An Ensemble Machine Learning Based Model for Multiple Imputation of Categorical Data

BACHELOR TERM PROJECT REPORT

By

ARYAN SINHA
(19MF3IM03)

Under the supervision of

Prof. J. MAITI
Department of Industrial and Systems Engineering, IIT Kharagpur

Department of Industrial and Systems Engineering
Indian Institute of Technology Kharagpur
West Bengal, India
4th May, 2023

DECLARATION

I certify that

a. The work contained in this report has been done by me under the guidance of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

Date: 4thMay, 2023
Place: Kharagpur

Name: ARYAN SINHA
Roll number: 19MF3IM03

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
KHARAGPUR-721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**An Ensemble Machine Learning Based Model for Multiple Imputation of Categorical Data**" submitted by **Aryan Sinha (19MF3IM03)** to Indian Institute of Technology, Kharagpur towards partial fulfillment of requirements for the award of degree of Dual Bachelor of Technology (Hons.) in Mechanical Engineering is a record of bona fide work carried out by him under my supervision and guidance during Autumn Semester 2022-2023.

Date: 04/05/2023
Place: Kharagpur

Prof J Maiti
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur
Kharagpur, India

Acknowledgement

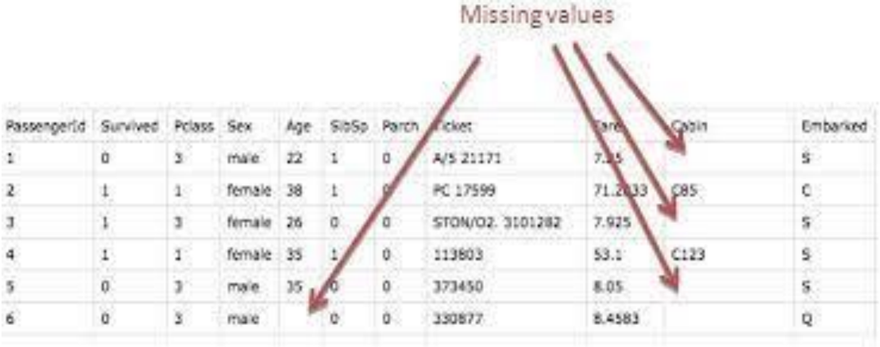
I would like to express my gratitude to **Prof. J Maiti** for giving me this great opportunity and providing his valuable guidance and support in completing my project.

We acknowledge the **Centre of Excellence in Safety Engineering and Analytics (CoE-SEA)** (www.iitkgp.ac.in/department/SE), IIT Kharagpur and Safety Analytics & Virtual Reality (SAVR) Laboratory (www.savr.iitkgp.ac.in) of the Department of Industrial & Systems Engineering, IIT Kharagpur for experimental/computational and research facilities for this work.

CONTENTS

Chapter1 **Introduction**

The values or data for some variables in the supplied dataset that are not stored (or not existent) are referred to as missing data. Nearly all sample surveys and censuses suffer from item nonresponse. Missing data is a common issue in statistical analysis and can occur in any type of data. Incomplete data can bias the results of the machine learning models and/or reduce the accuracy of the model. For any dataset, the missing values are generally represented by blank values. An example of how missing dataset look like is shown below in the figure.



Missing values

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q

Fig 1.1. Illustration of missing values

(adapted from https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/)

1.1 **Missing dataset**

Missing data is a common issue that arises in data analysis. It can occur due to various reasons, and identifying the cause of missing data is essential for effectively handling it. There can be numerous reasons why certain values are missing from the dataset, and each reason affects the approach of handling the missing data.

One of the primary reasons for missing data is improper maintenance of data, which can lead to data corruption. This can happen due to software glitches, hardware failures, or even natural disasters. Another common reason for missing data is human error, such as failing to record data or making mistakes during data entry.

Additionally, missing data can be intentional or unintentional. Sometimes, the user may intentionally leave out values due to privacy concerns or other reasons. Item nonresponse is another reason for missing data, which means the participant refused to respond to certain questions or provide certain values.

Understanding the reason behind missing data is crucial for effectively handling it. Different approaches, such as imputation or deletion, may be used depending on the cause and extent of missing data. By identifying the cause of missing data, researchers and analysts can improve data quality and make more informed decisions.

1.2 **Types of Missing Data**

Broad classification of missing values are as follows:

MCAR
Missing Values
MNAR
MAR

Fig 1.2. Types of missing values

(adapted from https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/)

1.2.1 Missing Completely at Random (MCAR)

The likelihood of missing data in MCAR is constant across all observations. There is no correlation between the missing data in this instance and any other observed or missing (data that is not recorded) values in the provided dataset. In other words, missing values are totally unrelated to other data and no pattern can be found. In the case of MCAR data, the value may be absent as a result of human mistake, equipment or system failure, sample loss, or unsatisfactory technicality during the recording of the values.

Consider the situation of several overdue books in a library. In the computer system, several values for overdue books are missing. A human mistake, such as the librarian forgetting to fill in the values, might be the cause. Therefore, there is no connection between the missing values for overdue books and any other system variable or data. Since it is a rare occurrence, it should not be assumed. Such data have the benefit of maintaining the objectivity of the statistical study.

1.2.2 Missing at Random (MAR)

MAR data indicates that there is some relationship between the missing data and other values or data, and the cause of the missing values can be accounted for by variables for which you have complete information. In this case, not all of the observations are missing any data. There is some consistency in the missing numbers, and it is only missing within certain subsets of the data.

For instance, if you look at the survey results, you could discover that everyone has indicated their gender, but that the majority of those who indicated "female" had missing age figures. Therefore, the likelihood of missing data is solely dependent upon the observed value or data. The variables "Gender" and "Age" are connected in this situation. The 'Gender' variable can be used to explain why the 'Age' variable has missing values, but you cannot anticipate the missing value itself. Imagine a survey is conducted about overdue books in a library. The survey includes questions on gender and the quantity of overdue books. Assume that women are more likely to respond to the survey than men are. Therefore, another factor, namely gender, can be used to explain why the data is missing. The statistical analysis in this instance can produce bias. The only way to estimate the parameters objectively is to model the missing data.

1.2.3 Missing Not at Random (MNAR)

Missing values depend on the unobserved data. It is deemed missing not at random if there is a structure or pattern in the missing data and other observed data cannot account for it (MNAR). It may occur as a result of people being reluctant to supply the necessary information. Certain survey questions may go unanswered by a certain group of respondents.

Let’s look at this example. Consider a scenario where a library's name and the quantity of overdue books are mentioned in a vote. Therefore, the majority of those who have no overdue books are likely to respond to the poll. Less people are likely to respond to the poll if they have more overdue books. In this instance, the persons who have more past-due books determine the value of the number of overdue volumes that is missing.

1.3 **Problems associated with missing data**

Missing data can cause serious problems in the domain of machine learning and data analysis, since it affects the precision and dependability of the outcomes. The most serious issue brought on by missing data is biased findings. A biased sample that does not fairly represent the population might result from missing data. As a result, the model may learn patterns that don't correspond to reality, which might result in predictions and conclusions that are erroneous. The statistical power of the study is lowered by missing data. Prediction accuracy may suffer as it becomes harder to identify significant impacts with fewer data points. Missing data can be problematic for machine learning algorithms since they require entire datasets to identify patterns and generate reliable predictions. Additionally, missing data may obscure significant trends in the data and make it more challenging to determine relationships between variables.

The process of dealing with missing data might take a lot of time and work. To assess the reliability of the findings, data cleaning, imputation, and sensitivity analysis may be necessary. This may make it more difficult to carry out investigations and analysis quickly and may cause delays in making crucial decisions based on the findings.

Finally, missing data might make it harder to trust the outcomes of an investigation or analysis. Lack of confidence in the results might emerge from not being able to determine whether the absence of the missing data would have affected the outcomes or not. This can make it harder to utilize the data to inform key decisions since decision-makers may be unclear of how accurate and dependable the results are.

1.4 **Imputation**

Imputation is a statistical technique for substituting estimated values based on the available data for missing values in a dataset. Missing information may be the result of a number of factors, including incomplete surveys or questionnaires, technical difficulties during data collection, or purposeful omission of values. Imputation aids in maintaining the sample size and minimizing bias that would be introduced into statistical analysis if observations with missing values were simply removed.

Imputation methods come in a variety of forms, from straightforward ones like mean or median imputation to more intricate ones like multiple imputation, hot-deck imputation, and k-nearest neighbor imputation. The type of missing data and the objectives of the study will influence the imputation method selection.

Imputation is frequently employed in a variety of disciplines, including the social sciences, economics, epidemiology, and engineering, where research investigations and data analysis frequently involve missing data.

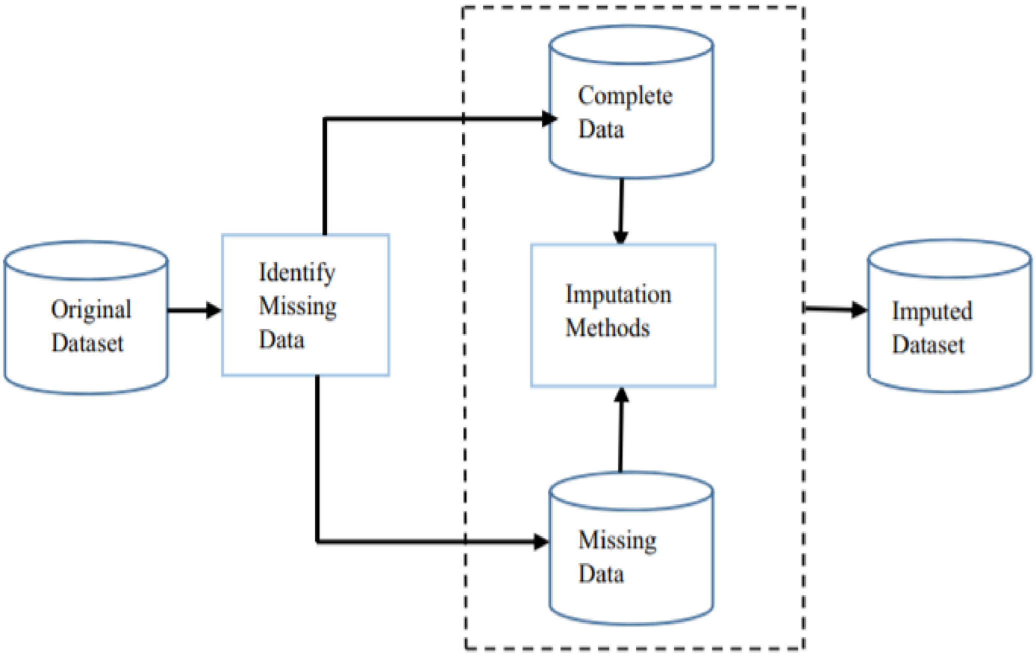


Fig 1.3. General architecture of missing value imputation

(adapted from https://bookdown.org/mwheymans/bookmi/multiple-imputation.html)

There broad classification of Imputation techniques are shown below –

1.4.1 Listwise deletion

You can remove the entire row if there are several missing values in a row. If there is a missing (column) value in every row, you might have to delete the entire set of data. With this technique, the analysis is done without include any observations that have any missing values. This approach is straightforward and simple to use, but if the missing data is not entirely random, it may result in information loss and skew the conclusions (MCAR). If the sample size is small, this method may also lower the analysis's power.

1.4.2 Single imputation

This method includes imputing a single value for each missing value. For continuous variables, the mean, median, or mode can be used, while for categorical variables, the value with highest frequency can be used. Single imputation is a simple approach, however if the missing data is not totally random (MCAR) or absent at random, it may produce biased findings (MAR). The uncertainty brought on by the missing data is also not taken into account by the imputed values, which can result in an underestimation of the standard errors.

- Imputation using the mean: The mean is an appropriate numerical value measurement for the missing location. Calculate the average of the observed values across the entire column before impute the missing attribute value. This approach is not always appropriate and occasionally could result in erroneous allegations.

$$\text{mean} = \frac{\text{sum of observed value}}{\text{total number of observed value}}$$

Eq 1.1

- <u>Imputation using the median</u>: The median calculates the numerical value of the middle position (middle value) throughout the whole column to replace the value of the missing attribute.

$$\text{median} \ = \frac{\text{observed value} \ + \ 1}{2} \text{ if observed is even}$$

Eq 1.2

$$\text{median} \ = \frac{\frac{\text{observed}}{2} + \frac{\text{observed}}{2} + 1}{2} \text{ if observed is odd}$$

Eq 1.3

- <u>Imputation using the mode:</u> For small scale data, the mode is the measurement of the essential leaning. The vacant place must be filled up with the values from each column that appear the most frequently. Since mode returns a data frame and contrasts mean and median, imputing with mode is a little more challenging.

$$\text{mode} = \ l + h\frac{(\text{fm} - \text{f1})}{(\text{fm} - \text{f1} + \text{fm} - \text{f2})}$$

Eq 1.4

where the ‘l’ is considered as lower limit and ‘h’ is size of observed data, f_m is iteration count.

- <u>Previous value imputation (PVI)</u>: Forward filling, sometimes referred to as PVI, is the process of impute the missing value using the previous value.
- <u>NVI</u>: Next Value Imputed using the next value to infer the missing value is referred to as backward filling or NVI.
- <u>Average of the previous and next value imputation</u>: Calculating the average of the prior and subsequent values in order to impute the missing value results in a more accurate approximation of the missing value.
- <u>Arbitrary Value Imputation</u>: Both numerical and categorical variables may be filled using this method. AVI imputation consists of assigning a random value to all missing

values inside an attribute. The arbitrary numbers in this case are 0, 999, a further 999 combinations of 9, or a negative number for a positive distribution.

- Frequent Category Imputation: Used to impute the category variable that is missing. Here, the missing position must be filled in with the values from each column that are repeated the most. This is referred to as mode imputation as well. Generally, this method is used for categorical data.

1.4.3 Multiple imputation

Multiple imputation is a statistical technique used to address missing data in a dataset. The method involves creation of multiple imputed datasets, where the final estimate is obtained by merging several imputations of the missing data that were created using this statistical technique. The process of multiple imputation involves three main steps:

- Imputation: In this step, estimated values based on a statistical model are used to replace missing values. The imputation model might be a straightforward approach, like mean imputation, or a more intricate approach, like regression imputation, which considers the connections between the variables in the dataset.
- Analysis: After the missing values are computed, each computed dataset is subjected to analysis separately. The analysis's findings are then integrated to get the outcome.
- Pooling: The analysis's findings are merged across all of the datasets that have been imputed in the last phase to provide a single set of estimates that takes the imputation process's uncertainty into account.

1.5 Why Multiple Imputation (MI)

Compared to other imputation methods such as listwise deletion or single imputation, multiple imputation provides several advantages that make it a more effective approach.

First of all, listwise deletion, which entails eliminating instances with any missing data from the analysis, is inferior to multiple imputation. Due to the smaller sample size, this approach may produce estimates that are biassed and have lower statistical power. But with multiple imputation, auxiliary variables can be added to the imputation model, increasing the precision of the imputed values.

Second, multiple imputation generates more accurate estimates and takes into consideration the uncertainty surrounding the real value in comparison to single imputation approaches like mean imputation or regression imputation. This is because the variance of the imputed values, which is a reflection of the uncertainty caused by the missing data, is taken into consideration. Researchers can measure the variability of the estimations across several imputations and achieve more precise and trustworthy estimates of the population characteristics by producing numerous imputed datasets.

In addition, when compared to single imputation procedures, multiple imputation can lower bias, standard errors, and enhance accuracy. Multiple imputation can lessen the bias caused by disregarding missing data and offer more precise estimates of population parameters by taking into account the uncertainty around missing values. By adding data from the imputed values, it also decreases the standard errors and improves the accuracy of the estimations.

As a strong approach to filling in the gaps left by missing data, multiple imputation has a number of benefits over other imputation methods. It can raise the precision of the outcomes, decrease bias and standard errors, and enhance estimate accuracy. Multiple imputation is a useful technique for researchers dealing with missing data since it accounts for the uncertainty caused by missing data and produces more accurate and robust results.

Chapter2 **Literature review**

Researchers have devised a range of default algorithms to execute multiple imputation, but there has been limited study comparing the performance of different methods, notably for categorical data. There are traditionally three multiple Imputation methods by which particularly categorical data can be imputed. The goal or dependent variable, which is based on observed data, is said to have a value. Missing data show that an attribute, often known as an independent variable, has no value. The next stage is to use ensemble methods to impute the missing values based on the correlation between the observed and missing data. The ensemble approaches concentrate on filling in the accurate value in the supplied dataset without sacrificing the quality of the information in order to analyze the data in an effective manner. Following is the table of all the notable research papers:

Table 2.1: Literature Survey

Sn no.	Title	Author’s name	Content
1	An Empirical Comparison of Multiple Imputation Methods for Categorical Data	Olanrewaju Akande, Fan Li & Jerome Reiter	Discuss about the three default methods (MI-GLM, MI-CART, MI-DPM) to solve MI problems.
2	Missing Data Imputation Using Ensemble Learning Technique: A Review	K. Jegadeeswari, R. Ragunath, and R. Rathipriya	Discusses some Ensemble learning methods.
3	Mice: Multivariate Imputation by Chained Equations in R	Stef van Buuren, Karin Groothuis-Oudshoorn	Detailed description about “mice” software package and its implementation

4	Multiple imputation of discrete and continuous data by fully conditional specification	Stef van Buuren	The paper gives detailed analysis of managing Imputation of continuous data using JM and FCS approach
5	Substantive model compatible multilevel multiple imputation: A joint modeling approach	M Quartagno, JR Carpenter	The paper investigates a Substantive model compatible multiple imputation (SMC-MI) strategy based on joint modelling of the covariates of the analysis model
6	Multiple imputation in multilevel models. A revision of the current software and usage examples for researchers	P García-Patos, R Olmos	Discusses multi-level hierarchical models using R mitml and mice packages
7	A comparison of joint model and fully conditional specification imputation for multilevel missing data	SA Mistler, CK Enders	The paper examines different multilevel multiple imputation approaches
8	Multiple imputation and ensemble learning for classification with incomplete data	CT Tran, M Zhang, P Andreae, B Xue, LT Bui	It proposes a combination of multiple imputation with ensemble learning for classification with incomplete data.

2.1 **Background on studies**

In statistical databases, multiple imputation is a popular method for filling in missing values. The imputer creates numerous, finished copies of the database by filling in missing values with draws from prediction models calculated from the observed data. Multiple imputation has been implemented by researchers using a number of basic algorithms, but there hasn't been much work assessing how well various techniques perform, especially for categorical data. A fully Bayesian joint distribution based on Dirichlet process mixture models is used to compare the repeated sampling properties of three standard multiple imputation methods for categorical data: chained equations using generalized linear models, chained equations using classification and regression trees, and chained equations using generalized linear models and Dirichlet process mixture models.

2.2 **Chained equations using generalized linear models (GLM)**

The method uses a series of linear equations to impute the missing values in the data. Useful when the relationship between the variables is linear. With this approach, the missing values are imputed stepwise, with one variable being imputed at a time. The generalized linear model upon which the imputation is based permits the inclusion of both continuous and categorical data. In order to make sure that the imputed values are reasonable and accurately reflect the underlying data distribution, the model additionally takes into consideration the correlation between the variables.

When using chained equations, each variable with missing data is iterated through and its value is imputed using the data set's other variables. Until convergence is reached and a final data set with imputed values is obtained, this process is repeated numerous times. In order to create a set of full data sets that can be analyzed using conventional statistical methods, the imputed values are then joined. A versatile and effective approach for multiple imputation, chained equations employing generalized linear models has been demonstrated to yield accurate and dependable results in a variety of applications. As with any imputation technique, it's crucial to carefully consider the model's underlying assumptions and determine how sensitive the outcomes are to various modelling options. This technique has the benefit of being adaptable to a variety of research issues since it can manage missing data in both dependent and independent variables. Furthermore, the generalized linear model framework permits the modelling of various variable types, such as binary, count, and continuous variables, which can be crucial for analyzing complicated data sets.

As with any imputation technique, it's crucial to carefully consider the model's underlying assumptions and determine how sensitive the outcomes are to various modelling options. The accuracy of the imputed values, for instance, might be significantly impacted by the distributional assumptions chosen for the imputation model. In order to evaluate the robustness of the findings, sensitivity analyses should be explored and the potential effects of missing data on the validity of the study results should be taken into account.

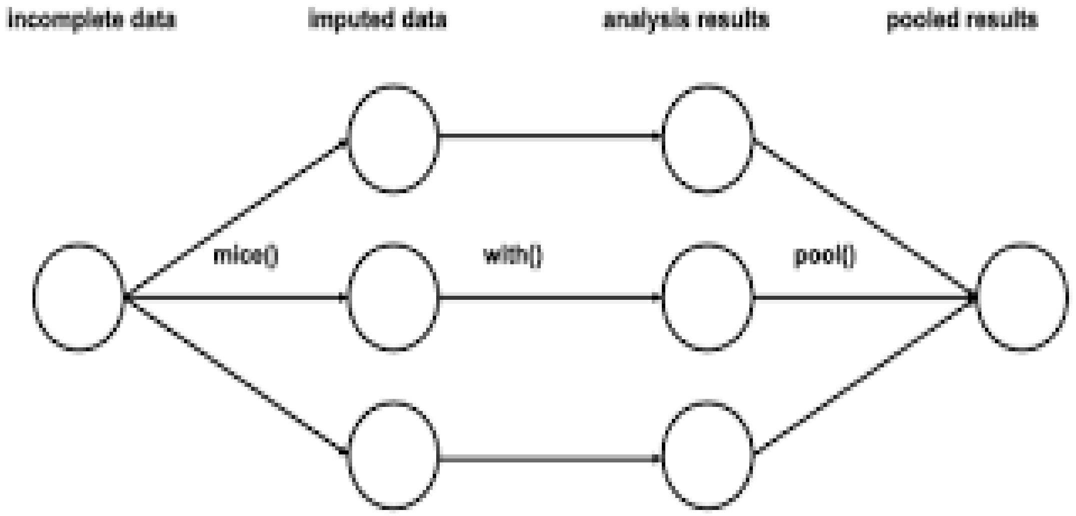


Fig 2.1. Generalized Linear models

(adapted from https://statisticaloddsandends.wordpress.com/2020/07/23/missing-data-and-multivariate-imputation-by-chained-equations-mice/)

2.3 **Chained equations using classification and regression trees (CART)**

The method uses a series of decision trees to impute the missing values in the data. The trees are constructed using the known values in the dataset and the relationships between the variables. With this approach, missing values are imputed using a decision tree-based algorithm that develops a set of if-then-else rules to forecast the missing values. By repeatedly dividing the observed data into smaller and smaller subgroups based on the values of other variables in the data set, the algorithm creates distinct decision trees for each variable with missing data. Once the convergence is attained, the process is repeated for each variable with missing data and the decision rules are used to impute the missing values. A versatile and effective method that can handle continuous and categorical variables as well as complicated interactions and non-linear correlations between variables is the classification and regression tree-based imputation method. It is especially helpful when there are many factors in the data collection, some of which could be highly connected.

As with any imputation technique, it's crucial to carefully consider the model's underlying assumptions and determine how sensitive the outcomes are to various modelling options. Additionally, overfitting can result in biased imputations in decision tree-based methods, which is another drawback. To prevent overfitting, it is crucial to strike a balance between the decision tree's complexity and the quantity of accessible data.

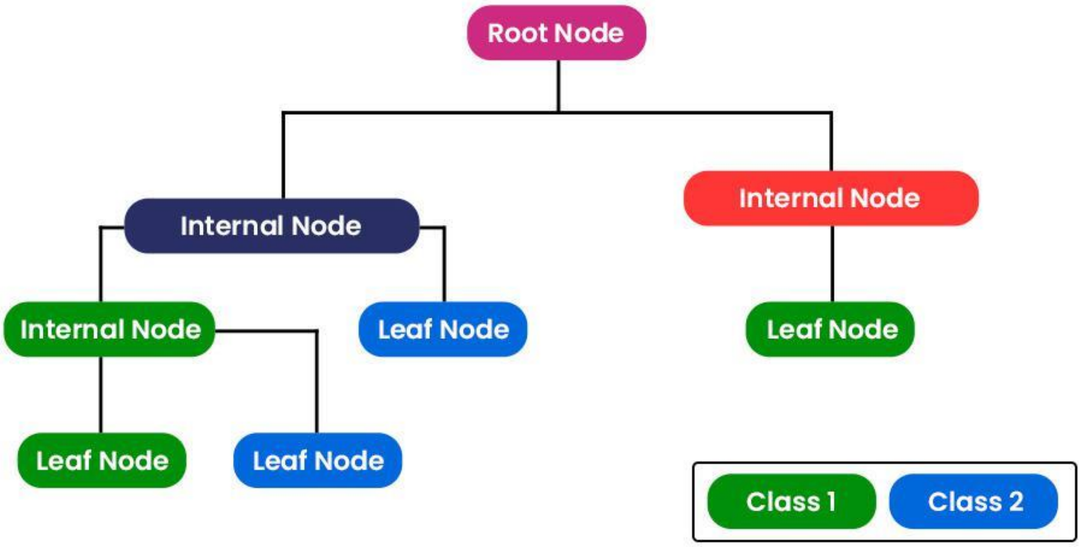


Fig 2.2. Classification and regression trees

(adapted from https://statisticaloddsandends.wordpress.com/2020/07/23/missing-data-and-multivariate-imputation-by-chained-equations-mice/)

2.4 **A fully Bayesian joint distribution based on Dirichlet process mixture models (DPM)**

This method uses a Bayesian model to impute the missing values in the data. This method is useful when the relationship between the variables is complex. Using a Dirichlet process mixing model, this technique generates a joint distribution for the observed data and the missing data. The mixture model, which supports the inclusion of both continuous and categorical variables, is a versatile and effective method for modelling large data sets. The model is capable of handling intricate interactions between variables as well as linear and non-linear correlations between variables.

When dealing with complex data sets with numerous variables and missing data, the Dirichlet process mixture model also allows for uncertainty in the number of clusters in the data set. A previous distribution that is modified depending on the observed data determines the number of clusters. For each missing observation, the joint distribution is employed to produce a variety of reasonable values. By selecting samples from the posterior distribution of the missing data given the observed data, the imputed values are generated. The posterior distribution gives an indication of the imputation uncertainty and captures the uncertainty in the missing data.

$$(\mathrm{\lambda kj1}, \ldots, \mathrm{\lambda kjDj}) \sim \mathrm{Dirichlet(1Dj)}$$

Eq 2.1

$$\mathrm{\pi k} = Vk \prod_{h<k} (1 - \mathrm{Vh})$$

Eq 2.2

$$\mathrm{Vk} \sim \mathrm{Beta}(1, \alpha)\ \mathrm{for}$$

Eq 2.3

$$\mathrm{k} = 1, \ldots, \mathrm{K} - 1; \mathrm{VK} = 1$$

Eq 2.4

$$\alpha \sim \mathrm{Gamma}(0.25, 0.25)$$

Eq 2.5

The fact that this approach offers a consistent framework for managing missing data and enables the integration of all information included in the data set, including observed data and previous knowledge about the data generation process, is one of its advantages. Additionally, it offers a measurement of the imputed values' level of uncertainty, which is useful when drawing inferences and conclusions from the data.

The completely Bayesian joint distribution based on Dirichlet process mixing models, on the other hand, can be computationally demanding and may necessitate a considerable quantity of data to get reliable findings. The model's underlying assumptions must also be carefully considered, and the sensitivity of the results to various modelling options must be evaluated.

For addressing missing data, multiple imputation utilising a fully Bayesian joint distribution based on Dirichlet process mixing models is a versatile and effective technique. Analysis of complicated data sets with several variables and missing data may make use of it particularly well. The underlying assumptions must be carefully considered, as with any imputation approach, and the results' sensitivity to various modelling options must be evaluated.

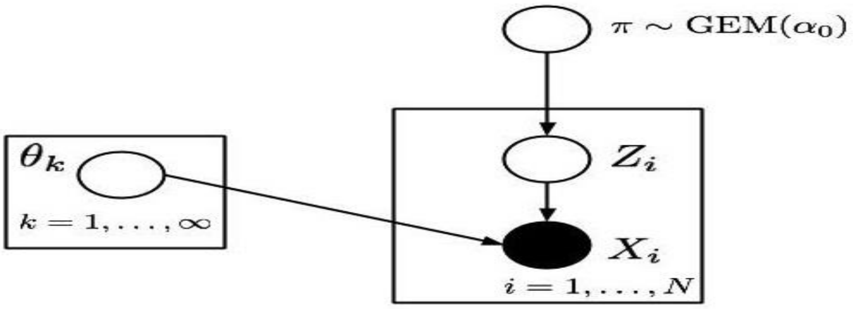


Fig 2.3. Dirichlet process mixture models

(adapted from https://blog.datumbox.com/the-dirichlet-process-mixture-model/)

2.5 **Ensemble Learning method**

A machine learning approach called ensemble learning combines different models to increase the reliability and accuracy of predictions. The fundamental tenet of ensemble learning is that a collection of models may outperform any one model alone. The goal of the ensemble technique is to combine weak models to get powerful findings. Either a single improper learning algorithm (homogeneous model) or many improper learning algorithms (heterogeneous model) should be used in the combination. EL model learning is a broad topic that is condensed by the user's concepts in any given sector. EL is broken down into many sorts of approach, such as boosting, bagging, and stacking.

2.5.1 Bagging

Bagging is the most straightforward yet effective method combining bootstrap with aggregating is known as bootstrap aggregating. Several subsamples are gathered and drawn using the bootstrap method. It is a technique that involves building multiple independent models and combining their predictions to create a final prediction that is more robust and accurate than the predictions of individual models. All of the bootstrapped subsamples must be conducted using the decision tree. The decision tree outputs will finally be aggregated using aggregate.

A number of bootstrap samples are created by randomly sampling the training data throughout the bagging process. A different model is trained for each bootstrap sample, and these models are then integrated to get an overall forecast. The average or majority vote of each individual model forecast is often used to calculate the final prediction.

The risk of overfitting, a major issue in machine learning where a model grows too sophisticated and starts to perform badly on fresh data, is one of the advantages of bagging. Bagging can produce a more consistent and precise forecast by averaging out the noise and volatility in the data through the use of numerous models.

Bagging also has the benefit of being adaptable to many machine learning techniques, such as regression models, decision trees, and neural networks. Bagging can also be applied to classification and regression issues.

Bagging is a potent method for enhancing the stability and accuracy of machine learning models overall. Several applications, including speech recognition, image recognition, and natural language processing, have successfully used it.

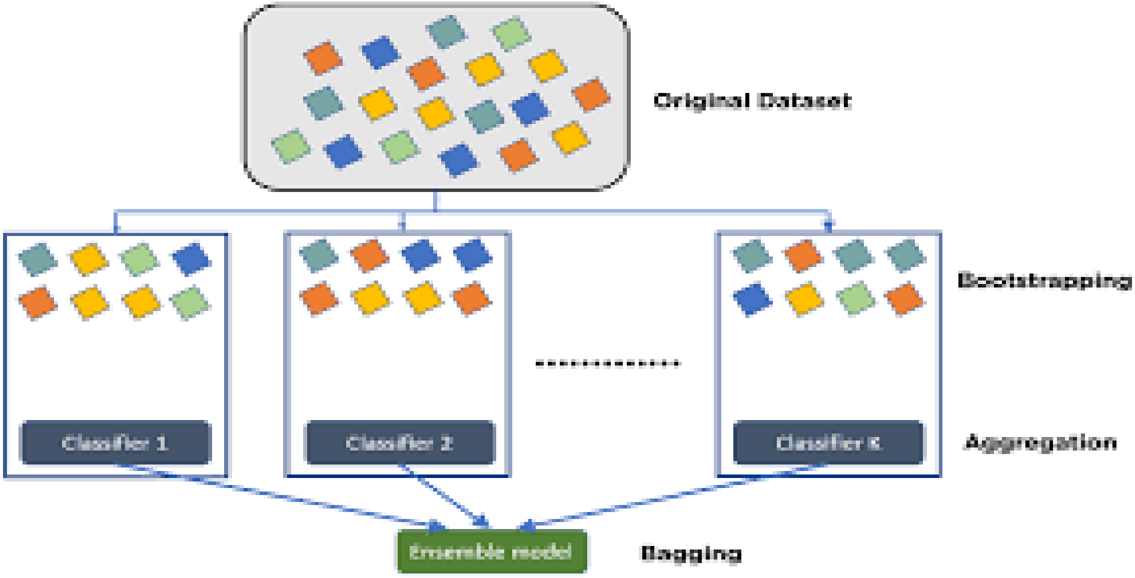


Fig 2.4. Bagging method flowchart

(adapted from https://www.simplilearn.com/tutorials/machine-learning-tutorial/bagging-in-machine-learning)

2.5.2 Boosting

Another popular ensemble learning technique for raising the precision of computer learning models is boosting. A number of weak models are consecutively trained in Boosting, as opposed to Bagging, and each new model is taught to fix the flaws in the preceding models.

An initial weak model is trained on the complete training set as the first step in the boosting process. Any machine learning technique that can provide predictions that are more accurate than random is considered to be a weak model. After the initial model has been trained, the data is reweighted so that the examples that were incorrectly classified are given a higher weight and the examples that were correctly classified are given a lower weight.

The procedure is then continued until a set number of models have been trained or the accuracy on the training data reaches a plateau, after which a new weak model is trained on the reweighted data. The final forecast is calculated as a weighted average of the predictions from each individual model, with the weights given according to the precision of each model.

One of Boosting's main benefits is that it may greatly increase the accuracy of weak models, often yielding outcomes that are superior to even the finest individual models. Additionally adaptable, boosting can be used with a variety of machine learning algorithms, including regression models, neural networks, and decision trees.

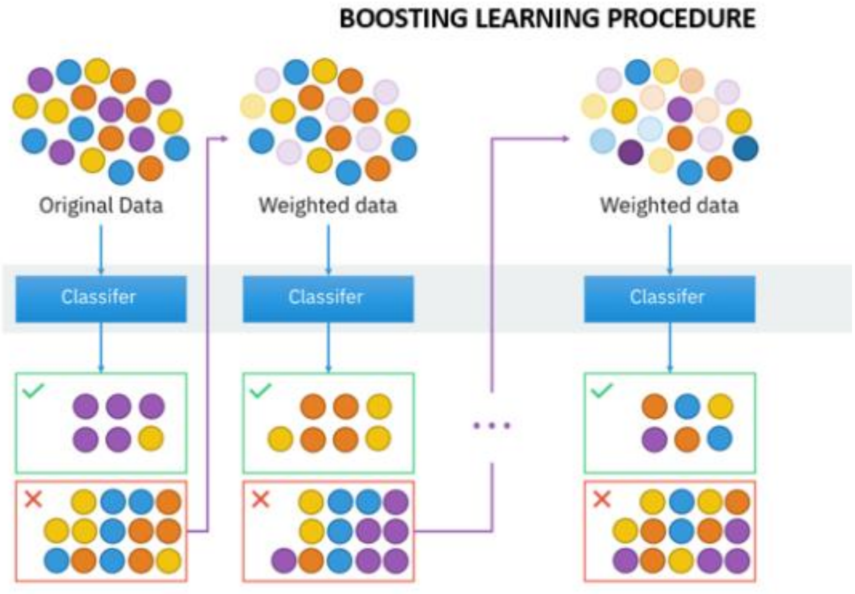


Fig 2.5. Boosting ensemble learning

(adapted from https://www.simplilearn.com/tutorials/machine-learning-tutorial/boosting-in-machine-learning)

The process of boosting is iterative and involves some weight. Reweighting the weight should be done after each iteration. All of the findings are then evaluated together, and a decision is made using a majority voting system. The boosting algorithms come in two varieties.

- The AdaBoosing method - AdaBoost is a specific form of Boosting, where the algorithm adjusts the weights of the data points in each iteration to focus on the incorrectly classified samples. It is commonly used for binary classification problems but can be extended to multiclass problems as well. It operates on a statistical (numerical) foundation sequentially training a series of weak models, such as decision trees or linear classifiers, on the training data. In each iteration, the algorithm assigns higher weights to the incorrectly classified data points and lower weights to the correctly classified ones. The goal is to give more emphasis to the hard-to-classify data points, which should improve the accuracy of the subsequent models.

- The gradient boosting approach – This method uses regression as its foundation. Gradient Boosting uses a learning rate, also known as the shrinkage parameter, to control the contribution of each tree to the final prediction. A smaller learning rate reduces the contribution of each tree, which can help prevent overfitting and improve the generalization performance of the model. However, Gradient Boosting is computationally intensive and requires careful tuning of hyperparameters, such as the learning rate, the number of trees, and the depth of each tree.

2.5.3 Stacking

The meta model algorithm is stacking. The hierarchical paradigm for implementing learner dependence is fascinating. Training a collection of base models on the training data is the first step in the stacking process. Any machine learning method capable of making precise predictions can be used as the basic model. Following training, the base models are used to make predictions on a validation set that wasn't used during base model training.

Following that, a meta-model is trained to identify the correlation between the predictions from the base models and the actual target variable using the predictions from the base models as its input. The validation set is used to train the meta-model, which can be any machine learning approach, such a neural network or a linear regression model. The learner draws inspiration from the various learner outputs from earlier in this lesson. It results in a decrease in bias and variant error. Level 0 model, for instance, is a fundamental model that predicts and compiles the training data. Level 1 of a sequential model is improving on the preceding level's prediction integration.

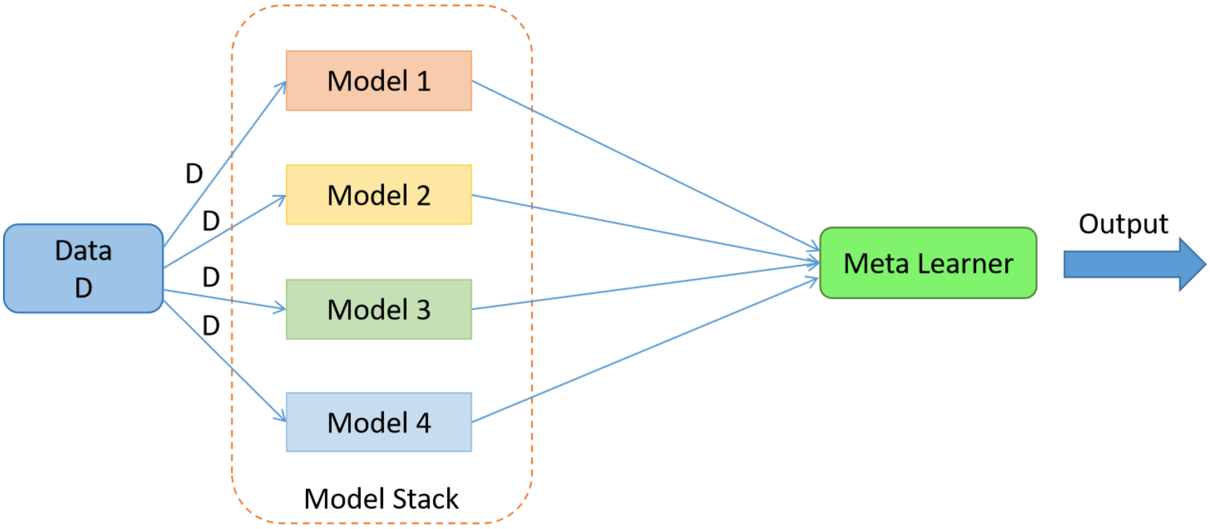


Fig 2.6. Method of Stacking

(adapted from https://www.simplilearn.com/tutorials/machine-learning-tutorial/stacking-in-machine-learning)

Once trained, the meta-model can be used to predict outcomes based on fresh data. The meta-model receives the predictions from the basis models and outputs the final forecast.

One of the main benefits of stacking is that it may combine the capabilities of many models, leading to outcomes that are superior than those of any one model. Stacking is adaptable and may be used with a variety of machine learning techniques.

But when dealing with a large number of base models or a complicated meta-model, stacking can be computationally demanding. Additionally, Stacking can be challenging in some applications because it needs a lot of data to train both the base models and the meta-model.

2.6 **Research Gaps**

- All these methods are only built on the working model predicting the missing values. An apparent disadvantage is that they are sensitive to certain misspecifications of the working model.
- The accuracy of the traditional methods is quite less when the percentage of missing data is high.
- The performance of different imputation methods may depend on the specific characteristics of the data.

Chapter3 **Methodology**

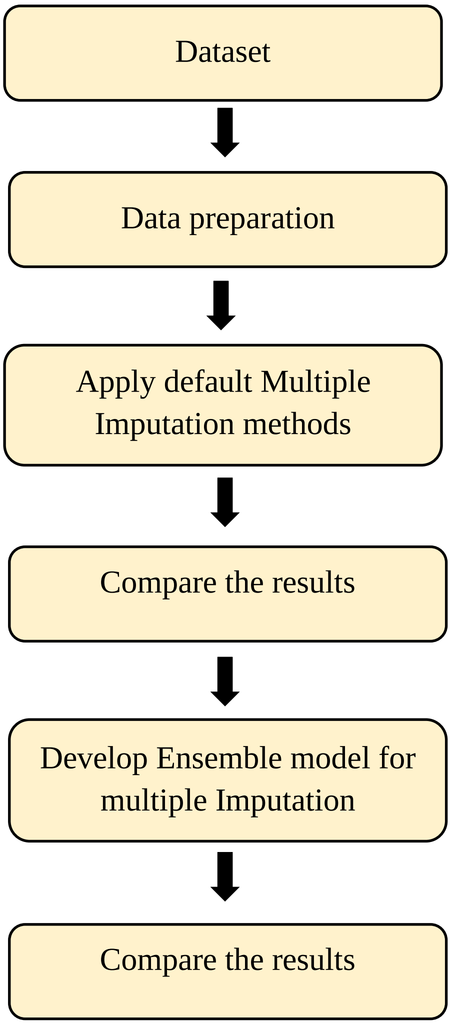
In this paper, we focus only on the categorical data.

3.1 **Multiple Imputation for Categorical Data**

Description of various notation for approaching MI in the context of categorical data. Let $Y_{ij} \in \{1,..., D_j\}$ be the value of variable j for individual i, where j = 1,..., p and i = 1,..., n. For each individual i, let $Y_i = (Y_{i1},...,Y_{ip})$. Let $Y = (Y_1,..., Y_n)$ be the n × p matrix comprising the data for all records included in the sample. We write $Y = (Y_{obs}, Y_{mis})$, where Y_{obs} and Y_{mis} are respectively the observed and missing parts of Y. We write $Y_{mis} = (Y_{mis1},..., Y_{misp})$, where Y_{misj} represents all missing values for variable j, where j = 1,..., p. Similarly, we write $Y_{obs} = (Y_{obs1},..., Y_{obsp})$ for the corresponding observed data. In MI, the analyst generates values of Y_{mis} using models estimated with Y_{obs}. This results in a completed dataset Y.

3.2 **Methodology Flowchart**

The methodology followed in case of the given dataset is shown in the following figure**.


Dataset
Data preparation
Apply default Multiple Imputation methods
Compare the results
Develop Ensemble model for multiple Imputation
Compare the results

3.3 **Why Ensemble learning method**

Ensemble learning methods are useful in missing data imputation because they combine the outputs of multiple models to produce a single, more accurate result. While traditional methods like multiple imputation can provide good results in many cases, they can still produce biased or incomplete results if the model used for imputation is not appropriate for the data or if there is a high degree of missingness in the dataset. Ensemble methods, on the other hand, can overcome some of these limitations by combining multiple models to provide more robust and accurate imputations. For example, if one model performs poorly for a particular subset of the data, the ensemble can compensate by relying more heavily on the other models. ensemble methods can be particularly useful because they can capture more complex relationships between variables than traditional methods.

Increased Accuracy: By integrating the strengths of various models, ensemble learning can increase the accuracy of predictions.

Robustness: By minimizing the impact of outliers or data noise, ensemble learning can enhance the robustness of predictions.

Generalization: By lowering overfitting to the training data, ensemble learning can enhance the generalization of predictions.

Diversity: By combining models with diverse strengths and weaknesses, ensemble learning can increase the diversity of predictions.

3.4 **MaxVoting**

For the implementation of Ensemble method, we used a MaxVoting Ensemble method.

MaxVoting is an ensemble method used in machine learning, where multiple models are trained to solve a problem, and the final prediction is made based on the majority vote of the individual models. This method is simple and widely used due to its effectiveness in improving the accuracy of predictions. The MaxVoting ensemble method works by combining the predictions of multiple models trained on the same dataset. Each individual model is trained using a subset of the training data or a different algorithm, resulting in diverse models with different biases and strengths. The ensemble method combines the predictions of these models by selecting the most frequent prediction as the final output.

In the case of classification problems, the MaxVoting ensemble method can be applied to the predicted class labels of each individual model. For example, if there are five models, and four of them predict the class label A, while one predicts class label B, the MaxVoting ensemble method will select class label A as the final output. In the case of regression problems, the MaxVoting ensemble method can be applied to the predicted values of each individual model. The final output is the average of the predicted values.

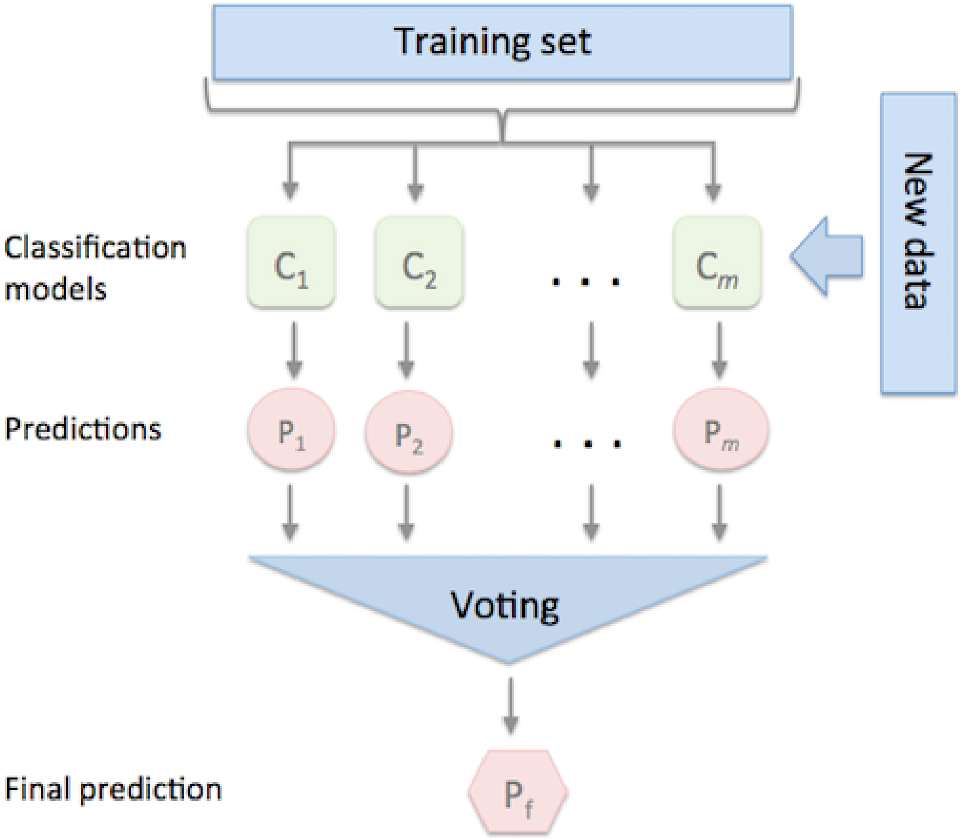


Fig 3.1. MaxVoting Ensemble algorithm flowchart

(adapted from https://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/)

The MaxVoting ensemble method is useful when there is a high degree of variance in the predictions of individual models, which can happen due to the noise in the training data or due to the limitations of the algorithm used. By combining the predictions of multiple models, the variance is reduced, and the accuracy of the predictions is improved. However, the MaxVoting ensemble method may not be effective if the individual models are highly correlated, which can happen if the same algorithm is used to train all the models or if the same set of features are used. In such cases, the ensemble method may not result in a significant improvement in accuracy.

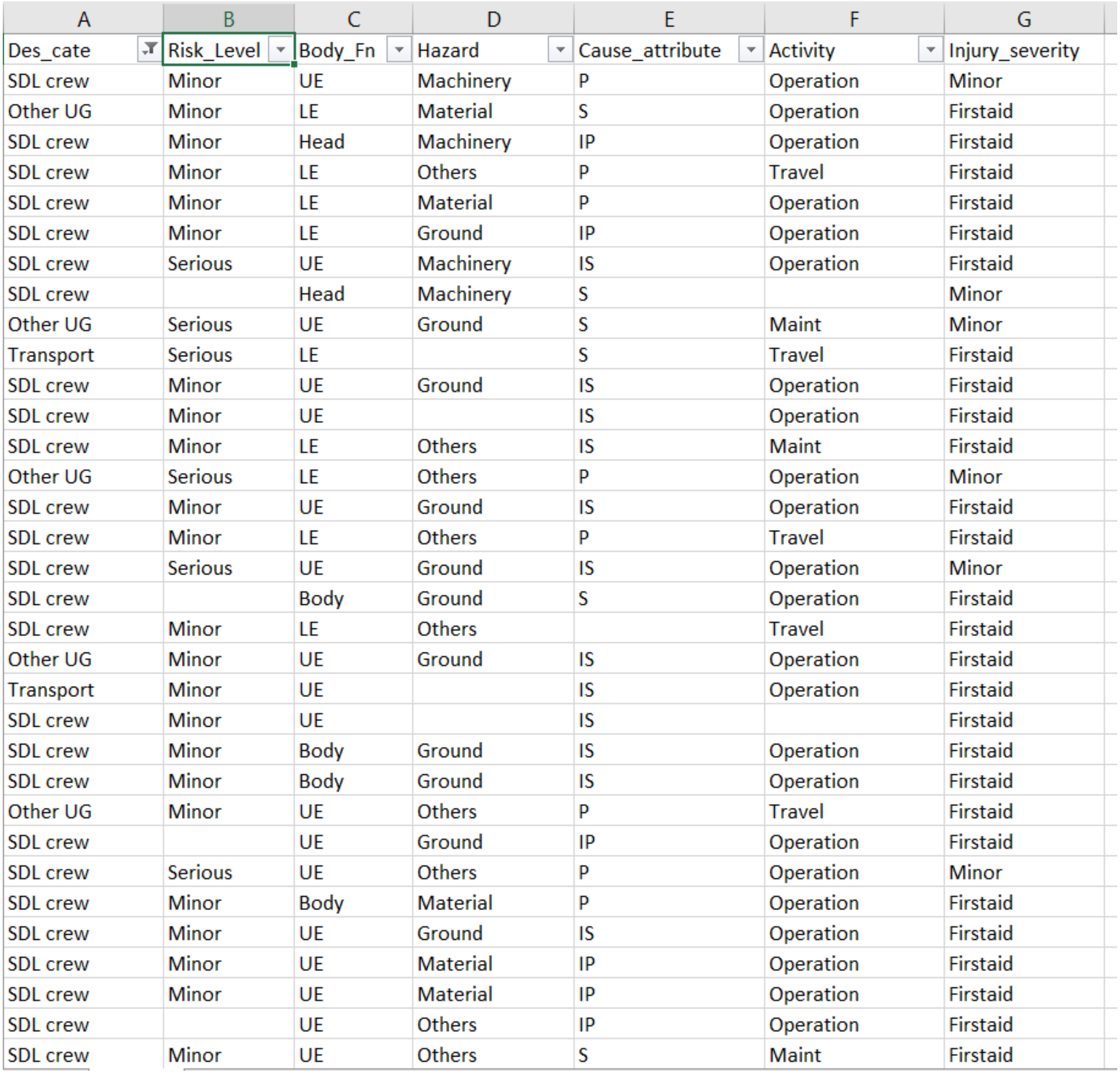
In conclusion, the MaxVoting ensemble method is a simple and effective way to improve the accuracy of predictions in machine learning. It works by combining the predictions of multiple models trained on the same dataset, and selecting the most frequent prediction as the final output. However, care must be taken to ensure that the individual models are diverse, and not highly correlated, for the ensemble method to be effective.

Chapter4 **Case Study**

4.1 **Data description**

Source: The data has been collected from a mine field in Eastern India.

The dataset describes the Injury severity of a worker in a mining field, i.e., what are the various factors which could lead to an injury of a worker in a mining field. The dataset consists of 7 columns and 500 observations of categorical data.



A	B	C	D	E	F	G
Des_cate	Risk_Level	Body_Fn	Hazard	Cause_attribute	Activity	Injury_severity
SDL crew	Minor	UE	Machinery	P	Operation	Minor
Other UG	Minor	LE	Material	S	Operation	Firstaid
SDL crew	Minor	Head	Machinery	IP	Operation	Firstaid
SDL crew	Minor	LE	Others	P	Travel	Firstaid
SDL crew	Minor	LE	Material	P	Operation	Firstaid
SDL crew	Minor	LE	Ground	IP	Operation	Firstaid
SDL crew	Serious	UE	Machinery	IS	Operation	Firstaid
SDL crew		Head	Machinery	S		Minor
Other UG	Serious	UE	Ground	S	Maint	Minor
Transport	Serious	LE		S	Travel	Firstaid
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
SDL crew	Minor	UE		IS	Operation	Firstaid
SDL crew	Minor	LE	Others	IS	Maint	Firstaid
Other UG	Serious	LE	Others	P	Operation	Minor
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
SDL crew	Minor	LE	Others	P	Travel	Firstaid
SDL crew	Serious	UE	Ground	IS	Operation	Minor
SDL crew		Body	Ground	S	Operation	Firstaid
SDL crew	Minor	LE	Others		Travel	Firstaid
Other UG	Minor	UE	Ground	IS	Operation	Firstaid
Transport	Minor	UE		IS	Operation	Firstaid
SDL crew	Minor	UE		IS		Firstaid
SDL crew	Minor	Body	Ground	IS	Operation	Firstaid
SDL crew	Minor	Body	Ground	IS	Operation	Firstaid
Other UG	Minor	UE	Others	P	Travel	Firstaid
SDL crew		UE	Ground	IP	Operation	Firstaid
SDL crew	Serious	UE	Others	P	Operation	Minor
SDL crew	Minor	Body	Material	P	Operation	Firstaid
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
SDL crew	Minor	UE	Material	IP	Operation	Firstaid
SDL crew	Minor	UE	Material	IP	Operation	Firstaid
SDL crew		UE	Others	IP	Operation	Firstaid
SDL crew	Minor	UE	Others	S	Maint	Firstaid

Fig 4.1. Actual Incomplete dataset

The various categories of different columns are given below:

Des_cate: SDL crew, Other UG, Miner, Engineering, Transport, Supervisor

Risk_Level: Minor, Serious, Fatal

Body_Fn: UE, LE, Head, Body, Nil, MBP

Hazard: Machinery, Material, Others, Ground

Cause_attribute: P, S, IP, IS

Activity: Operation, Travel, Maint

Injury_severity: Minor, Firstaid, Nearmiss

The figure shown describes the frequency of the missing data in the dataset. The figure denotes that there are 370 rows with no missing data, while other rows have missing data in one/multiple columns. The following attributes stands for-

IS – Injury Severity (0 missing value)

BF – Body function (1 missing value)

H – Hazard (17 missing values)

A – Activity (18 missing values)

CA – Cause Attribute (23 missing values)

RL – Risk Level (39 missing values)

D – Description (41 missing values)

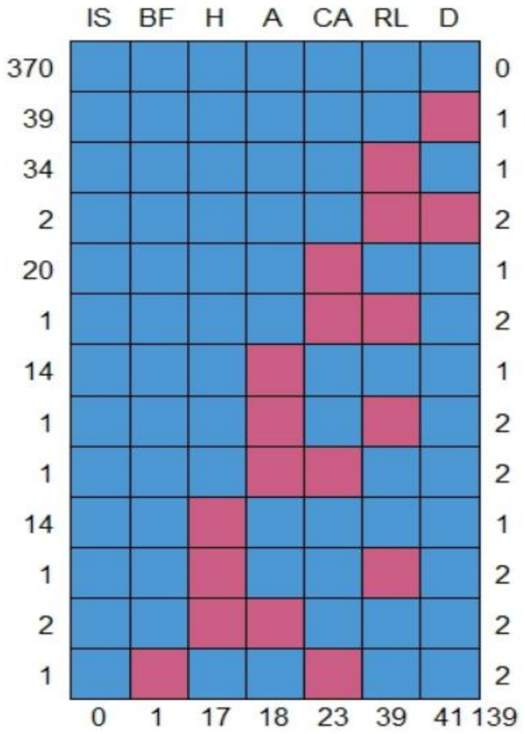


Fig 4.1. Pattern of missing data in the dataset

4.2 **Data preparation**

The data was categorical so we had to perform Encoding to implement any function. There are several encoding methods, including:

- One-Hot Encoding: This method creates a binary column for each unique value of the categorical variable. If a record has a certain value, the corresponding column will have a value of 1, while all other columns will have a value of 0.
- Label Encoding: This method assigns a numerical value to each unique value of the categorical variable. The values are assigned in an arbitrary manner, based on the order in which the values appear in the dataset.

- Ordinal Encoding: This method assigns a numerical value to each unique value of the categorical variable based on their order or rank. For example, the values "low", "medium", and "high" could be encoded as 1, 2, and 3, respectively.
- Count Encoding: This method replaces each unique value of the categorical variable with the number of times it appears in the dataset.
- Target Encoding: This method replaces each unique value of the categorical variable with the mean or median of the target variable for records that have that value.
- Binary Encoding: This method creates binary columns for each unique value of the categorical variable.

To keep the dataset simple, we perform Label Encoding where each unique value of the categorical variable is assigned a numerical value, based on the order in which the values appear in the dataset.

Code to perform label encoding is shown in Appendix Fig I.1.

4.3 **MI-GLM Implementation**

For categorical data, most implementations of MI-GLM use logistic regressions for binary variables, or multinomial logistic regressions for unordered variables, or cumulative logistic regressions for ordered variables.

4.3.1 Internal algorithm involved

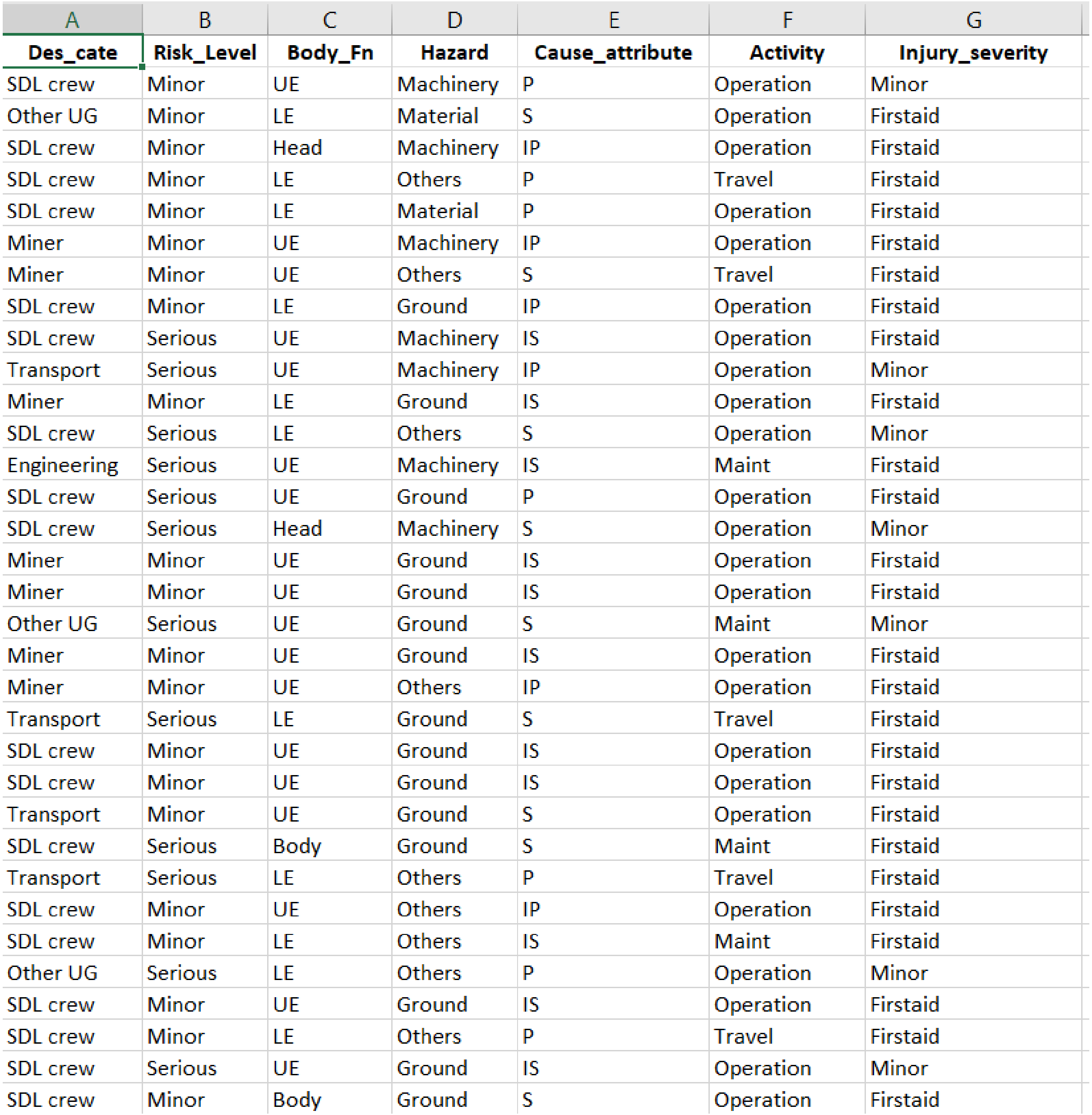
With this initial set of completed data, we use an iterative process akin to a Gibbs sampler to update the imputations. At each iteration t of the updating, we estimate the predictive model, $(Y_{(1)} \mid Y_{obs(1)},\{Y\}^{(t-1)}_{(k)} : k > 1\})$, where $Y^{(t-1)}_{(k)}$ includes the set of observed and imputed values for variable k at iteration t − 1. We replace $Y^{(t-1)}_{mis(1)}$ with draws from this conditional distribution, $Y^{(t)}_{mis(1)}$. We repeat this process for each variable j with missing data, estimating and imputing from each predictive model, $(Y_{(j)} \mid Y_{obs(j)},\{Y^{(t)}_{(k)} : k < j\},\{Y^{(t-1)}_{(k)} : k > j\})$. We repeat the cycle t > 1 times. The values at the final iteration are used to create the completed dataset, $Y = (Y_{obs}, Y_{mis})$. The entire process is replicated number of rows times to create the full set of multiple imputations.

Multiple imputation using GLMs is performed using software package MICE (Multivariate Imputation by Chained Equations) with “polyreg” univariate imputation method.

Code to perform MI-GLM is shown in Appendix Fig I.2.

4.3.2 Data View

The glimpse of the complete dataset output through this method is shown below-



A	B	C	D	E	F	G
Des_cate	**Risk_Level**	**Body_Fn**	**Hazard**	**Cause_attribute**	**Activity**	**Injury_severity**
SDL crew	Minor	UE	Machinery	P	Operation	Minor
Other UG	Minor	LE	Material	S	Operation	Firstaid
SDL crew	Minor	Head	Machinery	IP	Operation	Firstaid
SDL crew	Minor	LE	Others	P	Travel	Firstaid
SDL crew	Minor	LE	Material	P	Operation	Firstaid
Miner	Minor	UE	Machinery	IP	Operation	Firstaid
Miner	Minor	UE	Others	S	Travel	Firstaid
SDL crew	Minor	LE	Ground	IP	Operation	Firstaid
SDL crew	Serious	UE	Machinery	IS	Operation	Firstaid
Transport	Serious	UE	Machinery	IP	Operation	Minor
Miner	Minor	LE	Ground	IS	Operation	Firstaid
SDL crew	Serious	LE	Others	S	Operation	Minor
Engineering	Serious	UE	Machinery	IS	Maint	Firstaid
SDL crew	Serious	UE	Ground	P	Operation	Firstaid
SDL crew	Serious	Head	Machinery	S	Operation	Minor
Miner	Minor	UE	Ground	IS	Operation	Firstaid
Miner	Minor	UE	Ground	IS	Operation	Firstaid
Other UG	Serious	UE	Ground	S	Maint	Minor
Miner	Minor	UE	Ground	IS	Operation	Firstaid
Miner	Minor	UE	Others	IP	Operation	Firstaid
Transport	Serious	LE	Ground	S	Travel	Firstaid
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
Transport	Minor	UE	Ground	S	Operation	Firstaid
SDL crew	Serious	Body	Ground	S	Maint	Firstaid
Transport	Serious	LE	Others	P	Travel	Firstaid
SDL crew	Minor	UE	Others	IP	Operation	Firstaid
SDL crew	Minor	LE	Others	IS	Maint	Firstaid
Other UG	Serious	LE	Others	P	Operation	Minor
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
SDL crew	Minor	LE	Others	P	Travel	Firstaid
SDL crew	Serious	UE	Ground	IS	Operation	Minor
SDL crew	Minor	Body	Ground	S	Operation	Firstaid

Fig 4.2. Complete dataset using GLM

4.4 MI-CART Implementation

MI-CART operates like MI-GLM, except that CART models are used in place of logistic regressions. Classification trees are used for categorical data.

4.4.1 Internal algorithm involved

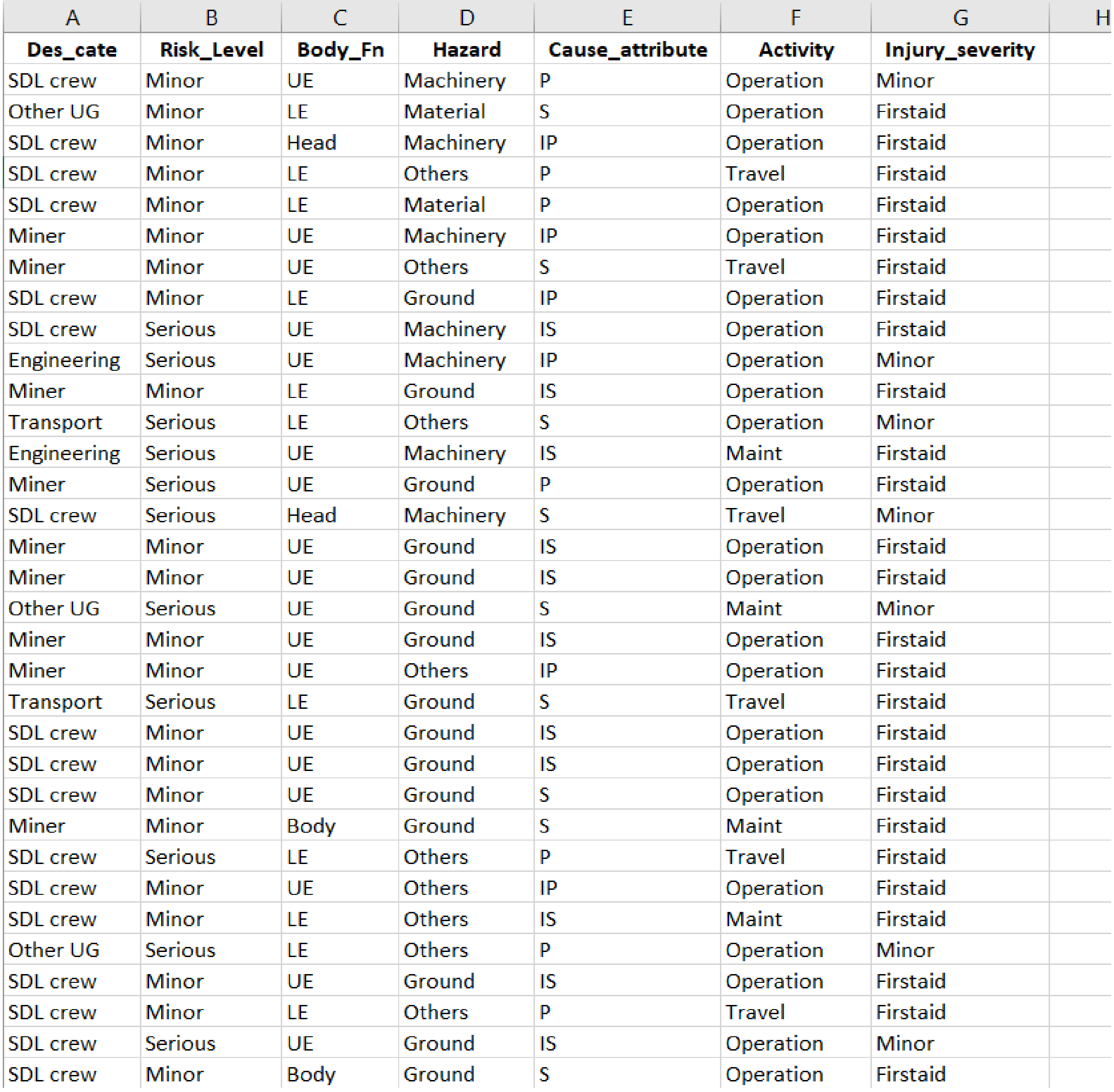
It uses recursive partitioning of the predictor space to create a tree structure, where each leaf represents a subset of units with similar predictor values. The values of the outcome variable in each leaf are considered as draws from the conditional distribution of the outcome variable for that subset of predictor values. To generate initial imputations, CART models are trained on the available cases for each outcome variable, conditional on the other observed variables. The imputations are obtained by dropping down the tree for the outcome variable until finding the appropriate leaf, and then sampling from the values in that leaf.

Multiple imputation using CART is performed using software package MICE with "cart" univariate imputation method.

Code to perform MI-CART is shown in Appendix Fig I.3.

4.4.2 Data View

The glimpse of the complete dataset output through this method is shown below-



A	B	C	D	E	F	G	H
Des_cate	**Risk_Level**	**Body_Fn**	**Hazard**	**Cause_attribute**	**Activity**	**Injury_severity**	
SDL crew	Minor	UE	Machinery	P	Operation	Minor	
Other UG	Minor	LE	Material	S	Operation	Firstaid	
SDL crew	Minor	Head	Machinery	IP	Operation	Firstaid	
SDL crew	Minor	LE	Others	P	Travel	Firstaid	
SDL crew	Minor	LE	Material	P	Operation	Firstaid	
Miner	Minor	UE	Machinery	IP	Operation	Firstaid	
Miner	Minor	UE	Others	S	Travel	Firstaid	
SDL crew	Minor	LE	Ground	IP	Operation	Firstaid	
SDL crew	Serious	UE	Machinery	IS	Operation	Firstaid	
Engineering	Serious	UE	Machinery	IP	Operation	Minor	
Miner	Minor	LE	Ground	IS	Operation	Firstaid	
Transport	Serious	LE	Others	S	Operation	Minor	
Engineering	Serious	UE	Machinery	IS	Maint	Firstaid	
Miner	Serious	UE	Ground	P	Operation	Firstaid	
SDL crew	Serious	Head	Machinery	S	Travel	Minor	
Miner	Minor	UE	Ground	IS	Operation	Firstaid	
Miner	Minor	UE	Ground	IS	Operation	Firstaid	
Other UG	Serious	UE	Ground	S	Maint	Minor	
Miner	Minor	UE	Ground	IS	Operation	Firstaid	
Miner	Minor	UE	Others	IP	Operation	Firstaid	
Transport	Serious	LE	Ground	S	Travel	Firstaid	
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid	
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid	
SDL crew	Minor	UE	Ground	S	Operation	Firstaid	
Miner	Minor	Body	Ground	S	Maint	Firstaid	
SDL crew	Serious	LE	Others	P	Travel	Firstaid	
SDL crew	Minor	UE	Others	IP	Operation	Firstaid	
SDL crew	Minor	LE	Others	IS	Maint	Firstaid	
Other UG	Serious	LE	Others	P	Operation	Minor	
SDL crew	Minor	UE	Ground	IS	Operation	Firstaid	
SDL crew	Minor	LE	Others	P	Travel	Firstaid	
SDL crew	Serious	UE	Ground	IS	Operation	Minor	
SDL crew	Minor	Body	Ground	S	Operation	Firstaid	

Fig 4.3. Complete dataset using CART

4.5 **MI-DPM Implementation**

The MI-DPM procedure assumes that the distribution of the categorical data can be characterized using a latent class model.

4.5.1 Internal algorithm involved

This method assumes that all possible combinations of variables are possible a priori, meaning there are no structural zeros. Each individual is assigned to one of the K latent classes. Within each class, all variables are assumed to follow independent multinomial distributions.

To express this as a formal probability model, we let $z_i \in (1,..., K)$ represent the latent class of individual i. We define π_k as the probability of an individual belonging to latent class k, for k = 1,..., K. We also define λ_{kjy} as the probability that variable Yi j takes on the value y for records in latent class k.

We further define π as a vector of probabilities $(\pi_1,...,\pi_K)$, and λ as a set of conditional probabilities $\{\lambda_{kjy} : k = 1,..., K; j = 1,..., p; y = 1,..., Dj\}$. These probabilities are used to impute the missing data, where the imputations are obtained by sampling from the posterior predictive distribution of the missing data given the observed data and the estimated parameters of the model.

We can formulate the marginal probability for any quantity by averaging over the latent classes, for example –

$$\Pr(\mathrm{Yi1} = \mathrm{y1}, \ldots, \mathrm{Yip} = \mathrm{yp} \mid \lambda, \pi) = \sum_{k=1}^{n} \pi\mathrm{k} \prod_{j=1}^{p} \lambda\mathrm{kjy}$$

Eq 4.1

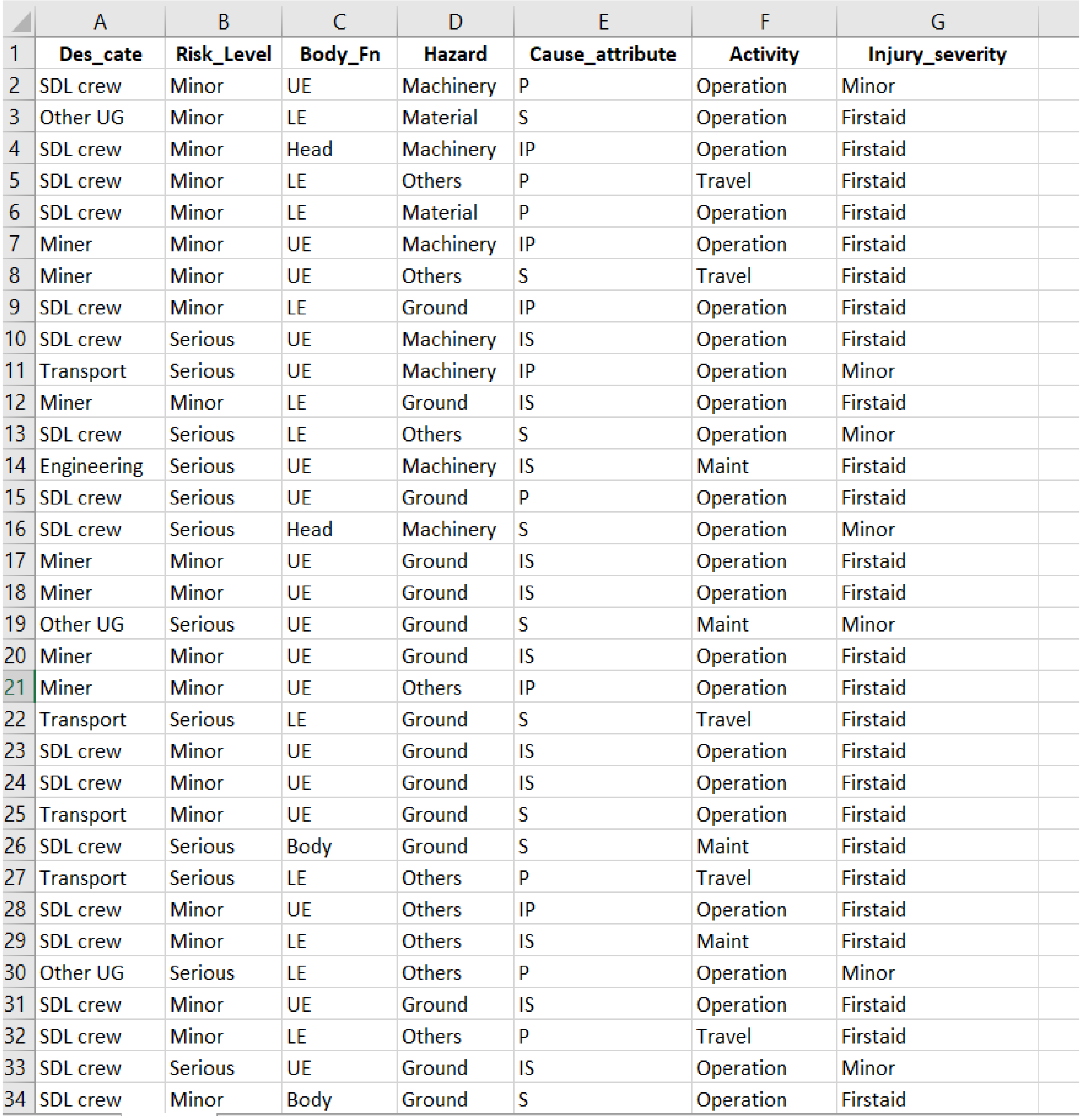
Using a Gibbs sampler, the posterior inferences are derived. Missing values are handled within the sampler by the Gibbs sampler. The Gibbs sampler initially samples a value of the latent class indicator for each record using the equation from the previous section, given a draw of the parameters and observed data. Values for the missing elements are then sampled using independent draws once the latent class indicator has been sampled. In datasets with numerous categorical variables, computation and imputation are made simpler by the independence of variables within the latent classes. We can sample values for the missing data using separate draws since the conditional independence assumption enables us to determine the posterior probabilities of each variable given the latent class.

To implement the DPMPM can be performed using we use the "NPBayesImpute" package in R. (NPBayesImpute: Non-Parametric Bayesian Multiple Imputation for Categorical Data)

Code to perform MI-DPM is shown in Appendix Fig I.4

4.5.2 Data View

The glimpse of the complete dataset output through this method is shown below-



	A	B	C	D	E	F	G
1	**Des_cate**	**Risk_Level**	**Body_Fn**	**Hazard**	**Cause_attribute**	**Activity**	**Injury_severity**
2	SDL crew	Minor	UE	Machinery	P	Operation	Minor
3	Other UG	Minor	LE	Material	S	Operation	Firstaid
4	SDL crew	Minor	Head	Machinery	IP	Operation	Firstaid
5	SDL crew	Minor	LE	Others	P	Travel	Firstaid
6	SDL crew	Minor	LE	Material	P	Operation	Firstaid
7	Miner	Minor	UE	Machinery	IP	Operation	Firstaid
8	Miner	Minor	UE	Others	S	Travel	Firstaid
9	SDL crew	Minor	LE	Ground	IP	Operation	Firstaid
10	SDL crew	Serious	UE	Machinery	IS	Operation	Firstaid
11	Transport	Serious	UE	Machinery	IP	Operation	Minor
12	Miner	Minor	LE	Ground	IS	Operation	Firstaid
13	SDL crew	Serious	LE	Others	S	Operation	Minor
14	Engineering	Serious	UE	Machinery	IS	Maint	Firstaid
15	SDL crew	Serious	UE	Ground	P	Operation	Firstaid
16	SDL crew	Serious	Head	Machinery	S	Operation	Minor
17	Miner	Minor	UE	Ground	IS	Operation	Firstaid
18	Miner	Minor	UE	Ground	IS	Operation	Firstaid
19	Other UG	Serious	UE	Ground	S	Maint	Minor
20	Miner	Minor	UE	Ground	IS	Operation	Firstaid
21	Miner	Minor	UE	Others	IP	Operation	Firstaid
22	Transport	Serious	LE	Ground	S	Travel	Firstaid
23	SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
24	SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
25	Transport	Minor	UE	Ground	S	Operation	Firstaid
26	SDL crew	Serious	Body	Ground	S	Maint	Firstaid
27	Transport	Serious	LE	Others	P	Travel	Firstaid
28	SDL crew	Minor	UE	Others	IP	Operation	Firstaid
29	SDL crew	Minor	LE	Others	IS	Maint	Firstaid
30	Other UG	Serious	LE	Others	P	Operation	Minor
31	SDL crew	Minor	UE	Ground	IS	Operation	Firstaid
32	SDL crew	Minor	LE	Others	P	Travel	Firstaid
33	SDL crew	Serious	UE	Ground	IS	Operation	Minor
34	SDL crew	Minor	Body	Ground	S	Operation	Firstaid

Fig 4.4. Complete dataset using DPM

4.6 **MaxVoting Ensemble method Implementation**

4.6.1 Code description

Multiple imputation models are created using the traditional methods, i.e., MI-GLM, MI-CART and MI-DPM. A logistic regression model that has been trained using the results of the individual imputation models often makes up this meta-model. In order to produce a final imputation estimate, the meta-model learns to evaluate the accuracy and applicability of each individual model's predictions. The code to is shown in Appendix Fig I.5.

After producing the results, we compare the accuracy of the above results.

Chapter5 **Results**

The accuracies of various methods have been calculated as follows:

- The training dataset was the original dataset with each missing value rows deleted. The training dataset is shown in Appendix Fig. I.7.
- A logistic regression model was trained on that that dataset for calculating accuracy.
- The test dataset were the three complete datasets formed after performing the traditional Multiple Imputation methods over the original dataset.
- Their accuracies were further calculated as shown in Fig I.9.
- Then, an ensemble learning method developed were trained using three classifiers namely, Decision tree classifier, KNN Classifier and Gaussian Naive Bayes classifier as shown in Fig I.5.
- Its accuracy was calculated as shown in Fig I.10.

The results are shown in the table below –

Table 5.1. Accuracies of models

Method	Accuracy
Multiple Imputation using Generalized Linear models	75.2%
Multiple Imputation using Classification and regression trees	76.4%
Multiple Imputation using a Bayesian joint distribution based on Dirichlet process mixture models (DPM)	72.4%
MaxVoting Ensemble Model	83.8%

Chapter6 **Conclusion and Discussion**

The method of multiple Imputation is the best one when considering missing data in a larger dataset. The robustness of this method gives it an edge over single Imputation. The three standard methods which were discussed in this paper has been used over the years for Multiple Imputation with fair amount of accuracy. Also, the method of ensemble learning when applied to the dataset gives increased accuracy providing us with the benefit of more precise and robust complete datasets.

The outcomes of the simulation point to a number of broad conclusions regarding the three MI methods for categorical data. The default applications of MI-GLM with main effects, undoubtedly the most used multiple imputation solution, come first to be overall inferior to MI-CART and MI-DPM. With these latter techniques, significant dependence structures that are missed by default applications of MI-GLM. Of fact, with the generalized linear models, one might utilize more complex predictor functions, but with greater picking suitable groups of interactions from dimensional variables. Adding effects to the conditional models is difficult.

The analysis of complicated survey data is one area where multiple imputation utilizing ensemble learning may be employed. The sample strategy can significantly affect the processing of complicated survey data, and missing data can be a regular issue. This problem may be solved by multiple imputation utilizing ensemble learning, which produces imputed datasets that are representative of the population and take the intricate sample design into account. examination of longitudinal data is another. Missing data can be a prevalent issue since longitudinal studies frequently include repeated measurements of the same variables over time. By combining data from several time periods and producing imputed datasets that are consistent with the observed data, multiple imputation utilizing ensemble learning can assist to overcome this problem.

Additionally, a variety of industries, including healthcare, finance, the social sciences, and more, can use multiple imputation using ensemble learning. It may be applied in any circumstance where correct imputation is required for analysis and missing data is a concern. Overall, multiple imputation using ensemble learning is an effective method with a wide range of applications. It is an important tool for researchers and analysts dealing with missing data because of its capacity to produce precise imputed datasets and include uncertainty into the imputed values.

References

Akande, O., Li, F., & Reiter, J. (2017). An empirical comparison of multiple imputation methods for categorical data. *The American Statistician*, 71(2), 162-170.

Agresti A. (2013), *Categorical Data Analysis* (3rd ed.), Hoboken, NJ: Wiley

Arnold, B. C., and Press, S. J. (1989), "Compatible Conditional Distributions," *Journal of the American Statistical Association*, 84, 152–156.

Brand, J. P. L. (1999*), Development, Implementation and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets*, Dissertation, Erasmus University.

Brieman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression tree analysis.

Burgette, L. F., and Reiter, J. P. (2010), "Multiple Imputation for Missing Data via Sequential Regression Trees," *American Journal of Epidemiology*, 172, 1070–1076.

B. (1998), "The NHANES III Multiple Imputation Project," in *Proceedings of the survey Research Methods Section of the American Statistical Association*, pp. 28–37.

CHING, P., ZELL, E., EZZATIRICE, T., & MASSEY, J. (1995, June). IMMUNIZATION COVERAGE AMONG 2-YEAR-OLD CHILDREN-THE STATE AND LOCAL-AREA IMMUNIZATION COVERAGE AND HEALTH SURVEY (SLICHS). In *AMERICAN JOURNAL OF EPIDEMIOLOGY* (Vol. 141, No. 11, pp. S9-S9). 624 N BROADWAY RM 225, BALTIMORE, MD 21205: AMER J EPIDEMIOLOGY.

Jegadeeswari, K., Ragunath, R., & Rathipriya, R. (2022). Missing Data Imputation Using Ensemble Learning Technique: A Review. *Soft Computing for Security Applications: Proceedings of ICSCS 2022*, 223-236.

van Buuren, S. (2007), "Multiple Imputation of Discrete and Continuous Data by Fully Conditional Specification," *Statistical Methods in Medical Research*, 16, 219–242.

Gelman, A., and Speed, T. P. (1993), "Characterizing a Joint Probability Distribution by Conditionals," *Journal of the Royal Statistical Society,* Series B, 55, 185–188.

Harel, O., and Zhou, X. H. (2007), "Multiple Imputation: Review of Theory, Implementation and Software," *Statistics in Medicine*, 26, 3057–3077.

Raghunathan, T. E., and Rubin, D. B. (1997), "Roles for Bayesian Techniques in Survey Sampling," in *Proc. Survey Methods Section of the Statistical Society of Canada*, pp. 51–55.

Raghunathan, T. E., Solenberger, P. W., and Hoewyk, V. J. (2002), *Iveware: Imputation and Variance Estimation Software User Guide*, Ann Arbor, MI: Survey Research Center, Institute for Social Research, University of Michigan.

Reiter, J. P., and Raghunathan, T. E. (2007), "The Multiple Adaptations of Multiple Imputation," *Journal of the American Statistical Association*, 102, 1462–1471.

Royston, P., and White, I. R. (2011), "Multiple Imputation by Chained Equations (mice): Implementation in Stata," *Journal of Statistical Software*, 45, 1–20.

Su, Y. S., Gelman, A., Hill, J., & Yajima, M. (2011). Multiple imputation with diagnostics (mi) in R: Opening windows into the black box. *Journal of Statistical Software*, *45*, 1-31.

——— (1987), *Multiple Imputation for Nonresponse in Surveys*, New York: Wiley.

——— (1996), "Multiple Imputation After 18+ Years," *Journal of the American Statistical Association*, 91, 473–489.

——— (2003), "Nested Multiple Imputation of NMES via Partially Incompatible MCMC," *Statistica Neerlandica*, 57, 3–18.

Schafer, J. L. (1997), *Analysis of Incomplete Multivariate Data*, London: Chapman and Hall.

Schafer, J. L., Ezzati-Rice, T. M., Johnson, W., Khare, M., Little, R. J. A., and Rubin, D.

Van Buuren, S., Boshuizen, H. C., & Knook, D. L. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in medicine*, *18*(6), 681-694.

Van Buuren, S., Brand, J. P., Groothuis-Oudshoorn, C. G., & Rubin, D. B. (2006). Fully conditional specification in multivariate imputation. *Journal of statistical computation and simulation*, *76*(12), 1049-1064.

Van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, *45*, 1-67.

Vermunt, J. K., Van Ginkel, J. R., Van der Ark, L. A., & Sijtsma, K. (2008). 9. Multiple imputation of incomplete categorical data using latent class analysis. *Sociological Methodology*, *38*(1), 369-397.

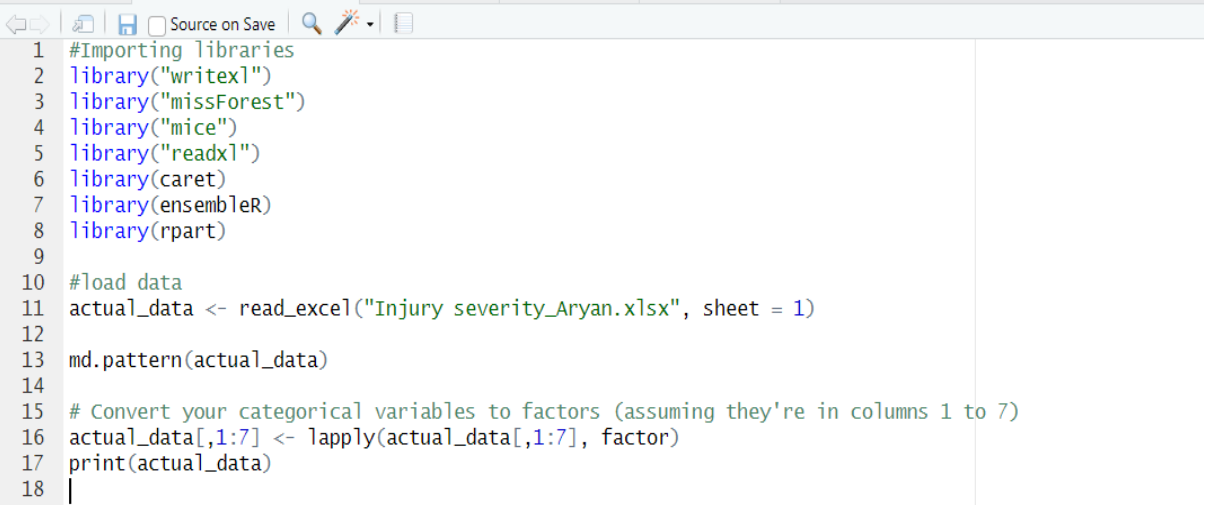
Yuan, Y. (2011). Multiple imputation using SAS software. *Journal of Statistical Software*, *45*, 1-25.

Tamboli, N. (2023, April 27). Effective strategies for handling missing values in data analysis (updated 2023). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/

Kjytay, &amp; Kjytay. (2020, July 23). Missing data and multivariate imputation by chained equations (mice). Statistical Odds &amp; Ends.
https://statisticaloddsandends.wordpress.com/2020/07/23/missing-data-and-multivariate-imputation-by-chained-equations-mice/

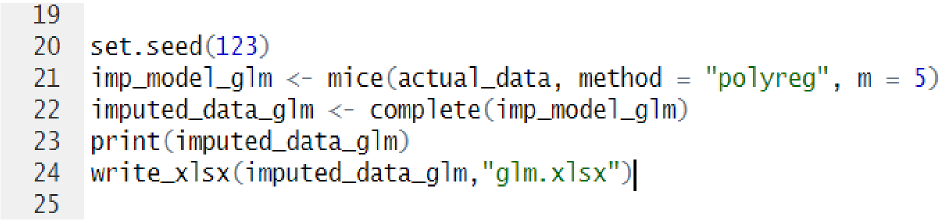
Appendix-I

Screenshots of codes:



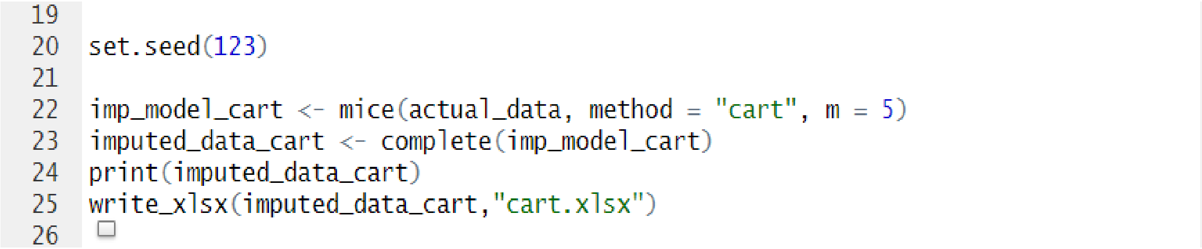
```r
#Importing libraries
library("writexl")
library("missForest")
library("mice")
library("readxl")
library(caret)
library(ensembleR)
library(rpart)

#load data
actual_data <- read_excel("Injury severity_Aryan.xlsx", sheet = 1)

md.pattern(actual_data)

# Convert your categorical variables to factors (assuming they're in columns 1 to 7)
actual_data[,1:7] <- lapply(actual_data[,1:7], factor)
print(actual_data)
```

Fig I.1. Label Encoding R code



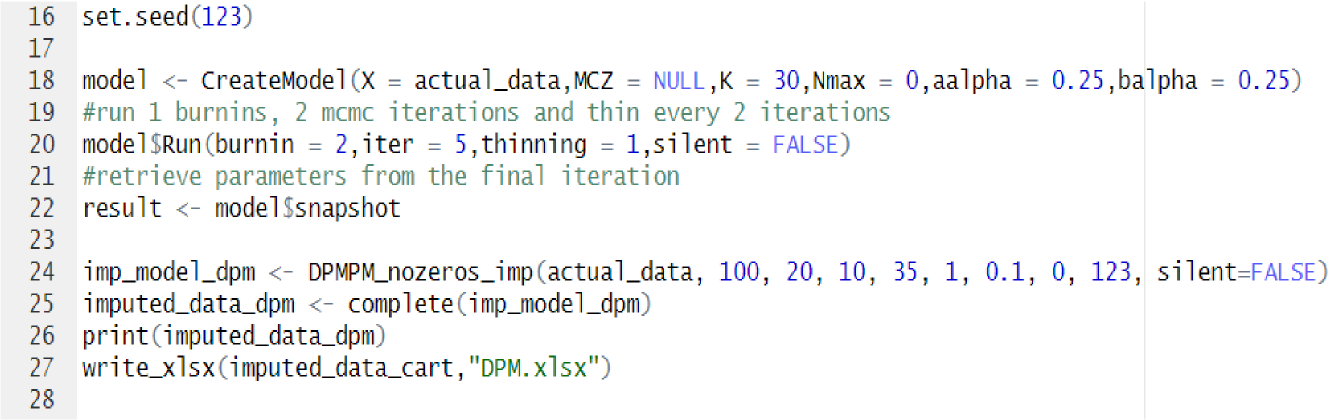
```r
set.seed(123)
imp_model_glm <- mice(actual_data, method = "polyreg", m = 5)
imputed_data_glm <- complete(imp_model_glm)
print(imputed_data_glm)
write_xlsx(imputed_data_glm,"glm.xlsx")
```

Fig I.2. MI-GLM R code



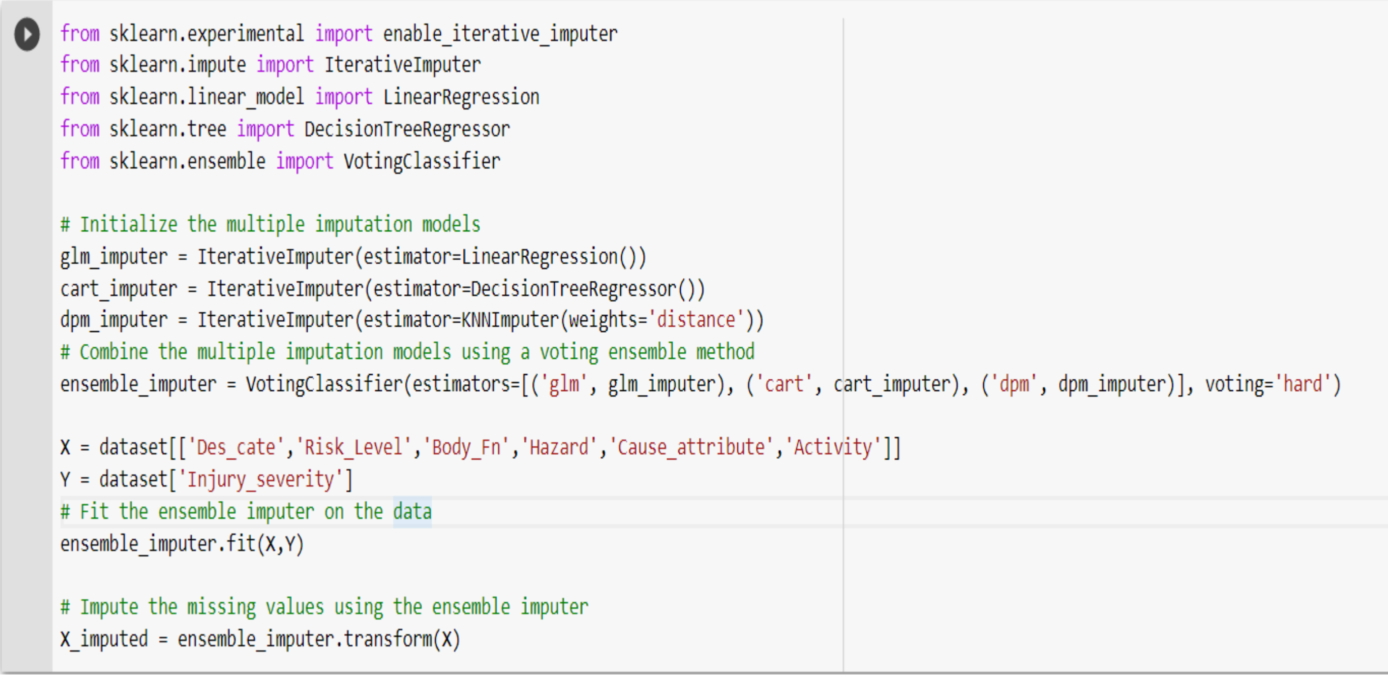
```r
set.seed(123)

imp_model_cart <- mice(actual_data, method = "cart", m = 5)
imputed_data_cart <- complete(imp_model_cart)
print(imputed_data_cart)
write_xlsx(imputed_data_cart,"cart.xlsx")
```

Fig I.3. MI-CART R code



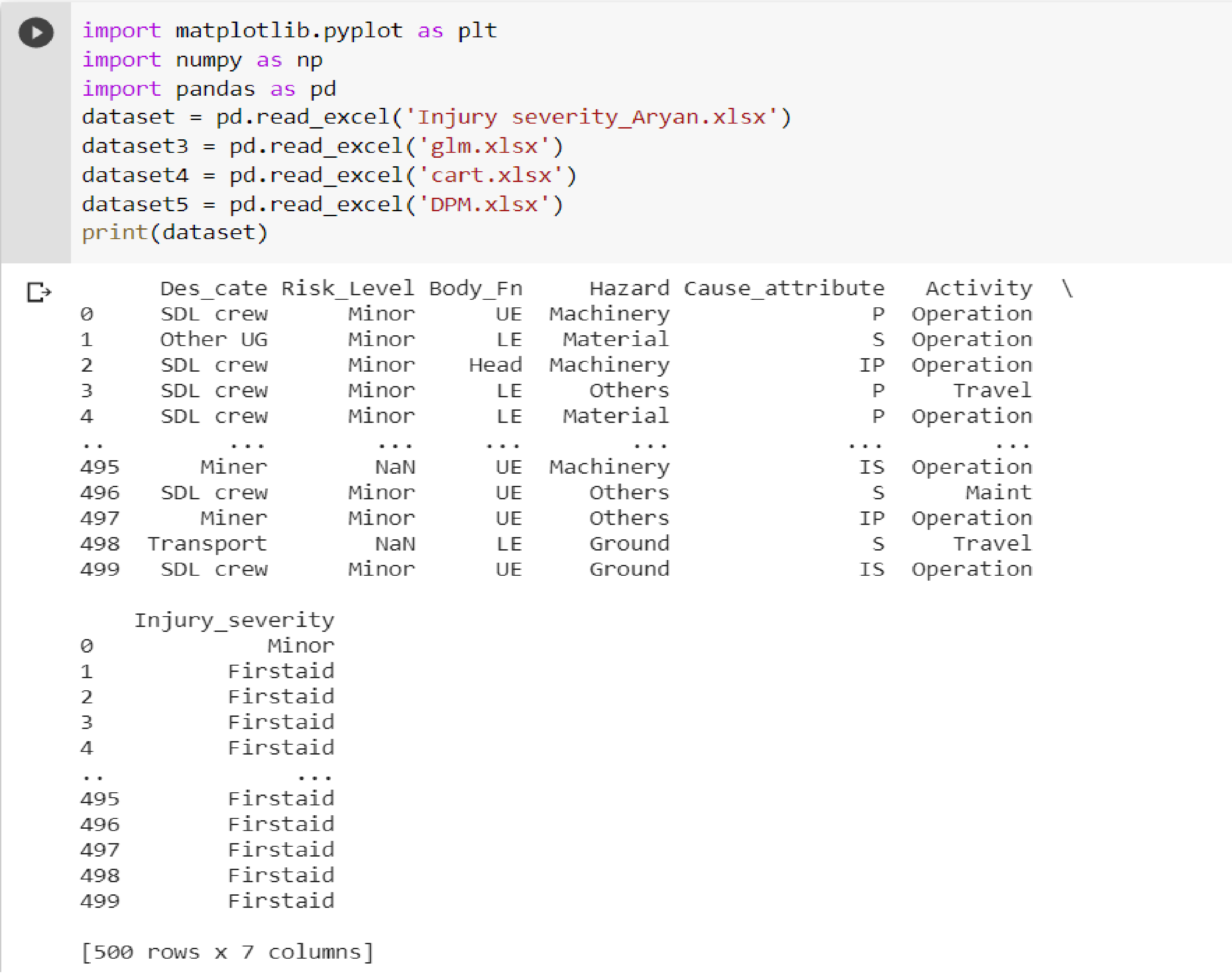
```
set.seed(123)

model <- CreateModel(X = actual_data,MCZ = NULL,K = 30,Nmax = 0,aalpha = 0.25,balpha = 0.25)
#run 1 burnins, 2 mcmc iterations and thin every 2 iterations
model$Run(burnin = 2,iter = 5,thinning = 1,silent = FALSE)
#retrieve parameters from the final iteration
result <- model$snapshot

imp_model_dpm <- DPMPM_nozeros_imp(actual_data, 100, 20, 10, 35, 1, 0.1, 0, 123, silent=FALSE)
imputed_data_dpm <- complete(imp_model_dpm)
print(imputed_data_dpm)
write_xlsx(imputed_data_cart,"DPM.xlsx")

```

Fig I.4. MI-DPM R code



```
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import VotingClassifier

# Initialize the multiple imputation models
glm_imputer = IterativeImputer(estimator=LinearRegression())
cart_imputer = IterativeImputer(estimator=DecisionTreeRegressor())
dpm_imputer = IterativeImputer(estimator=KNNImputer(weights='distance'))
# Combine the multiple imputation models using a voting ensemble method
ensemble_imputer = VotingClassifier(estimators=[('glm', glm_imputer), ('cart', cart_imputer), ('dpm', dpm_imputer)], voting='hard')

X = dataset[['Des_cate','Risk_Level','Body_Fn','Hazard','Cause_attribute','Activity']]
Y = dataset['Injury_severity']
# Fit the ensemble imputer on the data
ensemble_imputer.fit(X,Y)

# Impute the missing values using the ensemble imputer
X_imputed = ensemble_imputer.transform(X)
```

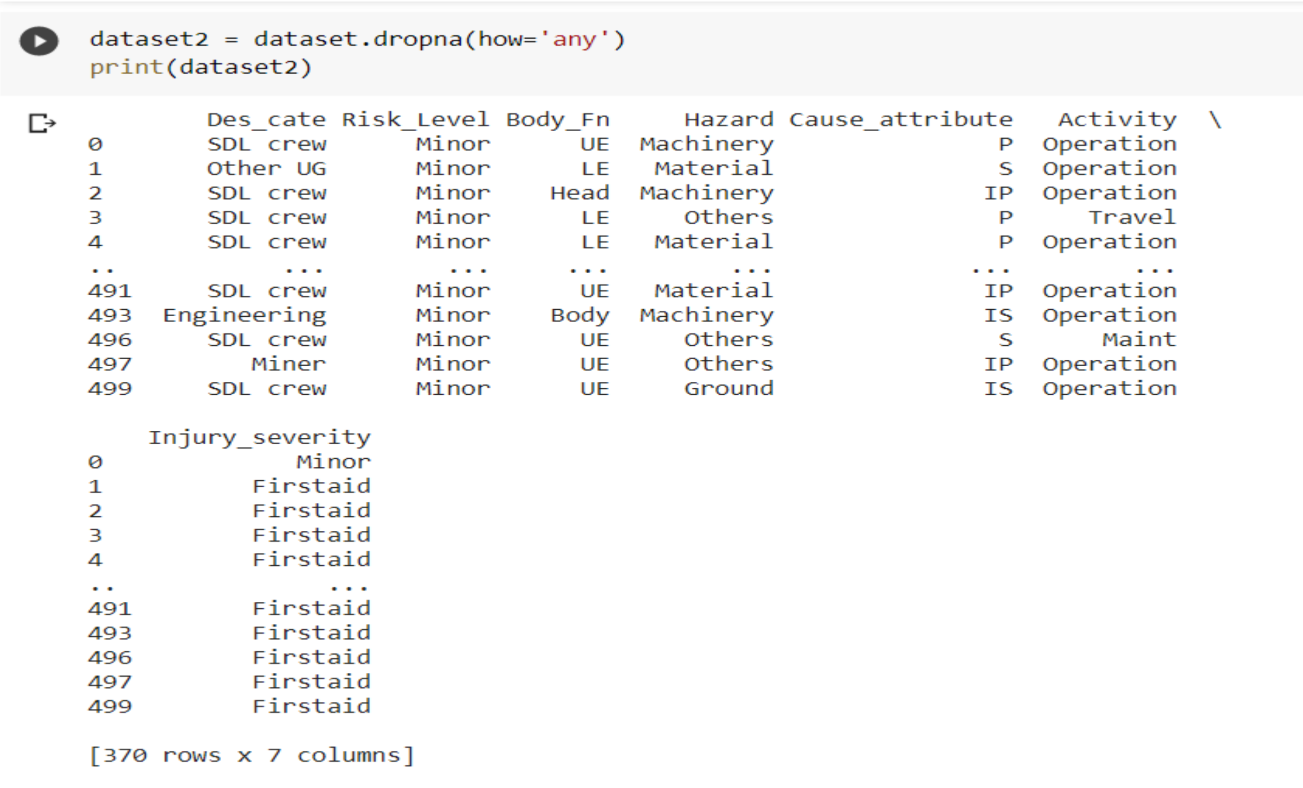
Fig I.5. Training a MaxVoting ensemble method



```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
dataset = pd.read_excel('Injury severity_Aryan.xlsx')
dataset3 = pd.read_excel('glm.xlsx')
dataset4 = pd.read_excel('cart.xlsx')
dataset5 = pd.read_excel('DPM.xlsx')
print(dataset)
```

```
      Des_cate Risk_Level Body_Fn     Hazard Cause_attribute   Activity  \
0     SDL crew      Minor      UE  Machinery               P  Operation
1     Other UG      Minor      LE   Material               S  Operation
2     SDL crew      Minor    Head  Machinery              IP  Operation
3     SDL crew      Minor      LE     Others               P     Travel
4     SDL crew      Minor      LE   Material               P  Operation
..         ...        ...     ...        ...             ...        ...
495      Miner        NaN      UE  Machinery              IS  Operation
496   SDL crew      Minor      UE     Others               S      Maint
497      Miner      Minor      UE     Others              IP  Operation
498  Transport        NaN      LE     Ground               S     Travel
499   SDL crew      Minor      UE     Ground              IS  Operation

    Injury_severity
0             Minor
1          Firstaid
2          Firstaid
3          Firstaid
4          Firstaid
..              ...
495        Firstaid
496        Firstaid
497        Firstaid
498        Firstaid
499        Firstaid

[500 rows x 7 columns]
```

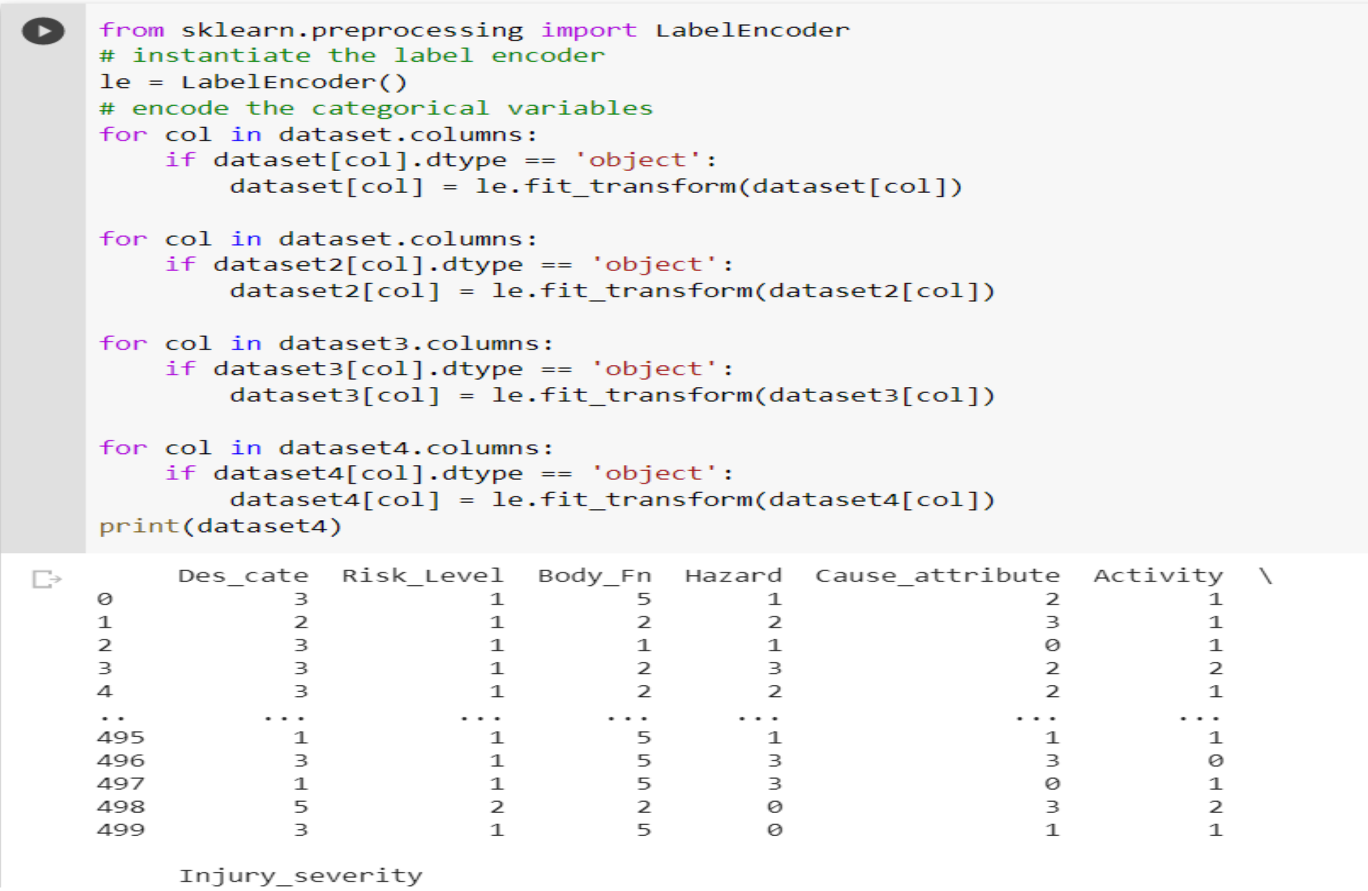
Fig I.6. Loading all the datasets



```
dataset2 = dataset.dropna(how='any')
print(dataset2)
```

```
        Des_cate Risk_Level Body_Fn     Hazard Cause_attribute   Activity  \
0       SDL crew      Minor      UE  Machinery               P  Operation   
1       Other UG      Minor      LE   Material               S  Operation   
2       SDL crew      Minor    Head  Machinery              IP  Operation   
3       SDL crew      Minor      LE     Others               P     Travel   
4       SDL crew      Minor      LE   Material               P  Operation   
..           ...        ...     ...        ...             ...        ...   
491     SDL crew      Minor      UE   Material              IP  Operation   
493  Engineering      Minor    Body  Machinery              IS  Operation   
496     SDL crew      Minor      UE     Others               S      Maint   
497        Miner      Minor      UE     Others              IP  Operation   
499     SDL crew      Minor      UE     Ground              IS  Operation   

    Injury_severity  
0             Minor  
1          Firstaid  
2          Firstaid  
3          Firstaid  
4          Firstaid  
..              ...  
491        Firstaid  
493        Firstaid  
496        Firstaid  
497        Firstaid  
499        Firstaid  

[370 rows x 7 columns]
```

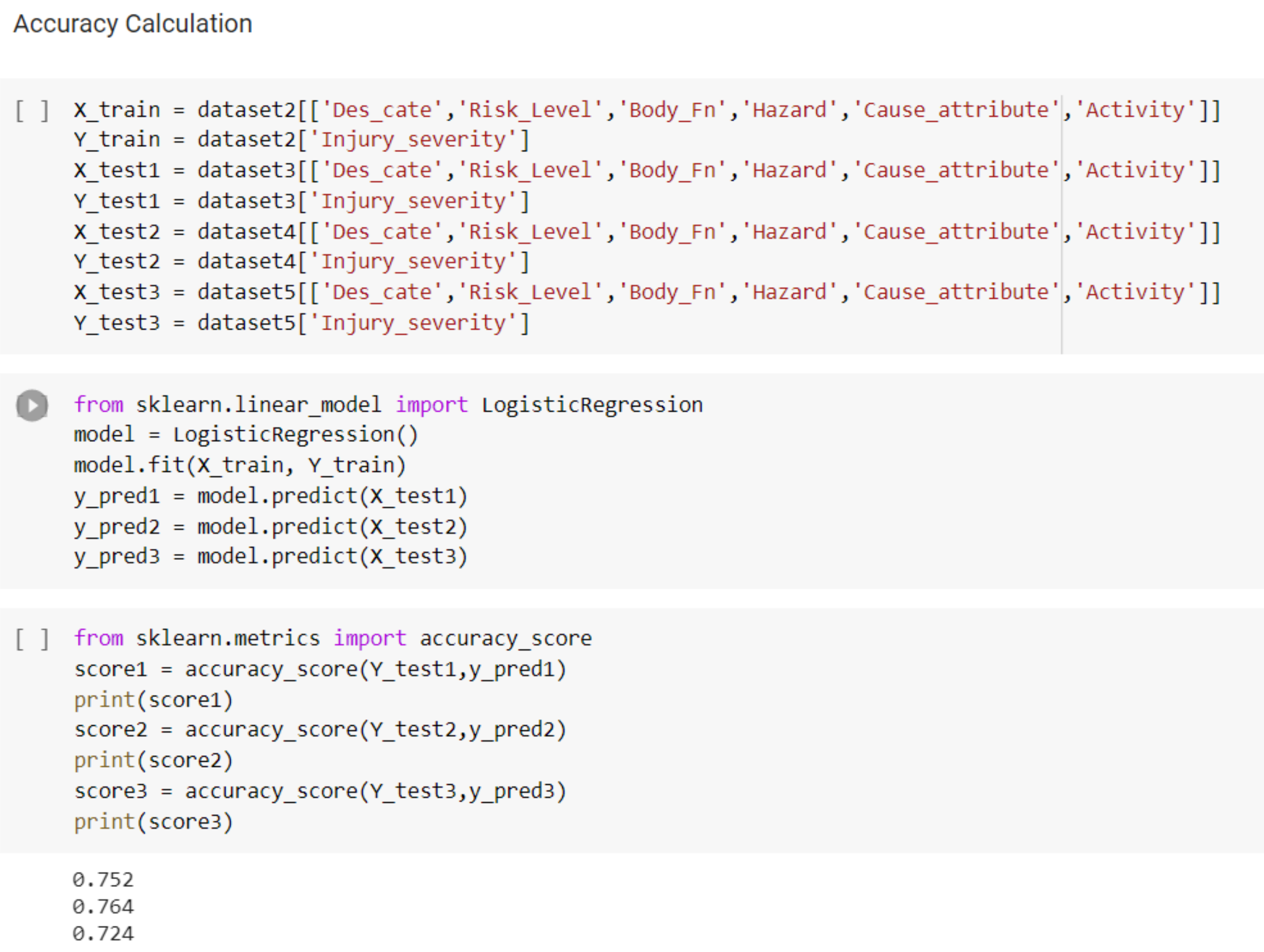
Fig I.7. Preparation of training dataset



```
from sklearn.preprocessing import LabelEncoder
# instantiate the label encoder
le = LabelEncoder()
# encode the categorical variables
for col in dataset.columns:
    if dataset[col].dtype == 'object':
        dataset[col] = le.fit_transform(dataset[col])

for col in dataset.columns:
    if dataset2[col].dtype == 'object':
        dataset2[col] = le.fit_transform(dataset2[col])

for col in dataset3.columns:
    if dataset3[col].dtype == 'object':
        dataset3[col] = le.fit_transform(dataset3[col])

for col in dataset4.columns:
    if dataset4[col].dtype == 'object':
        dataset4[col] = le.fit_transform(dataset4[col])
print(dataset4)
```

```
     Des_cate  Risk_Level  Body_Fn  Hazard  Cause_attribute  Activity  \
0           3           1        5       1                2         1   
1           2           1        2       2                3         1   
2           3           1        1       1                0         1   
3           3           1        2       3                2         2   
4           3           1        2       2                2         1   
..        ...         ...      ...     ...              ...       ...   
495         1           1        5       1                1         1   
496         3           1        5       3                3         0   
497         1           1        5       3                0         1   
498         5           2        2       0                3         2   
499         3           1        5       0                1         1   

     Injury_severity  
```

Fig I.8. Label Encoding of every dataset



Accuracy Calculation

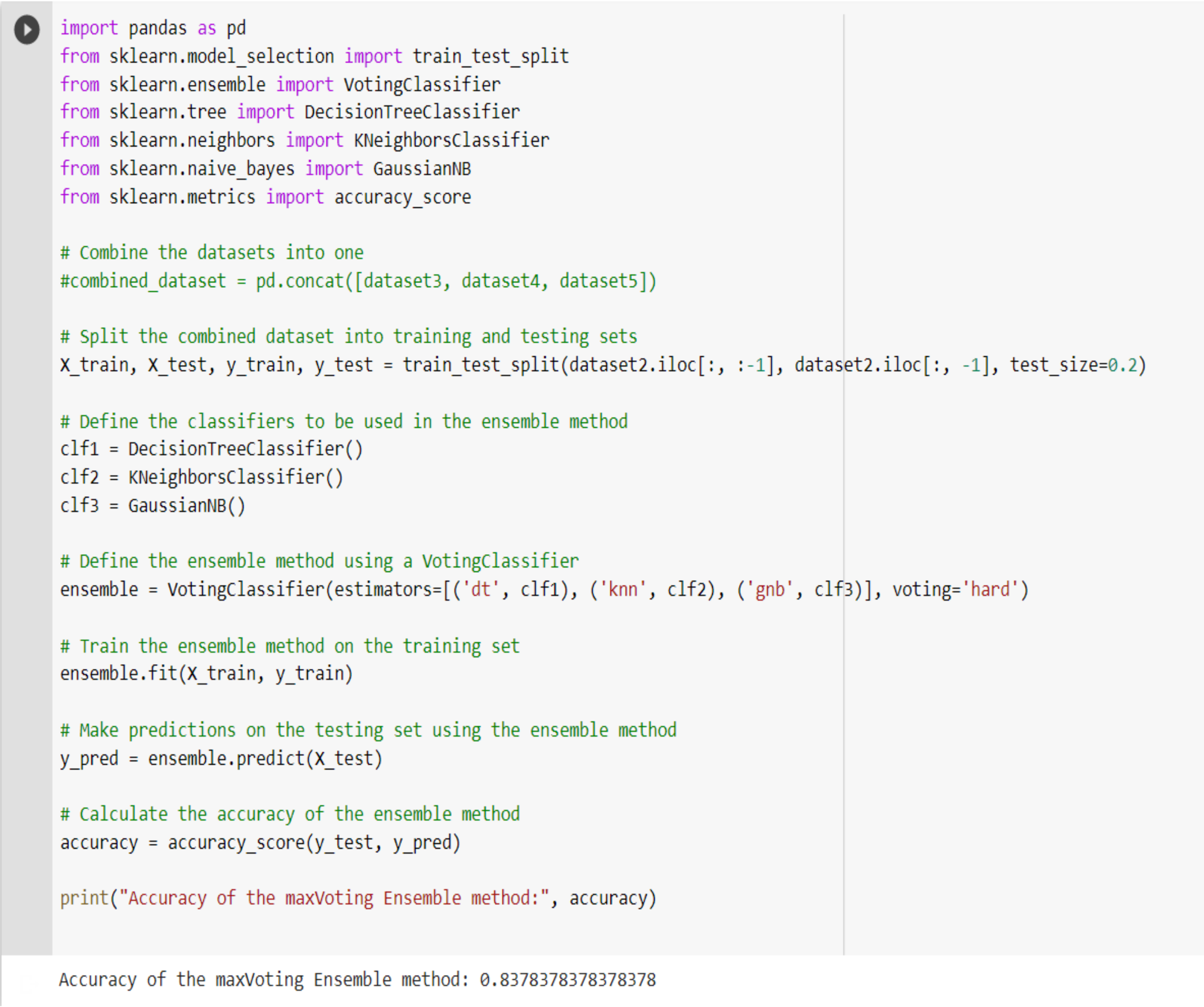
```
X_train = dataset2[['Des_cate','Risk_Level','Body_Fn','Hazard','Cause_attribute','Activity']]
Y_train = dataset2['Injury_severity']
X_test1 = dataset3[['Des_cate','Risk_Level','Body_Fn','Hazard','Cause_attribute','Activity']]
Y_test1 = dataset3['Injury_severity']
X_test2 = dataset4[['Des_cate','Risk_Level','Body_Fn','Hazard','Cause_attribute','Activity']]
Y_test2 = dataset4['Injury_severity']
X_test3 = dataset5[['Des_cate','Risk_Level','Body_Fn','Hazard','Cause_attribute','Activity']]
Y_test3 = dataset5['Injury_severity']
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, Y_train)
y_pred1 = model.predict(X_test1)
y_pred2 = model.predict(X_test2)
y_pred3 = model.predict(X_test3)
```

```
from sklearn.metrics import accuracy_score
score1 = accuracy_score(Y_test1,y_pred1)
print(score1)
score2 = accuracy_score(Y_test2,y_pred2)
print(score2)
score3 = accuracy_score(Y_test3,y_pred3)
print(score3)
```

```
0.752
0.764
0.724
```

Fig I.9. Accuracy Scores of MI-GLM, MI-CART and MI-DPM respectively



```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import VotingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Combine the datasets into one
#combined_dataset = pd.concat([dataset3, dataset4, dataset5])

# Split the combined dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(dataset2.iloc[:, :-1], dataset2.iloc[:, -1], test_size=0.2)

# Define the classifiers to be used in the ensemble method
clf1 = DecisionTreeClassifier()
clf2 = KNeighborsClassifier()
clf3 = GaussianNB()

# Define the ensemble method using a VotingClassifier
ensemble = VotingClassifier(estimators=[('dt', clf1), ('knn', clf2), ('gnb', clf3)], voting='hard')

# Train the ensemble method on the training set
ensemble.fit(X_train, y_train)

# Make predictions on the testing set using the ensemble method
y_pred = ensemble.predict(X_test)

# Calculate the accuracy of the ensemble method
accuracy = accuracy_score(y_test, y_pred)

print("Accuracy of the maxVoting Ensemble method:", accuracy)
```

```
Accuracy of the maxVoting Ensemble method: 0.8378378378378378
```

Fig I.10. Accuracy score of Ensemble method