nebula**

|  |
| --- |
| ***In the Classify\_b:*** Code is working smoothly.  ***In the Classify\_c:*** Code is working smoothly.  ***In the Classify\_o:*** Code is working smoothly. |

## Groupe 5 **- Warriers**

|  |
| --- |
| ***In the Classify\_b / In the Classify\_c:***  I think the syntax is a little bit confusing as you are using “c0.csv” directly and not considering other options. I think I understand your process, but I would say it is likely to be inexact for other values.  The function works but at the end you don’t provide the result so the user can’t get access to the class like asked in the group project.  Might have been more clear if you had implemented for each function:    ***In the Classify\_o:***  ***Une image contenant texte, Police, blanc, algèbre  Description générée automatiquement***  Personally, I would have test with the different values (“o0”,”o1” and so on) to prove the efficiency of the function. |

## Groupe 6 **- A3MOS**

|  |
| --- |
| ***In the Classify\_b:***  No errors detected and a fluid process. No changes to recommend.  ***In the Classify\_c:***Function doesn’t work as R is not able to read the following:  Une image contenant texte, Police  Description générée automatiquement  Impossibility to continue the code after that.  Moreover,it seems that sometimes R struggles to open your file properly and I had to do it again several times to make sure it was reading the “cX.csv”.  ***In the Classify\_o:***  The code is working smoothly.  No specific recommendations to suggest. |

Based our analysis on how many test values we run on each exercise and how many of them came correct as accuracy. Thus, here are our results:

* Nebula
* A3MOS
* Innovista
* Warrier
* Mckintel

# Standout Performers

Nebula stood out for their exceptional accuracy in categorizing data, demonstrating precision and attention to detail.

Mckintel earned high praise for customer service, responding promptly to inquiries with solutions that solved problems precisely and inspirationally. Their dedication established them as a top performer.

Innovista impressed with their quick responsiveness, addressing client needs rapidly to guarantee a positive experience from start to finish. Their commitment never wavered.

A3MOS gained admiration for the simplicity and accessibility of their solution, enabling smooth implementation through user-friendly design. They made complexity look effortless.

Warrier minimized errors impressively, delivering reliable outcomes that positioned them as a preferred choice when accuracy is paramount. They set the standard for precision.

# Learning Experience

**Working with other groups** was both a complex and enriching experience.

Thanks to other projects and recommendations, it brought diverse perspectives and ideas to the table. It led to more creative problem-solving solutions on our own code.

Moreover, it also allowed us to exchange skills and knowledge with other teams. Therefore, the peer reviews and collaborations contributed to an improvement on our code quality. Indeed, having multiple eyes on the code helped spot unseen errors, enhanced readability and enforced best practices.

However, sometimes differences in terminology, coding styles, or understanding of the issue arose and led to misunderstanding and a decrease of productivity.

Thus, here are some **insights** highlighting the multiple points explained above:

* The group A3MOS forgot to mention that the working directory must be set by the team evaluating the code in C\_Classifier. Therefore, without this indication the code didn’t work whereas if they provided us this information from the beginning the code would have run smoothly. After a few misunderstandings and several mail exchanges, the problem was resolved.
* Another problem encountered was that most groups didn’t include screenshots of our code when they reviewed it and therefore grasping the exact issues, they mentioned was harder than it should have been.

# Conclusion

In conclusion, our group embarked on a comprehensive exploration and analysis using R code. Through collaborative efforts, each team member contributed their expertise, skills, and insights to produce a robust solution.

The process of evaluation other groups codes not only allowed us to critically assess various approaches but also provided valuable data for our own code. The diversity in coding styles and problem-solving strategies encountered during this process broadened our understanding of how R can be applied.

The achievements of having three functions which give an answer to the problem encountered demonstrate our ability to apply statistical and data analysis techniques to real-world problems. Therefore, we should be able to implement the skills honed during this project in professional settings in the future.

# References

* Codebase - <https://drive.google.com/drive/folders/1n3b2UWJpR_5jk_ynJhy-w03-QrlmocOy>
* <https://www.youtube.com/watch?v=9lRv01HDU0s>
* <https://www.youtube.com/watch?v=38SUUaMX5Rg&t=5s>

A computer screen with arrows pointing to the screen

Description automatically generated