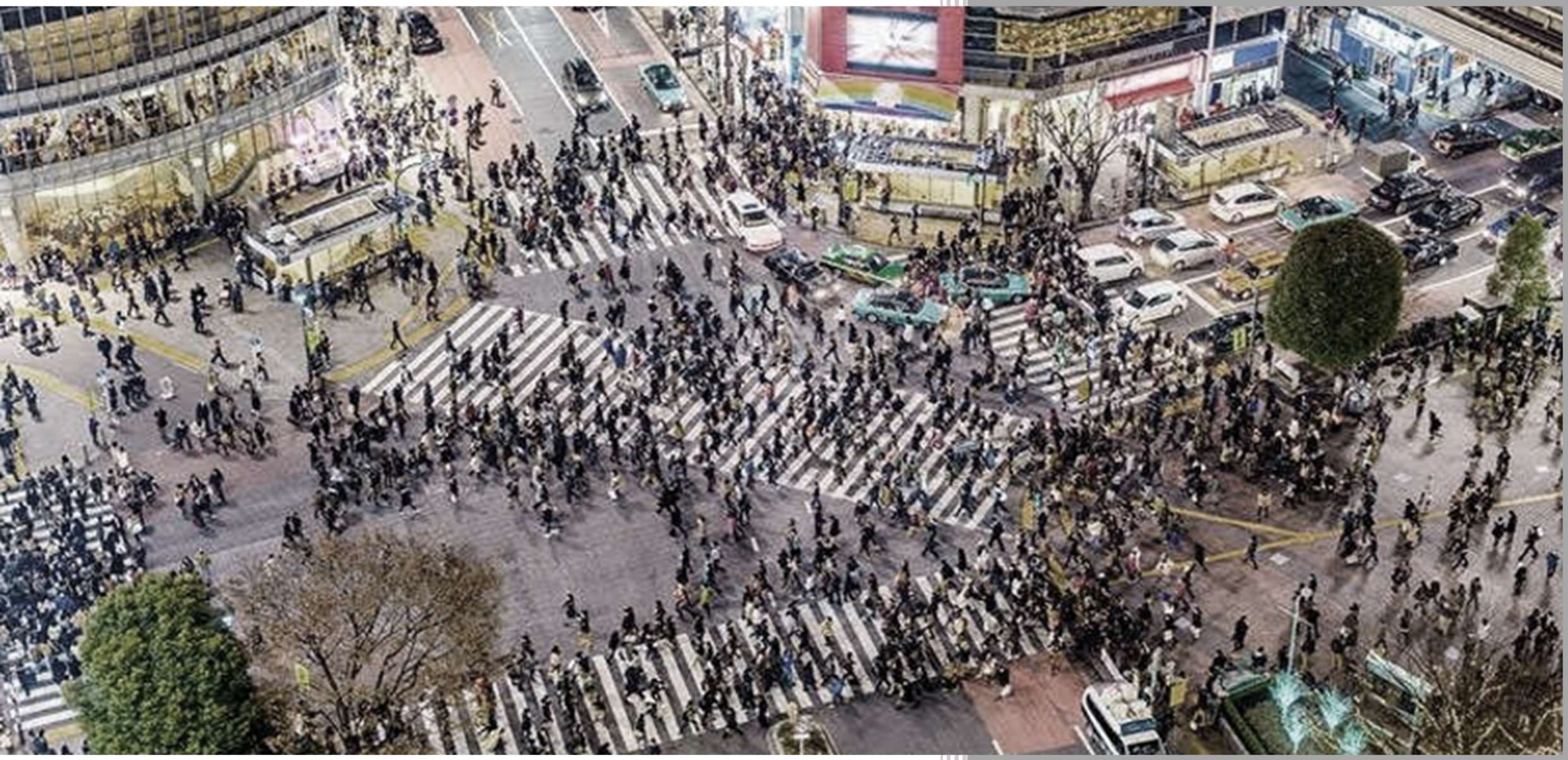
Data Mining II

BUDGETING FOR THE FUTURE: RESIDENT LIFESTYLE EXPLORATION TO EMPOWER CITY COUNCILS



Daniel Santos – 20230299
Mariana Pires – 20230330
Mariana Ribeiro – 20230303
Nuno Neves – 20230413
Vanda dos Santos Paiva - 20230337

NOVA IMS – Information Management School

2023/2024

INDEX

ABSTRAT

This research delves into the utilization of predictive modelling to enhance participatory budgeting within Mining City. The study is situated in the context of the growing necessity for data-driven decision-making in the allocation of public resources, with the objective of enhancing efficiency and inclusivity. The central hypothesis posits that the fusion of predictive analytics with citizen input can streamline budgeting procedures, ultimately resulting in a more efficient and fair distribution of resources.

In order to examine this hypothesis, a predictive model was created utilizing historical budgeting data, demographic details, and feedback from citizens gathered through surveys and public discussions. Various machine learning methods, such as linear regression and decision trees, were employed to detect patterns and anticipate future budgeting requirements based on community priorities.

The outcomes illustrated that the predictive model could reliably predict budget allocations that corresponded to citizen preferences, showing a notable correlation between projected and actual results. The model effectively pinpointed crucial areas of public concern, including infrastructure, healthcare, and education, enabling more focused and responsive budgeting choices.

Based on these findings, it is concluded that predictive modelling can serve as a beneficial tool in participatory budgeting, heightening the precision and significance of budget allocations. By integrating real-time data and ongoing citizen feedback, the model can be refined further to accommodate evolving community needs. This strategy not only enhances the efficiency of resource distribution but also nurtures increased transparency and trust between citizens and local authorities. Subsequent efforts will concentrate on integrating a broader range of data sources, exploring advanced modelling strategies, and designing user-friendly interfaces to engage citizens more effectively in the budgeting process.

INTRODUCTION

Participatory budgeting is a crucial strategy in municipal governance, addressing the challenges of citizen engagement and resource allocation. As a democratic approach to decision-making, it empowers community members to actively shape how public funds are distributed, promoting transparency and accountability in local government structures. By involving residents in the budgeting process, participatory budgets ensure that public resources align with community needs and priorities, while also fostering a sense of ownership and trust in governmental institutions. To accurately reflect the needs and aspirations of all citizens, a city council must thoroughly understand its population before proposing participatory budgeting.

The city council of Mining City is concerned about the effectiveness of its participatory budgeting process. As a result, the council members have decided to collect extensive data and conduct surveys to gain valuable insights into the demographics and lifestyles of their residents.

This effort identified key segments such as 'Travel Enthusiast', 'Health-Conscious', 'Adventure Seeker’, 'Fitness Enthusiast’, and 'Investor’. The council plans to use this information to make better decisions regarding fund allocation. Our main goal is to develop a predictive model that will allow the government to identify where to allocate the participatory budgeting.

Doing some research on participatory budgeting, it highlights the importance of demographic analysis and predictive modelling to improve the effectiveness of the processes. The studies in Polish cities using the Participatory Budgeting Library show that different voting formats and demographic factors significantly influence which projects are funded. This research suggests that understanding the specific needs and voting patterns of residents can lead to better fund allocation. Additionally, another research underscores that participatory democracy initiatives, can enhance civic engagement and trust in government, particularly when the process is inclusive and transparent (Oxford Academic). For example, the Scottish Government’s participatory budgeting initiatives have demonstrated the value of tailoring the budgeting process to the community's needs through detailed demographic insights.

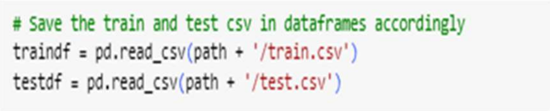
Based on these findings, Mining City’s approach to collecting demographic and lifestyle data, and identifying key segments such as 'Travel Enthusiast' and 'Health-Conscious', is expected to improve their process. By using predictive modelling with our help, the city can anticipate which projects will gain the most support, leading to a more efficient and equitable allocation of funds, and increased community engagement and satisfaction.

1. DATA EXPLORATION AND PREPROCESSING

To successfully execute our data mining project, there is several crucial initial steps that were undertaken:

1) **Importing Libraries and Packages –** the first step was to import the necessary libraries and packages, in order to allow us to run both simple and complex code, including hyperparameter tuning. So, we could handle data manipulation, visualization, preprocessing and model building.

2) **Importing csv Files with Datasets -** The next step was importing the csv files. For that we created a shared dedicated e-mail, where we organized a folder to store all the CSV files, with that approach we ensured that everyone on the team could run the same code and Google Collab could access the files through a consistent path.
3) **Defining Data Frames –** After importing the csv files, we defined two main data frames: *"traindf"* and *"testdf".* By creating these data frames, we ensured that our analyses and operations weren't performed in the original csv files.



```
# Save the train and test csv in dataframes accordingly
traindf = pd.read_csv(path + '/train.csv')
testdf = pd.read_csv(path + '/test.csv')
```

1.1. Business Understanding

1.1.1. Business Objectives

In the world of business, setting clear objectives is essential for achieving effective citizen engagement and efficient resource allocation. These objectives guide city councils in their efforts to foster transparency and community trust. This report aims to enhance the participation of the budgeting process by leveraging data and predictive modelling. For that we defined three main goals:

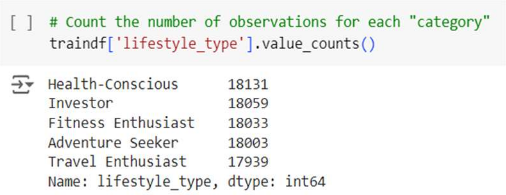
1) **Achieving more accurate resource allocation –** The city council of Mining City is concerned about the effectiveness of its participation on the budgeting process. So, the council members have decided to collect extensive data and conduct surveys to gain valuable insights. Therefore, by developing a predictive model that incorporates this data, the council aims to ensure that public funds are allocated in an equitable way.
2) **Promoting Citizen Engagement and Trust –** One of the core goals of participatory getting is to involve community members, and for that is key the identification of the types of population, such as "Travel Enthusiast", "Health Conscious", "Investor", etc. By understanding these segments, the council can tailor its engagement strategies to resonate with diverse groups.
3) **Enhancing Transparency and community trust –** Transparency in how public funds are distributed is essential for maintaining public trust. The predictive model and the insights gained from the data collection will be shared with the public.

To achieve this, there are several specific points for the success of the project. Identification of relevant variables, using predictive modelling algorithms to forecast preferences and priorities of various segments *(lifestyle_type).*

By understanding and addressing these objectives, the team aims to build a robust predictive model that not only improves fund allocation but also strengthens community ties and enhances the governance process in Mining City.

1.1.2. MetaData

The training data set contains 21 features that describe some attributes associated with each citizen. Our analysis focuses on the target variable, representing the final competition result. In that project, our target variable is *"lifestyle_type"* and the goal is select the correct variables, create a predictive model and make predictions on unseen data (first in our validation set and after that in the test set). So, our two main goals are:



```
[ ] # Count the number of observations for each "category"
    traindf['lifestyle_type'].value_counts()

    Health-Conscious      18131
    Investor              18059
    Fitness Enthusiast    18033
    Adventure Seeker      18003
    Travel Enthusiast     17939
    Name: lifestyle_type, dtype: int64
```

1. **Enhancing competition outcomes -** We aim to determine if the dataset features can improve competition outcomes. By understanding the relationship between these features and the outcome.
2. **Identifying influential features -** We seek to identify the most impactful features for our prediction model. This allows us to prioritize and focus on the key variables, leading to a more accurate and effective predictive model.

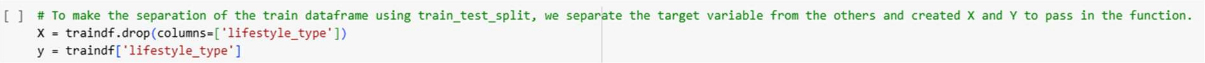
Detailed information about the initial variables of the dataset can be found in annex 1.

2. DATA UNDERSTANDING

In order to better understand the data, there were a few actions to perform, starting by identifying the number of columns and rows presented in the dataset, totalizing a total number of 21 and 90165, respectively, with no duplicated records.

Once performed these actions, the next step was to divide the dataset. The division was made into two parts: 70% for testing and 30% for validation.

To split the dataset, we started by separating the target variables from the remaining, resulting in two dataframes, x and y.



```
[ ] # To make the separation of the train dataframe using train_test_split, we separate the target variable from the others and created X and Y to pass in the function.
    X = traindf.drop(columns=['lifestyle_type'])
    y = traindf['lifestyle_type']
```

Using "train_test_split" function, we divided the training set into separate datasets:

- Training dataset: x_train & y_train - corresponding to 70% - serve to train the model and build it.
- Validation set: x_val & y_val - corresponding to 30% - to validate the model and make some checks.
- Test set: corresponding to the dataset that will be used to make the predictions - to evaluate the performance on the Kaggle competition.

In order to all the addressed transformation can be excluded in the overall features and in the target features too with the same indexer, we concatenated the x and y dataframes, in both train and validation dataset. This action allowed us to not replicate all the transformations in both dataframes and to decrease risk.

After splitting dataframes, since we were not sure about rounding the rating variables, we decided to add them to both datasets under the sentence of not choosing them after analysing the correlation matrix. The variables environmental awareness rating and health consciousness rating were rounded. We also added two new variables: gender and age.

Exploring the data we discovered null values, NA values in both train and validation datasets and described the numerical and categorical data (Annex 2 and 3).

2.1. Variables definition

To easier implement the transformations, observe the visualizations, and verify the consistencies, the variables were categorized, according to the datatype, into the groups below:

metric_features - ['last_year_avg_monthly_charity_donations', 'environmental_awareness_rating', 'financial_wellness_index', 'investment_portfolio_value', 'investments_risk_appetite', 'investments_risk_tolerance', 'tech_savviness_score','social_media_influence_score','entertainment_engagement_factor', 'avg_monthly_entertainment_expenses','avg_weekly_exercise_hours','health_consciousness_rating','stress_management_score','overall_well_being', 'environmental_awareness_rating_rounded', 'health_consciousness_rating_rounded', 'age']

categorical_features - ['name','title','date_of_birth','city','country','Interval of ages','gender']

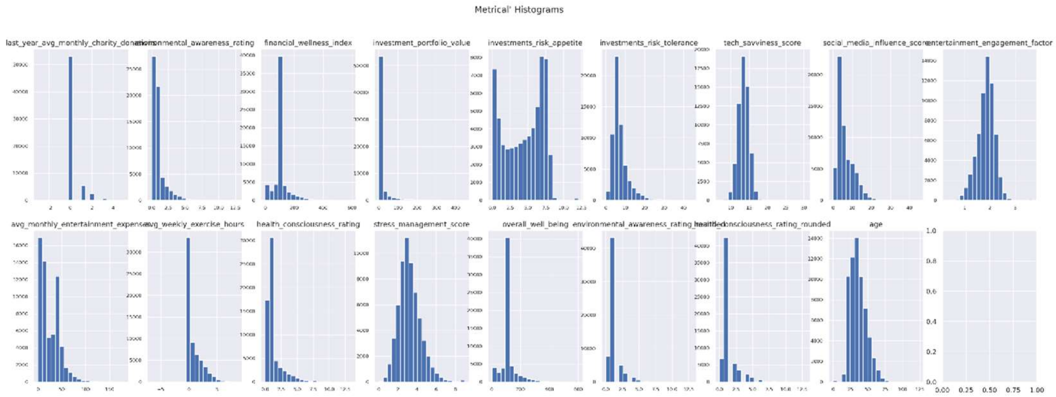
categorical_features_enc - ['city','country','Interval of ages','gender']

2.2. Data Visualization

The first action performed was creating a profile report as a support tool. This allowed us to easily address some points of the features and get insights and access the data in a consolidated manner (Annex 4).

2.2.1. Histograms

For visually explore the data, we started by plotting histograms of the metric features.

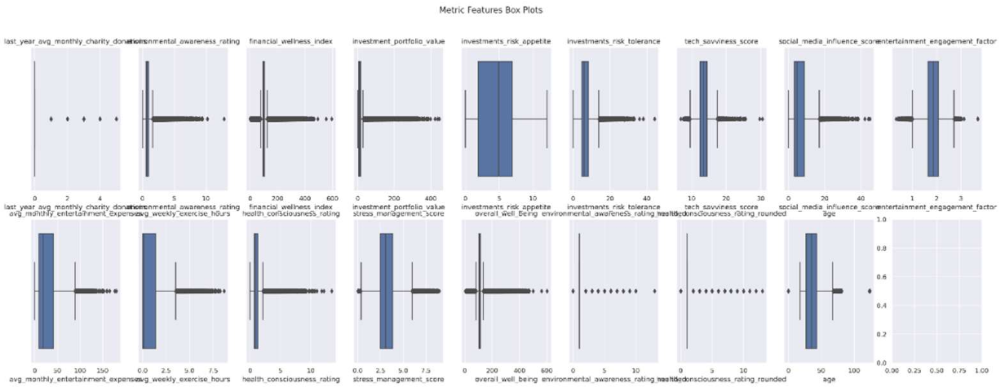


Analysing the histograms we noticed that both last_year_avg_monthly_charity_ donations and avg_weekly_exercise_hours variables contained negative values, which is inconsistent, since with its natures we can't have a negative amount of a donation and a negative number of hours. In response to these issues, we decided to drop the rows which contained negative values in these two columns (Annex 5).

We performed these actions for all datasets: train, validation and test.

2.2.2. Boxplots, Correlation & Outliers

The next visualization performed was boxplots, which helped us identify the outliers in order to reduce them. The representation of the metric features was the following:



Observing the boxplots we noticed the high number of outliers presented in our data, since it was to many, **we decided to first analyse the correlation and pairwise matrix's before treating them**. Even though we know that isn't the normal path, we first tried to eliminate some columns and reduce the number of operations in outlier treatment.

After examining Pairwise Relationships (Annex 6), which didn't give us significant insights, we decided to analyse the correlation matrix for the metric features, to better understand how variables are related to each other.

Analysing the correlation matrix of the metric features we concluded that the most correlated were "stress_management_score" and "overall_well_being", therefore we decided not to treat the outliers of those two variables.

To deal with the outliers of the remaining variables we preformed three methods – manual, IQR and limit imputation – and selected the more beneficial one.

2.2.2.1. Outlier Removal – Manual

The first approach was the manual method. By observing the metric features box plots we set different thresholds for the variables which outliers we wanted to remove. At the end, the percentage of loss data was too high, 22%. Even with 22% of the dataset removed, the problem was not solved as outliers persisted, making this method not a good choice.

2.2.2.2. Outlier Removal – IQR Method

The second method evaluated was the Interquartile Range (IQR), which is calculated as the difference between the third and the first quartile. Using the IQR, Q1 and Q3, we established the lower and upper bounds and then removed values that fall outside these two bounds. With this approach we were able to lose only 1,15% of data, however we still had some outliers that may not be relevant for understanding the model's forecast.

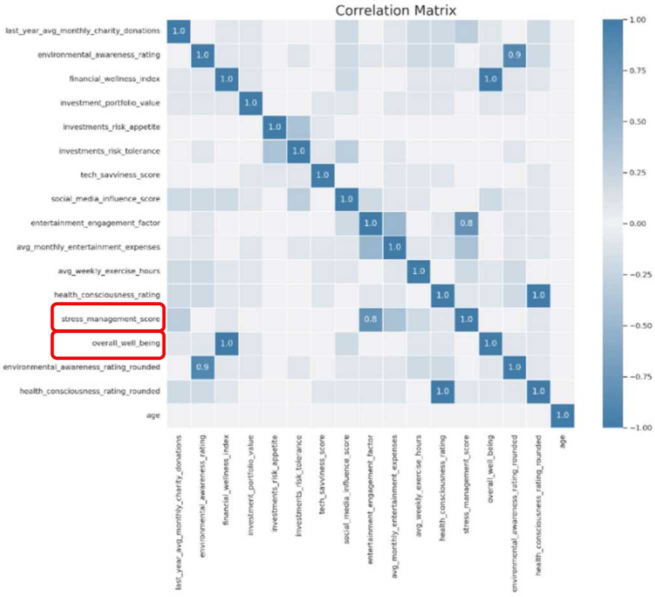
2.2.2.3. Outlier Limit Imposition

Lastly, the final approach was the limit imposition method, in which we set a threshold and replace all the values that exceed it for that specific value. With this method we were able to treat the outliers without losing any data at all. Compared to the previous methods, we considered this one the **best approach** and applied it to all train, validation and test sets.

2.2.3. Correlation Matrix

For better understanding the relationship, and its strength, between the variables after the outlier's treatment, we did another correlation matrix for the metric features.

Upon analysis of the matrix, and as pointed before, we concluded that the variables "stress_management_score" and "overall_well_being" were correlated with other variables, so we decided to drop those. We also dropped "environmental_awareness_rating_rounded" and "health_consciousness_rating_rounded" as it didn't prove it would improve our results.



Besides analysing the correlation, we also analysed the dependency of metric and categorical features with our target variable, lifestyle_type. Since our target variable is categorical, we first needed to convert it to numerical, however, since it contains more than two different string values, we couldn't perform a one-way ANOVA by directly comparing those string values as 0 and 1. Taking that into consideration, so we could preformed ANOVA, we first converted our categorical target variable to numerical codes in a copied dataframe, to not risk altering the original one. However, **we concluded that there was no significant dependence** between both metric and categorical features, as a result, no features were excluded.

2.2.4. Inconsistency Check

To identify potential inconsistencies, we started by checking duplicate records in the dataset. The results showed we had two records duplicates, however, upon analysis we decided to keep them since we did not agree with this result and believe these records are not duplicated.

We also explored the negative values and concluded that there were none in the dataset.

2.2.4.1. Missing Values

The next step was to treat the missing values. We started by testing drop them, however following that approach we were losing 20% of the dataset, and as we didn't want to lose so much data, we followed the approach of filling in the missing values.

For filling the missing values, we started by scaled the data using Min-Max Scaling and then used the KNN Imputer, estimating missing values based on similar datapoints. After doing it in train set, we applied it to validation and test sets.

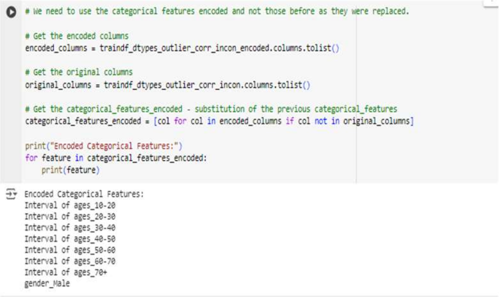
3. DATA PREPROCESSING

Effective data preprocessing is a critical step, as it lays the foundation for accurate and reliable analysis. In our project, where the goal is to enhance the participatory budgeting process in Mining City through predictive modelling, the step of careful preparation of the data is essential.

3.1. Encoding

To ensure that our machine learning algorithms could effectively interpret the categorical variables in our datasets, we performed encoding to transform categorical variables into numerical variables.

We utilized “pd.get_dummies()” function from the Pandas library to achieve this transformation. The “pd.get_dummies()” converts categorical variables into dummy/indicator variables, creating a binary column for each category. We applied this encoding process to the categorical variables: “gender” and “age interval”. For both “gender” and “age interval” variables, the function created new columns, assigning a value of 1 or 0.



```
# We need to use the categorical features encoded and not those before as they were replaced.

# Get the encoded columns
encoded_columns = traindf_dtypes_outlier_corr_incon_encoded.columns.tolist()

# Get the original columns
original_columns = traindf_dtypes_outlier_corr_incon.columns.tolist()

# Get the categorical_features_encoded - substitution of the previous categorical_features
categorical_features_encoded = [col for col in encoded_columns if col not in original_columns]

print("Encoded Categorical Features:")
for feature in categorical_features_encoded:
    print(feature)
```

```
Encoded Categorical Features:
Interval of ages_10-20
Interval of ages_20-30
Interval of ages_30-40
Interval of ages_40-50
Interval of ages_50-60
Interval of ages_60-70
Interval of ages_70+
gender_Male
```

After that, we just made sure there were no Nas in the columns. Even though we did this procedure, we kept the previous categorical features because in the end we realized that the age ranges didn’t help us and didn’t improve the model, therefore we dropped those columns.

3.2. Validating the datasets for modelling

We verified the consistency of column names and alignment within the train and validation sets. Additionally, we ensured that the test set does not include the “lifestyle_type” column, as it is in the test set those predictions about the “lifestyle_type” will be made and the model's performance will be evaluated. In this step, we realized that “interval of ages_10-20” wasn’t present in the testdf and, as expected, the “lifestyle_type” as a target variable was not present either.

3.3. Splitting

Initially, the dataset was divided into train (70%) and validation sets (30%). Within each set, the features (X) and the target variable (Y) in each of these sets were combined for ease of manipulation during preprocessing. However, it was necessary to separate them again. This separation was essential because the previous manipulations were performed on the data itself, and maintaining distinct train and validation sets with both X and Y facilitated subsequent steps in model training and evaluation. Additionally, the test set data frame was renamed to ensure enhanced consistency throughout the code.

3.4. Feature Selection

Feature selection is a critical process in the development of machine learning models. It involves identifying and selecting the most relevant features from a dataset to improve model performance. The main goals of feature selection are improving model performance, reduce overfitting and enhance training efficiency.

In our project some of the features after preprocessing might not be totally required, so we used three feature selection techniques – RFE with Logistic Regression, RFE with the Random Forest Classifier and Neural Network.

RFE methods offer interpretability, whereas Neural Networks excel at identifying complex patterns. The methods selected different feature sets, each contributing to model performance.

- RFE with Logistic Regression yielded a 0.638 F1 score with 13 features.
- RFE with Random Forest achieved a 0.776 F1 score with 12 features.
- Neural Networks method obtained a 0.781 score with 12 features.

After that we analysed the selected features from each of techniques, and the common selected by all methods are: {'tech_savviness_score', 'avg_monthly_entertainment_expenses', 'health_consciousness_rating','age','avg_weekly_exercise_hours', 'investments_risk_tolerance', 'investments_risk_appetite', 'financial_wellness_index','environmental_awareness_rating', 'investment_portfolio_value','social_media_influence_score','entertainment_engagement_factor'}.

3.5. Cross-Validation

Cross-Validation assesses model's adjustment to new dataset by dividing data into subsets and checking for overfitting risk. It evaluates model's consistency and effectiveness multiple times. It estimates model performance with limited data and influences hyperparameter tuning in Random Search Method.

In this section of the project, cross-validation methods were used to better evaluate and understand model performance. These tools provided valuable insights into the applicability and robustness of our models. Our group ensured that the performance metrics, accuracy and F1 score, were reliable, mitigating the risk of overfitting. We applied Stratified K-Folds and Traditional K-Folds cross-validation methods to each model (Linear Regression, Random Forest, Decision Tree and KNN), and for each, we conducted feature selection using Logistic Regression (LR), Random Forest Classifier (RFC), and Multi-Layer Perceptron (MLP).

Accuracy and F1 score are important for evaluating a classification model's performance. Accuracy shows the proportion of correct predictions. It indicates the overall correctness of the model. Accuracy can be deceptive with imbalanced data. F1 Score is the balance between precision and recall, useful for class imbalance. It considers false positives and false negatives for a detailed model evaluation. The model score takes into consideration these two metrics in order to deliver the most accurate result about the quality of the model.

The results obtained can be seen in annex 7.

Regarding Stratified K-Folds, taking into consideration all feature selection methods (LR, RFC, and MLP), the Random Forest Classifier consistently achieved the highest scores, approximately 77%, suggesting that it is the most robust model for this dataset. It displayed the same parameters for feature selection with RFC and MLP, while altering the minimum sample split from 5 to 3 on feature selection conducted with LR. In terms of Decision Trees, this model also demonstrated strong performance, effectively capturing underlying data patterns with consistent scores of about 71% across all feature selection methods. There were no parameter adjustments when changing the feature selection method.

KNN exhibited moderate performance, maintaining a stable score of 67%, which was superior to Logistic Regression, consistently showing the lowest scores. KNN maintained parameter consistency across the three feature selection methods. The consistency of these outcomes across various feature selection methods highlights the reliability of RFC as the top model, followed by Decision Trees and KNN.

When considering Cross-Validation made using considering Traditional K-Folds, the results obtained are similar with the results obtained when doing Cross-Validation with Stratified K-Folds. Therefore, our group can conclude that the most consistent model is Random Forest Classifier, having in note that Decision Trees also performed well on Cross-Validation.

4. MODELLING

4.1. Modelling selection

In this section we addressed different models to be used. We went through each model, and in each, we tried to reach the best outcome, by using normal approaches and with different features selected. The metrics chosen to better evaluate the models were accuracy, F1 Score (that takes into consideration precision and recall) and ROC-AUC. This approach ensures that our models are consistent and applicable to various scenarios.

4.1.1. KNN Classifier

K Nearest Neighbors (KNN) was selected for its simplicity, intuitiveness and effectiveness in handling non-parametric data. It classifies a data point based on how its neighbours are classified.

4.1.2. Random forest Classifier

Random Forest is an ensemble learning method that builds multiple decisions trees during training and outputs the mode of the classes for classification tasks. This model was chosen for its prowess in managing complex datasets and mitigating overfitting.

4.1.3. Ensemble Methods

Ensemble techniques are models that combine the decisions from multiple models to improve overall performance. Instead of relying on a single model to make predictions, ensemble techniques use collective wisdom of several models to achieve better results and by aggregating different models, the technique can mitigate the weakness of individual models.

4.1.4. Logistic regression

Logistic regression is a statistical method used for binary classification problems. It models the probability that a given input belongs to a particular class using logistic function. This model was chosen because it’s easy to implement, interpret, and provides probabilistic predictions.

4.1.5. Decision trees

Decision Trees are non-parametric supervised learning algorithms used for classification and regression. They work by splitting the data into subsets based on feature value tests. Decisions trees are lauded for their interpretability and data flexibility, besides that they can overfit the training data.

4.1.6. Neural network

Neural Network are inspired by the human brain and consist of layers of interconnected nodes (neurons). They can capture complex patterns and relationships in data through their deep learning capabilities. They are very flexible and can be applied to a variety of tasks.

4.1.7. Gradient boosted classifier

Gradient boosted classifier is an advanced ensemble technique that builds models sequentially, each new model correcting errors made by the previous ones. This model is very effective in reducing bias and variance, providing high accuracy.

4.2. Evaluating Metrics Results

The table of the results obtained for every metric evaluated is presented in annex 8. Our analysis is based on three main metrics to compare the models: Accuracy, F1 Score and ROC-AUC. The following analysis will focus on the models that performed better.

Regarding **K-Nearest Neighbors Classifier** (KNN), the model performed with a satisfactory performance having a balanced accuracy of approximately 68% for both for feature selection done with Logistic Regression and for Random Forest Classifier, and with ROC-AUC score is of approximately 89%, which shows a good ability to distinguish between classes. However, when observing the metrics within classes, KNN showed a great ability to predict Travel Enthusiasts (F1 Score of 88%) but struggled to identify other classes (F1 Scores between 61-67%), which shows sensitivity to the data structure.

About the **Random Forest Classifier** (RFC), this model shows a rather high value of accuracy, 77-78%, performing well on all categories, on for both methods of feature selection used. The values of F1 Score were, likewise KNN, higher for the Travel Enthusiast category, and therefore making this model reliable in this category. Despite the similarities with KNN, RFC was able to also recognize the category Investor where the F1 Score is 80%.

When focusing on the **Ensemble model**, we opted for the voting classifier which relies on combining strengths from various models, the results showed a consistent performance across all categories. Regarding Accuracy, this model achieved a value of 78,6%, indicating that the model correctly predicted a significant majority of the instances. Specifically, regarding the category identification, the model effectively identifies the categories Adventure Seekers and Fitness Enthusiasts, with a good balance between precision and recall, which minimizes false negatives and false positives. In what regards the Health-Conscious category, the model still performs well on identifying this category despite the recall value being lower than the precision value, which indicates that the model is effective on identifying this category with a good precision rate. The opposite case happens with the investor case, where both precision and recall values are higher than the category mentioned before. However, the recall value is higher than the precision value. This indicates that the model can minimize the cases of false negatives. The

Ensemble model performs extraordinarily well on identifying the Travel Enthusiasts, with a recall and precision values higher than 90%. This demonstrates the ability of the model to identify this category with very few errors. Therefore, the Ensemble model stands out by having a good balance between precision and recall values, particularly outstanding in the categories Investor and Travel Enthusiast. Its high overall accuracy shows the consistency of the model in making accurate predictions, which makes this model stand out for comprehensive and dependable classification across diverse data segments.

Our analysis found that **Neural Networks** (MLP) consistently delivered some of the best results among the models that were evaluated. When applying feature selection using LR, the Neural Network achieved an overall accuracy of 77.6%, indicating a strong performance across most categories. Regarding the F1 score among the categories, when talking about Adventure Seeker and Fitness Enthusiast categories, both had a precision and recall of 72%, which shows strong performance and balance, therefore, having a reliable classification for these categories. Regarding the Health-Conscious category, the model showed a good overall F1 Score, despite having a lower recall which shows that some Health-Conscious instances were missed, still having a good overall performance. When RFC was utilized for feature selection, the model attained an accuracy of 76.7%. Although precision saw an enhancement for Adventure Seekers, there was a minor decrease in recall, resulting in more instances being overlooked. Despite this observation, the performance in other categories remained stable with LR feature selection.

From an overall perspective, both approaches demonstrated comparable outcomes, yet there was a slightly superior performance in correctly identifying positive outcomes when logistic regression was utilized for feature selection. The primary distinction between the two feature selection methods lies in the precision-recall trade-offs, which indicate that RFC prioritizes precision for Adventure Seekers. Despite these small variations, the Neural Network consistently delivered a dependable performance across various categories, notably excelling in identifying Investors and Travel Enthusiasts.

From these results, there is, statistically, strong evidence to believe that the Random Forest Classifier and Neural Network models show superior performance across most metrics, particularly in terms of ROC-AUC, which indicates their strong ability in distinguishing between categories. It is also relevant to say that Ensemble methods also showed promising results with high accuracy, precision, and recall, that suggests their effectiveness in combining multiple model predictions. Regarding Logistic Regression, it is crucial to highlight that, while showing a moderate performance on delivering results, this model provides a reliable baseline for comparison with other models.

CONCLUSION

In conclusion, the analysis highlight that the Random Forest Classifier (RFC) and Neural Network models demonstrate superior performance across various metrics, notably in ROC-AUC, showcasing their strong capacity to differentiate between different categories. The RFC displayed high accuracy and effectively identified multiple categories, such as Investors. Neural Networks exhibited consistent and reliable performance, excelling in the identification of Investors and Travel Enthusiasts, while maintaining a robust balance between precision and recall. The Ensemble model also achieved high accuracy, precision, and recall, successfully recognizing a wide array of categories, and reducing false positives and negatives. Although Logistic Regression displayed modest performance, it serves as a dependable benchmark for comparison. In general, RFC, Neural Networks, and Ensemble methods are notable for their thorough and reliable classification abilities across a variety of data segments.

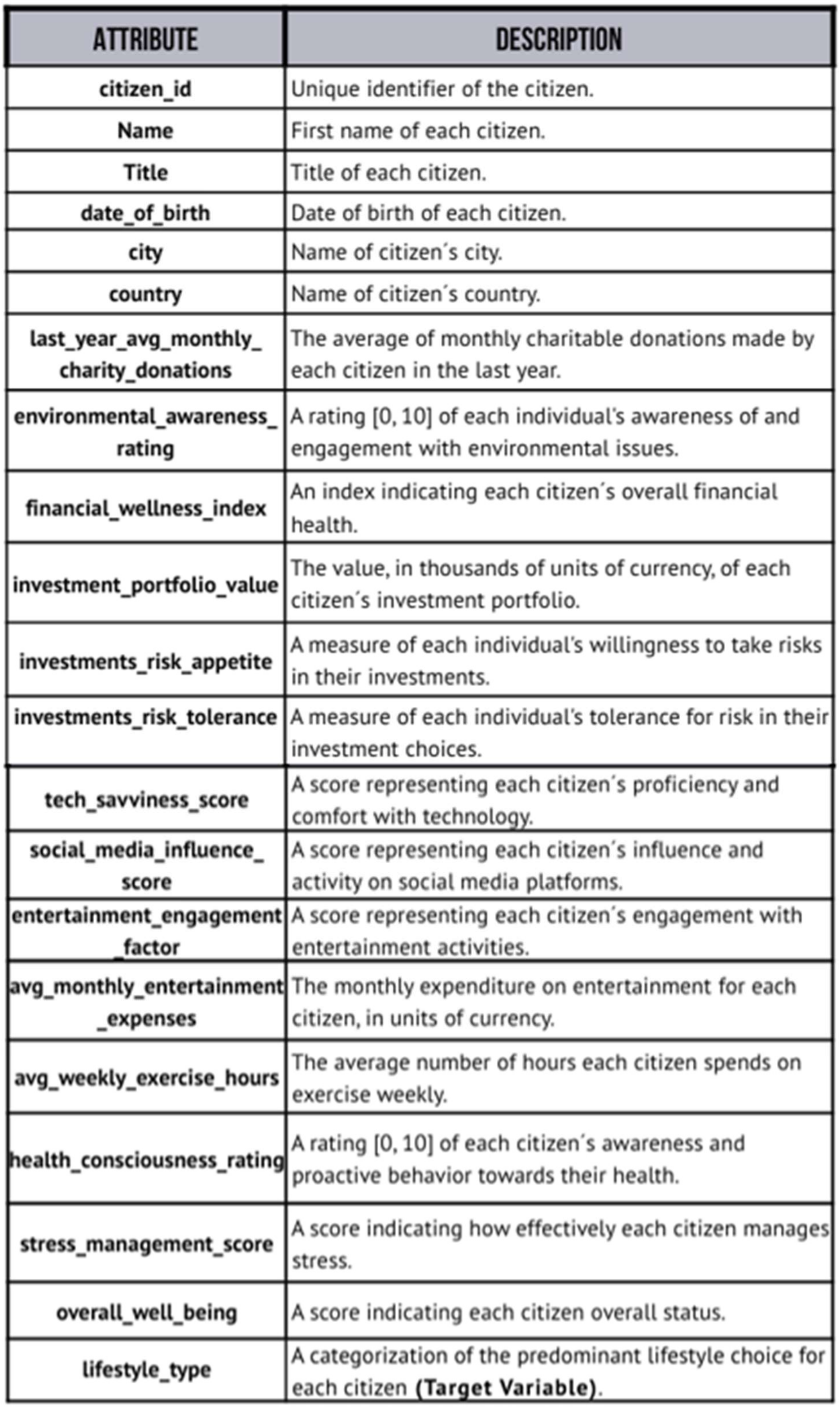
The outcomes of this research closely aligned with our original anticipations. We postulated that by incorporating predictive analytics and citizen feedback, the budgeting procedure could be improved to be more efficient and fairer. The findings substantiated our hypothesis, illustrating that our predictive model had the capability to enhance the correlation between budget allotments and citizen preferences. The ability to accurately predict budget requirements and identify critical areas of focus validated our belief that utilizing data-driven methods could significantly enhance participatory budgeting practices.

Throughout the project we encountered some limitations that made our work more challenging. The main difficulty was the extended time needed to run the code in light version of Collab. Therefore, to resolve this issue we decided, as a group, to upgrade the version to Collab Pro. This optimization allowed us to run the code faster and more smoothly, reducing our running time by over 50%. However, despite these improvements, we were unable to run the hyperparameter tuning for the models due to excessive run times. The process was so lengthy that it caused disconnections, making it impossible to obtain the hyperparameter results. Another problem encountered were the outliers. There were many of them, so it made it difficult to decide what was the best approach to use in treating them and what to remove. Lastly, but not least, was of deciding the best model to use: do we use every model or just select some.

We consider that future research should concentrate on the integration of a variety of data sources, such as real-time economic indicators and social media trends, in order to enhance the accuracy of predictions. The utilization of advanced modelling techniques, such as neural networks and ensemble methods, can further improve the precision of forecasts. The creation of user-friendly, interactive platforms is crucial for enhancing citizen engagement and facilitating feedback collection. Expanding the application of the model to different cities will allow for the evaluation of its scalability and generalizability. In addition, carrying out longitudinal studies will enable the assessment of the long-term effects of predictive modelling on participatory budgeting, offering valuable insights for continual enhancement and broader adoption.

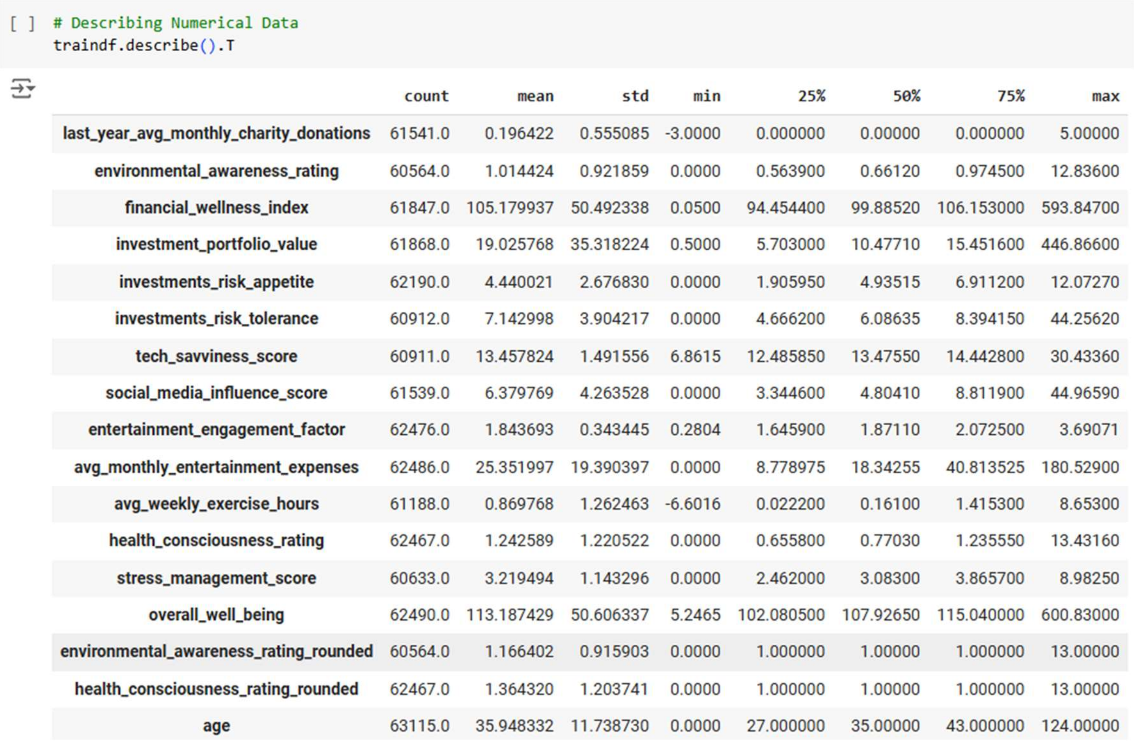
ANNEX

Annex 1



ATTRIBUTE	DESCRIPTION
citizen_id	Unique identifier of the citizen.
Name	First name of each citizen.
Title	Title of each citizen.
date_of_birth	Date of birth of each citizen.
city	Name of citizen´s city.
country	Name of citizen´s country.
last_year_avg_monthly_charity_donations	The average of monthly charitable donations made by each citizen in the last year.
environmental_awareness_rating	A rating [0, 10] of each individual's awareness of and engagement with environmental issues.
financial_wellness_index	An index indicating each citizen´s overall financial health.
investment_portfolio_value	The value, in thousands of units of currency, of each citizen´s investment portfolio.
investments_risk_appetite	A measure of each individual's willingness to take risks in their investments.
investments_risk_tolerance	A measure of each individual's tolerance for risk in their investment choices.
tech_savviness_score	A score representing each citizen´s proficiency and comfort with technology.
social_media_influence_score	A score representing each citizen´s influence and activity on social media platforms.
entertainment_engagement_factor	A score representing each citizen´s engagement with entertainment activities.
avg_monthly_entertainment_expenses	The monthly expenditure on entertainment for each citizen, in units of currency.
avg_weekly_exercise_hours	The average number of hours each citizen spends on exercise weekly.
health_consciousness_rating	A rating [0, 10] of each citizen´s awareness and proactive behavior towards their health.
stress_management_score	A score indicating how effectively each citizen manages stress.
overall_well_being	A score indicating each citizen overall status.
lifestyle_type	A categorization of the predominant lifestyle choice for each citizen **(Target Variable)**.

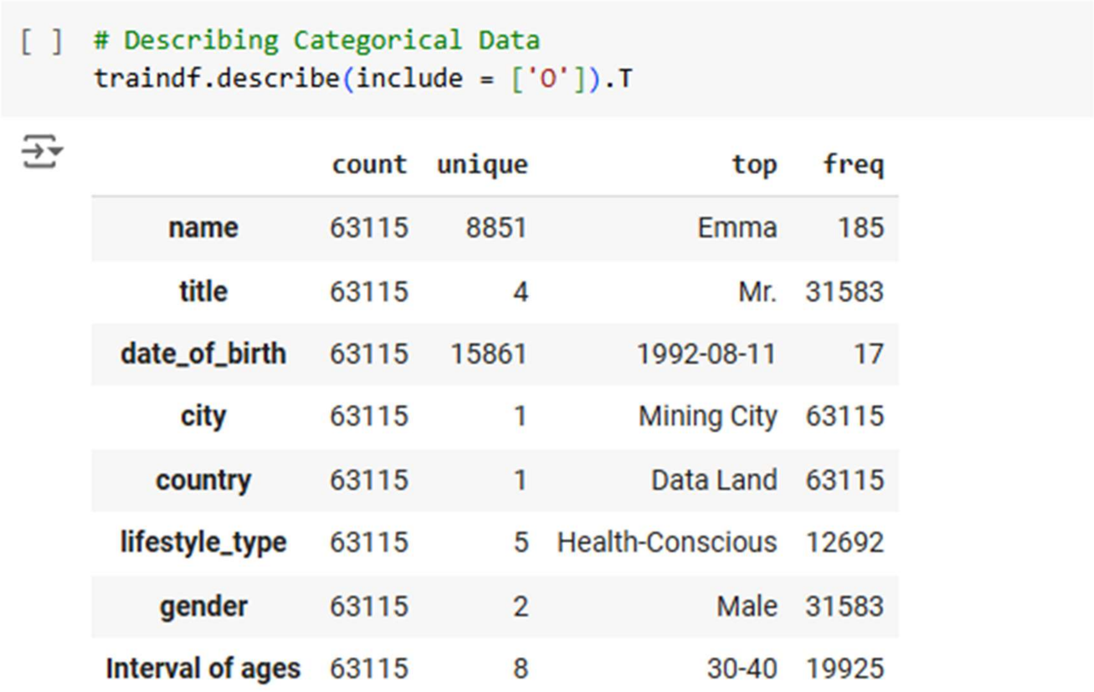
Annex 2



```
[ ] # Describing Numerical Data
    traindf.describe().T
```

	count	mean	std	min	25%	50%	75%	max
last_year_avg_monthly_charity_donations	61541.0	0.196422	0.555085	-3.0000	0.000000	0.00000	0.000000	5.00000
environmental_awareness_rating	60564.0	1.014424	0.921859	0.0000	0.563900	0.66120	0.974500	12.83600
financial_wellness_index	61847.0	105.179937	50.492338	0.0500	94.454400	99.88520	106.153000	593.84700
investment_portfolio_value	61868.0	19.025768	35.318224	0.5000	5.703000	10.47710	15.451600	446.86600
investments_risk_appetite	62190.0	4.440021	2.676830	0.0000	1.905950	4.93515	6.911200	12.07270
investments_risk_tolerance	60912.0	7.142998	3.904217	0.0000	4.666200	6.08635	8.394150	44.25620
tech_savviness_score	60911.0	13.457824	1.491556	6.8615	12.485850	13.47550	14.442800	30.43360
social_media_influence_score	61539.0	6.379769	4.263528	0.0000	3.344600	4.80410	8.811900	44.96590
entertainment_engagement_factor	62476.0	1.843693	0.343445	0.2804	1.645900	1.87110	2.072500	3.69071
avg_monthly_entertainment_expenses	62486.0	25.351997	19.390397	0.0000	8.778975	18.34255	40.813525	180.52900
avg_weekly_exercise_hours	61188.0	0.869768	1.262463	-6.6016	0.022200	0.16100	1.415300	8.65300
health_consciousness_rating	62467.0	1.242589	1.220522	0.0000	0.655800	0.77030	1.235550	13.43160
stress_management_score	60633.0	3.219494	1.143296	0.0000	2.462000	3.08300	3.865700	8.98250
overall_well_being	62490.0	113.187429	50.606337	5.2465	102.080500	107.92650	115.040000	600.83000
environmental_awareness_rating_rounded	60564.0	1.166402	0.915903	0.0000	1.000000	1.00000	1.000000	13.00000
health_consciousness_rating_rounded	62467.0	1.364320	1.203741	0.0000	1.000000	1.00000	1.000000	13.00000
age	63115.0	35.948332	11.738730	0.0000	27.000000	35.00000	43.000000	124.00000

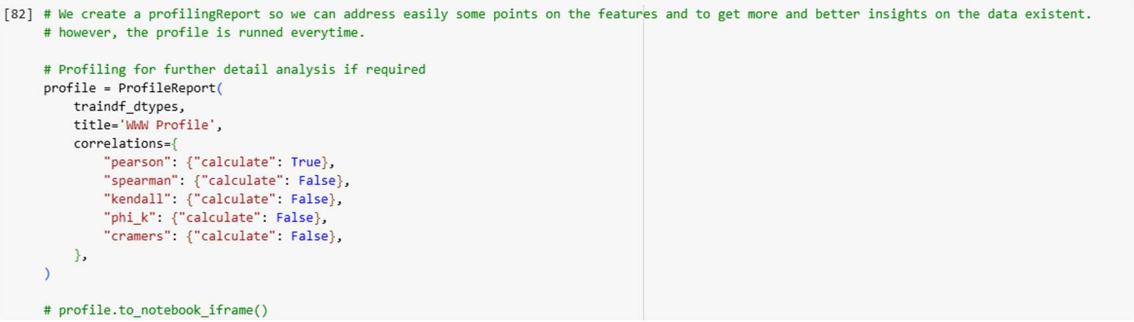
Annex 3



```
[ ] # Describing Categorical Data
    traindf.describe(include = ['O']).T
```

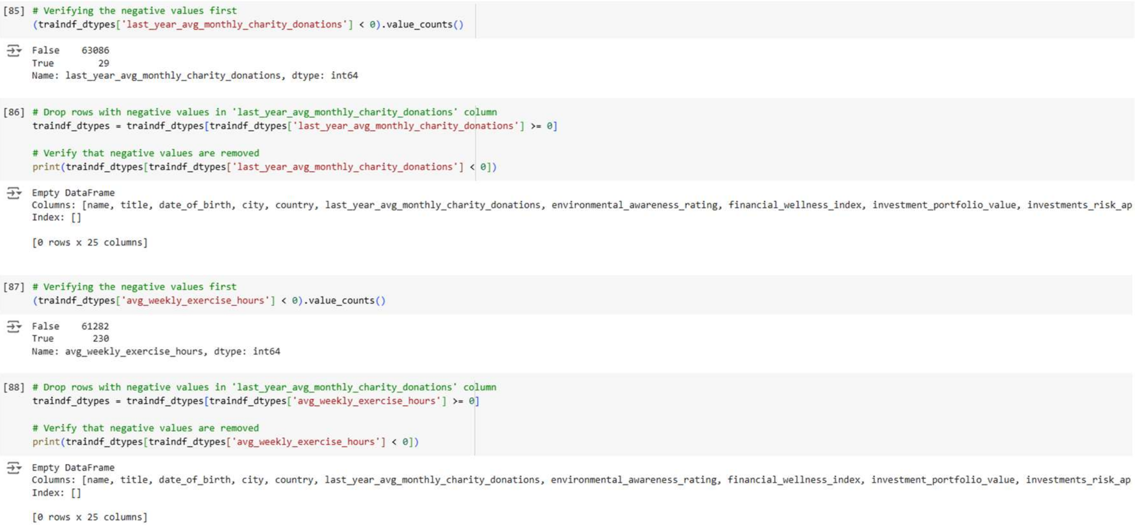
	count	unique	top	freq
name	63115	8851	Emma	185
title	63115	4	Mr.	31583
date_of_birth	63115	15861	1992-08-11	17
city	63115	1	Mining City	63115
country	63115	1	Data Land	63115
lifestyle_type	63115	5	Health-Conscious	12692
gender	63115	2	Male	31583
Interval of ages	63115	8	30-40	19925

Annex 4



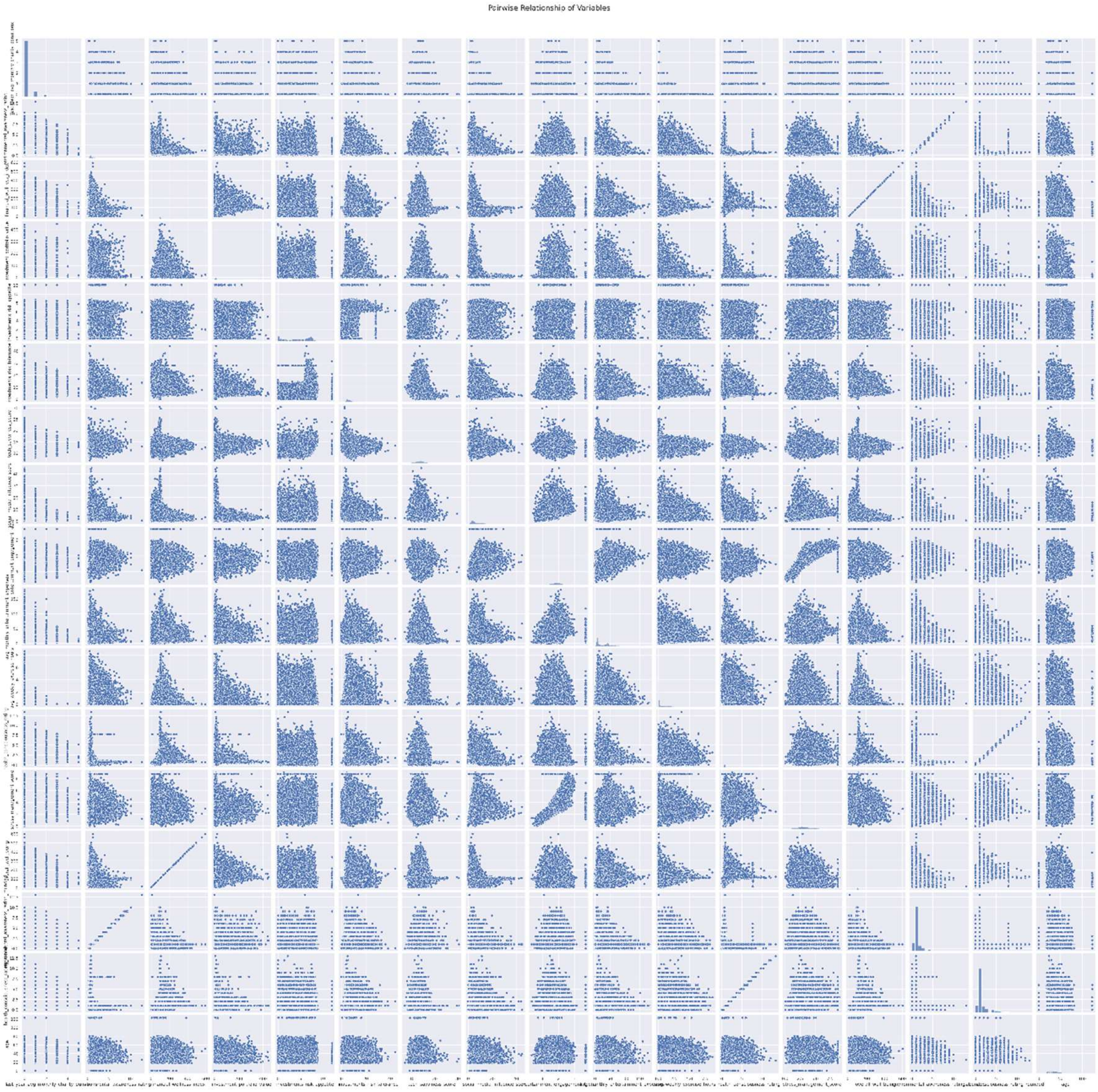
```
[82] # We create a profilingReport so we can address easily some points on the features and to get more and better insights on the data existent.
     # however, the profile is runned everytime.

     # Profiling for further detail analysis if required
     profile = ProfileReport(
         traindf_dtypes,
         title='WWW Profile',
         correlations={
             "pearson": {"calculate": True},
             "spearman": {"calculate": False},
             "kendall": {"calculate": False},
             "phi_k": {"calculate": False},
             "cramers": {"calculate": False},
         },
     )

     # profile.to_notebook_iframe()
```

Annex 5



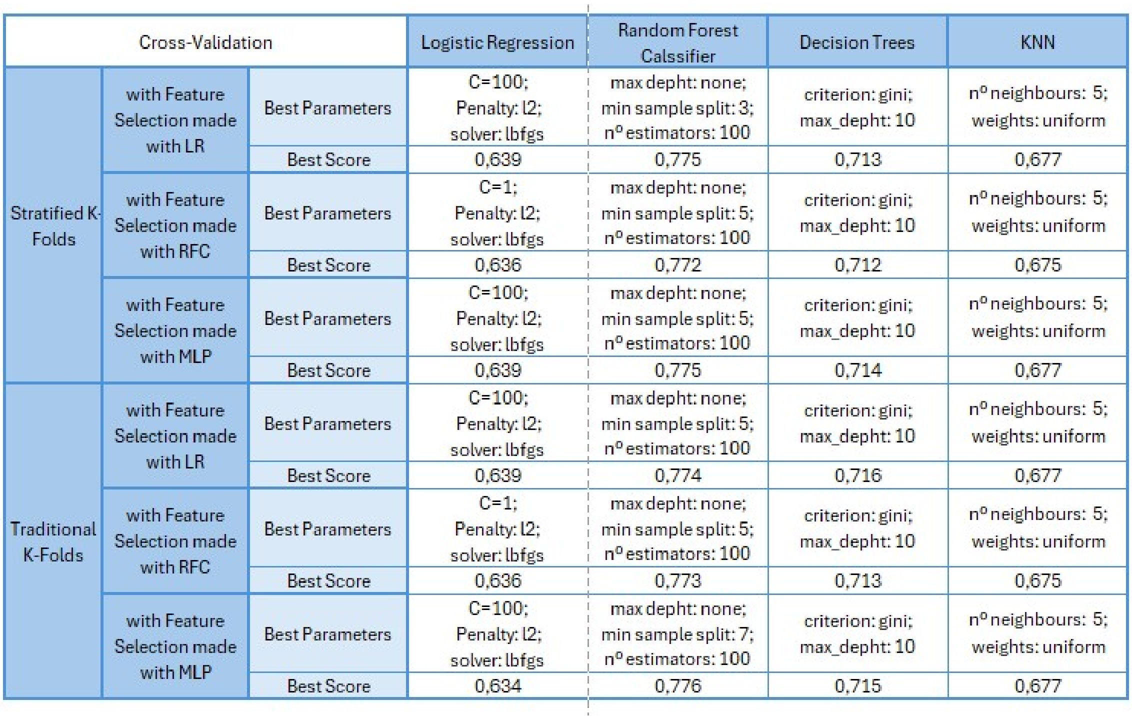
```
[85] # Verifying the negative values first
     (traindf_dtypes['last_year_avg_monthly_charity_donations'] < 0).value_counts()

     False    63086
     True        29
     Name: last_year_avg_monthly_charity_donations, dtype: int64

[86] # Drop rows with negative values in 'last_year_avg_monthly_charity_donations' column
     traindf_dtypes = traindf_dtypes[traindf_dtypes['last_year_avg_monthly_charity_donations'] >= 0]

     # Verify that negative values are removed
     print(traindf_dtypes[traindf_dtypes['last_year_avg_monthly_charity_donations'] < 0])

     Empty DataFrame
     Columns: [name, title, date_of_birth, city, country, last_year_avg_monthly_charity_donations, environmental_awareness_rating, financial_wellness_index, investment_portfolio_value, investments_risk_ap
     Index: []

     [0 rows x 25 columns]

[87] # Verifying the negative values first
     (traindf_dtypes['avg_weekly_exercise_hours'] < 0).value_counts()

     False    61282
     True       230
     Name: avg_weekly_exercise_hours, dtype: int64

[88] # Drop rows with negative values in 'last_year_avg_monthly_charity_donations' column
     traindf_dtypes = traindf_dtypes[traindf_dtypes['avg_weekly_exercise_hours'] >= 0]

     # Verify that negative values are removed
     print(traindf_dtypes[traindf_dtypes['avg_weekly_exercise_hours'] < 0])

     Empty DataFrame
     Columns: [name, title, date_of_birth, city, country, last_year_avg_monthly_charity_donations, environmental_awareness_rating, financial_wellness_index, investment_portfolio_value, investments_risk_ap
     Index: []

     [0 rows x 25 columns]
```

Annex 6



Pairwise Relationship of Variables

Annex 7



Cross-Validation			Logistic Regression	Random Forest Calssifier	Decision Trees	KNN

Stratified K-Folds	with Feature Selection made with LR	Best Parameters	C=100;
			Penalty: l2;
			solver: lbfgs
				max depht: none;
				min sample split: 3;
				nº estimators: 100
					criterion: gini;
					max_depht: 10
						nº neighbours: 5;
						weights: uniform

		Best Score	0,639	0,775	0,713	0,677

	with Feature Selection made with RFC	Best Parameters	C=1;
			Penalty: l2;
			solver: lbfgs
				max depht: none;
				min sample split: 5;
				nº estimators: 100
					criterion: gini;
					max_depht: 10
						nº neighbours: 5;
						weights: uniform

		Best Score	0,636	0,772	0,712	0,675

	with Feature Selection made with MLP	Best Parameters	C=100;
			Penalty: l2;
			solver: lbfgs
				max depht: none;
				min sample split: 5;
				nº estimators: 100
					criterion: gini;
					max_depht: 10
						nº neighbours: 5;
						weights: uniform

		Best Score	0,639	0,775	0,714	0,677

Traditional K-Folds	with Feature Selection made with LR	Best Parameters	C=100;
			Penalty: l2;
			solver: lbfgs
				max depht: none;
				min sample split: 5;
				nº estimators: 100
					criterion: gini;
					max_depht: 10
						nº neighbours: 5;
						weights: uniform

		Best Score	0,639	0,774	0,716	0,677

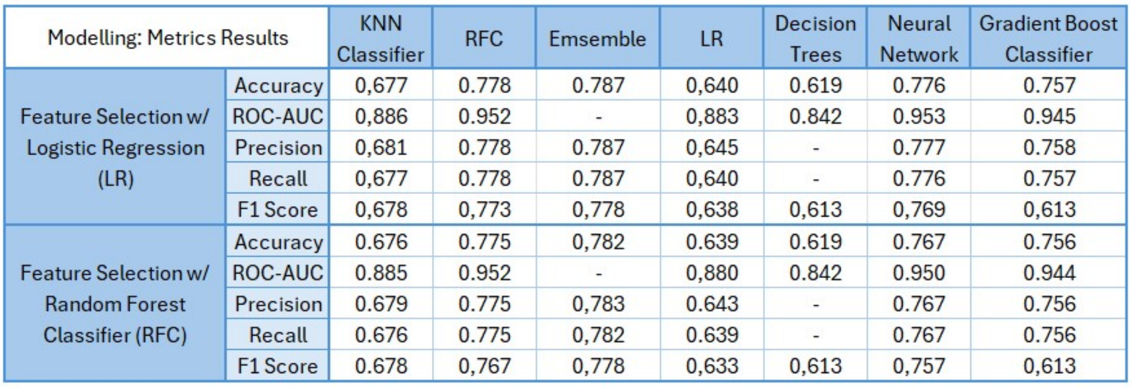
	with Feature Selection made with RFC	Best Parameters	C=1;
			Penalty: l2;
			solver: lbfgs
				max depht: none;
				min sample split: 5;
				nº estimators: 100
					criterion: gini;
					max_depht: 10
						nº neighbours: 5;
						weights: uniform

		Best Score	0,636	0,773	0,713	0,675

	with Feature Selection made with MLP	Best Parameters	C=100;
			Penalty: l2;
			solver: lbfgs
				max depht: none;
				min sample split: 7;
				nº estimators: 100
					criterion: gini;
					max_depht: 10
						nº neighbours: 5;
						weights: uniform

		Best Score	0,634	0,776	0,715	0,677

Annex 8



Modelling: Metrics Results		KNN Classifier	RFC	Emsemble	LR	Decision Trees	Neural Network	Gradient Boost Classifier
Feature Selection w/ Logistic Regression (LR)	Accuracy	0,677	0.778	0.787	0,640	0.619	0.776	0.757
	ROC-AUC	0,886	0.952	-	0,883	0.842	0.953	0.945
	Precision	0,681	0.778	0.787	0,645	-	0.777	0.758
	Recall	0,677	0.778	0.787	0,640	-	0.776	0.757
	F1 Score	0,678	0,773	0,778	0,638	0,613	0,769	0,613
Feature Selection w/ Random Forest Classifier (RFC)	Accuracy	0.676	0.775	0,782	0.639	0.619	0.767	0.756
	ROC-AUC	0.885	0.952	-	0,880	0.842	0.950	0.944
	Precision	0.679	0.775	0,783	0.643	-	0.767	0.756
	Recall	0.676	0.775	0,782	0.639	-	0.767	0.756
	F1 Score	0.678	0,767	0,778	0,633	0,613	0,757	0,613