Feature Engineering: Techniques and Applications

Mariana Teixeira[1], Luís Cavique[2]

[1]Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
marianaccteixeira@tecnico.ulisboa.pt

[2]Universidade Aberta, Lisboa, Portugal
luis.cavique@uab.pt

Abstract

Machine Learning is a rising concept in today's society. In the past decade, ML-based systems have become part of people's daily routines, and their usage has been disseminated through diverse sectors. This evolution is supported by the exponential increase in data created worldwide. Feature Engineering is a critical process focused on transforming data into suitable inputs for Machine Learning algorithms. This work explores the Feature Engineering process by developing a baseline for its implementation. Hence, a pipeline of Feature Engineering techniques and their taxonomy is proposed, along with a set of R scripts to implement. The validity of the code is then demonstrated through its application to a real-world dataset.

Keywords: data science, machine learning, data engineering, feature engineering, data transformation.

Título: Feature Engineering: Técnicas e Aplicações

Resumo: *Machine Learning* é um conceito em crescente evolução na sociedade atual. Na última década, os sistemas baseados em ML tornaram-se parte do quotidiano da população e a sua aplicação tem vindo a disseminar-se por diversos setores. Este crescimento é suportado pelo aumento exponencial da quantidade de dados gerados a nível mundial. *Feature Engineering* surge, assim, como um processo chave que permite transformar dados em *inputs* adequados para os algoritmos de *Machine Learning*. O presente trabalho pretende explorar o processo de *Feature Engineering*, com vista a desenvolver uma base de suporte à sua implementação. Por conseguinte, é proposta uma pipeline de técnicas de *Feature Engineering* em paralelo com a sua taxonomia, juntamente com um conjunto de *scripts* R, para as implementar. A validade do código é, posteriormente, demonstrada através da sua aplicação a um conjunto de dados reais.

Palavras-chave: ciência de dados, *machine* l*earning*, engenharia de dados, engenharia de atributos, transformação de dados.

1. Introduction

Machine Learning is active in several aspects of today's society, describing the capacity of systems to learn from problem-specific training data to automate the process of analytical model building and solve associated tasks [Kersting 2018]. In the past decade, ML-based systems have become a part of people's daily routines – self-driving cars, product recommendations, commuting predictions, and virtual assistants are a few examples of their applications. Considering all these innovations, the fact that the world is growing exponentially is not surprising, implying an equally significant increase in the volume of data collected. Data is becoming more meaningful and contextually relevant, breaking new grounds for Machine Learning [Kersting 2018].

The evolution of Machine Learning has also resulted in its dissemination through many different industries. It is a fact that data is a critical factor in any ML system – without proper data, the system is nothing but a hollow machine. Thus, considering the increasing diversification of sectors employing Machine Learning, it becomes clear that the data that feeds ML can come in all "sizes and shapes". This is where Feature Engineering arises. Feature Engineering exists because data does not have a systematic nature. Its time- and context-specific properties require domain expertise to engineer the features while minimizing potential information loss properly [Bastian *et al.* 2019]. There is no universal standard to automate this process, which depends on the data background and what it represents. The problem has shifted from collecting massive amounts of data to understanding it – turning it into knowledge, conclusions, and actions [Kersting 2018].

This paper intends to address the lack of a proper Feature Engineering baseline, i.e., a general sequence of steps that can be followed to handle a collection of data and turn it into valuable inputs for Machine Learning. Information about Feature Engineering is substantial but is also dispersed, unstructured, and usually context-specific. This work's primary goal is to define a general pipeline and taxonomy of Feature Engineering techniques and demonstrate its usage with a real-world dataset.

2. Feature Engineering Pipeline

There is no clear definition regarding the Feature Engineering workflow, i.e., the sequence of subprocesses to fully transform the dataset into ML-suitable data. When exploring the Feature Engineering domain, numerous processes can be found concerning different steps of its pipeline. Some of these have similar definitions and are used interchangeably. Some are mutually exclusive, some are compatible, and some overlap others, which makes it hard to formally describe a sequence of activities that can accurately encompass all the critical steps. All these different notions can lead to confusion when establishing a pipeline of processes for applying Feature Engineering. Table 1 arranges the existing processes according to their main purpose, dividing them into four categories.

Table 1. Feature Engineering processes by goal

Gathering data	Cleaning data	Transforming data	Resizing data
Data Collection	Data Clean(s)ing	Data Wrangling	Dimensionality Reduction
Data Extraction	Data Scrubbing	Data Munging	Data Selection
	Data Quality Assurance	Data Transformation	Feature Selection
	Data Quality Management	Feature Scaling	Feature Extraction
	Data Remediation		
ETL			
		Data Preparing	
		Data Preparation	
		Data Preprocessing	
		Feature Engineering	

Identifying these four distinct groups of processes was the first step in defining a logical sequence of general stages for Feature Engineering. By excluding the first category, "Gathering data", considering that this activity is not a part of Feature Engineering, three categories can be used as cornerstones for the workflow of processes. Hence, Figure 1 proposes a pipeline of three steps, comprising Data Cleaning, Data Transformation, and Data Reduction. By applying the most fitting techniques associated with each, implementing these three processes should result in a more suitable dataset for Machine Learning activities. The proposed pipeline aims to consolidate all the concepts analyzed above (Table 1) and simplify the feature engineering workflow.

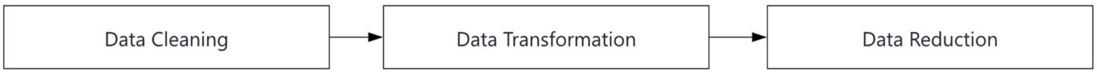


Figure 1. Pipeline of Feature Engineering processes

2.1 Data Cleaning

Data Cleaning is fixing the dataset by detecting and handling faulty data. It aims to make the dataset consistent by addressing problems such as missing values, data errors, heterogeneous formats, and duplicates. The most relevant techniques to address each issue are described in Figure 2.

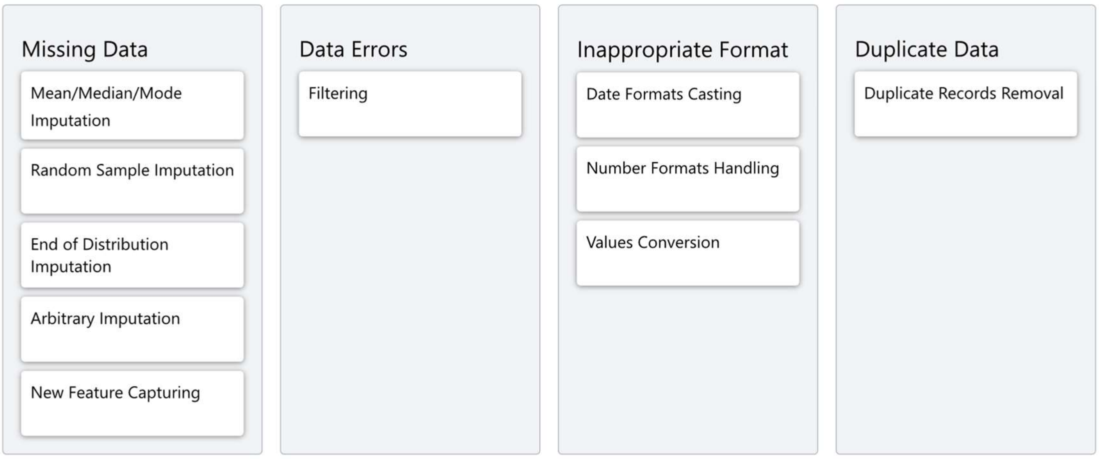


Figure 2. Data Cleaning issues and techniques

2.2 Data Transformation

Data Transformation is the process of converting data into a format that is suitable for Machine Learning algorithms. It can involve changing the data's type, structure, or values. Figure 3 presents four issues regarding Data Transformation, along with some Feature Engineering techniques that can be used to handle them.

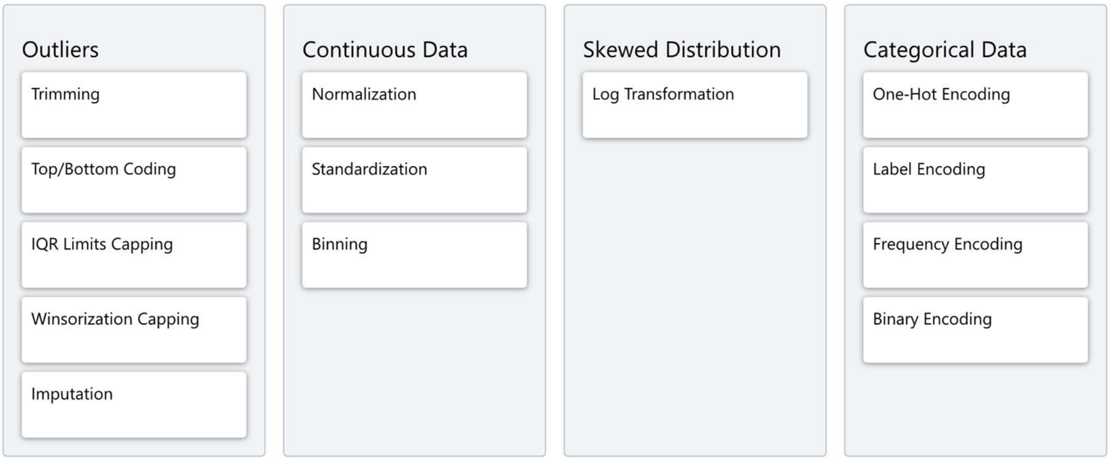


Figure 3. Data Transformation issues and techniques

2.3 Data Reduction

Data Reduction is reducing the dimensionality of a dataset by dropping a set of features while keeping the integrity and meaning of the original data. Its main goal is to decrease the volume of the data and increase the processing efficiency. Figure 4 identifies three different Feature Engineering techniques to address High Dimensionality.

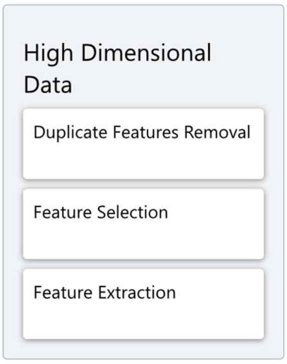


Figure 4. Data Reduction issues and techniques

3. Evaluation

This section uses a training dataset, obtained through the Kaggle repository to practically demonstrate the previously presented Feature Engineering techniques.

3.1 Data Description

The Titanic dataset offers information about 891 passengers who boarded the RMS Titanic, comprising 12 features described in Table 2.

Table 2. Dataset features description

#	Attribute Name	Type	Description
1	PassengerId	Integer	Key identifier of the passenger in the dataset
2	*Survived (Target)*	*Integer*	*Indicator of the passenger survival* *Possible Values:* (0 = No; 1 = Yes)
3	PClass	Integer	Boarding class of the passenger *Possible Values:* (1 = 1st; 2 = 2nd; 3 = 3rd)
4	Name	Categorical	Name of the passenger
5	Sex	Categorical	Sex of the passenger
6	Age	Numeric	Age of the passenger *Note 1:* If the Age is estimated, it is in the xx. five formats. *Note 2:* If the Age is less than 1, it is in fractional format.
7	SibSp	Integer	Number of siblings/spouses aboard
8	Parch	Integer	Number of parents/children aboard
9	Ticket	Categorical	Ticket Number *Note 1:* It is not unique and can be shared by passengers.
10	Fare	Numerical	Fare (in Pre-1970 British Pounds)
11	Cabin	Categorical	Cabin
12	Embarked	Categorical	Port of Embarkation *Possible Values:* (C = Cherbourg; Q = Queenstown; S = Southampton)

3.2 Feature Engineering Application

The demonstration of the afore-presented techniques is divided into different subsections, each regarding a different dataset problem. Different techniques concerning the three steps of the Feature Engineering pipeline are applied to the Titanic dataset. The techniques were applied in an R environment, with a previously developed script comprising functions for each of the techniques presented earlier.

Missing Data. Running some simple instructions in the R console is enough to identify *Age*, *Embarked*, and *Cabin* as the features containing missing values.

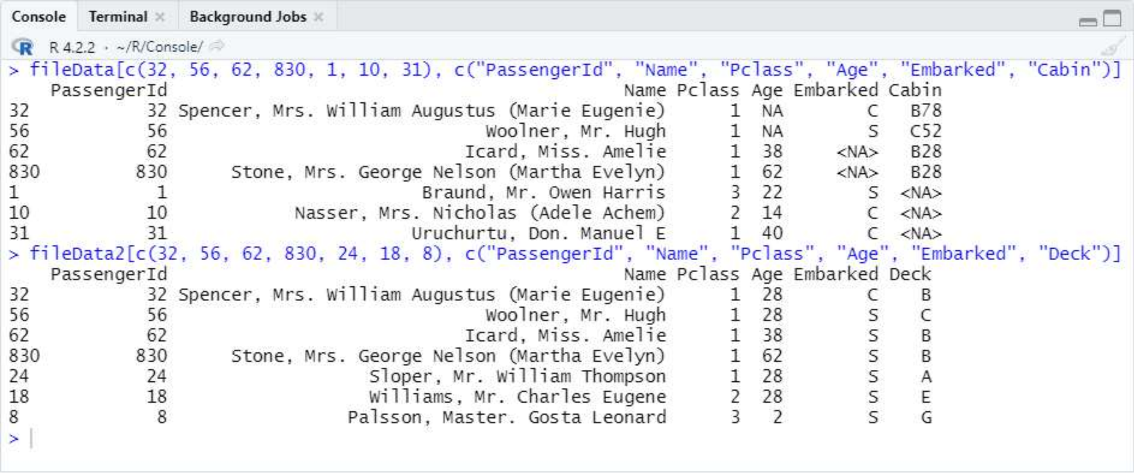
Age. As a numerical variable with a distribution close to normal but still asymmetrical, the handling of *Age*-missing data can be achieved through Median Imputation. Figure 5 shows a portion of the resulting dataset, where the empty age values were replaced by the distribution median (28).

Embarked. This feature is of type categorical and has three possible values. The Mode Imputation technique is used to impute its missing values. Figure 5 shows the application of Mode Imputation to a portion of the dataset. The missing values were replaced by the distribution mode (S - Southampton).

Cabin. This feature has the most missing values in the dataset, comprising a letter corresponding to the deck where the cabin is allocated and a number to identify the cabin itself. The individual number is too granular and can be ignored, keeping only the deck letter as a feature. For the imputation of the missing values, the Random Sample Imputation technique can be used on the newly derived feature *Deck*. For this scenario, it is important to consider the relationship between *Cabin/Deck* and *PClass*, since some decks were only meant for specific classes, as detailed below.

- First Class passengers stayed on decks A, B, C, D, and E (T can be ignored since there is only one record);
- Second Class passengers stayed on decks D, E, and F;
- Third Class passengers stayed on decks E, F, and G.

Hence, the Random Sample Imputation technique can be used separately for the three different classes, considering the samples determined above. Figure 5 shows the before and after of a portion of the dataset, where the empty fields for *Cabin/Deck* were filled with random values from their respective samples.

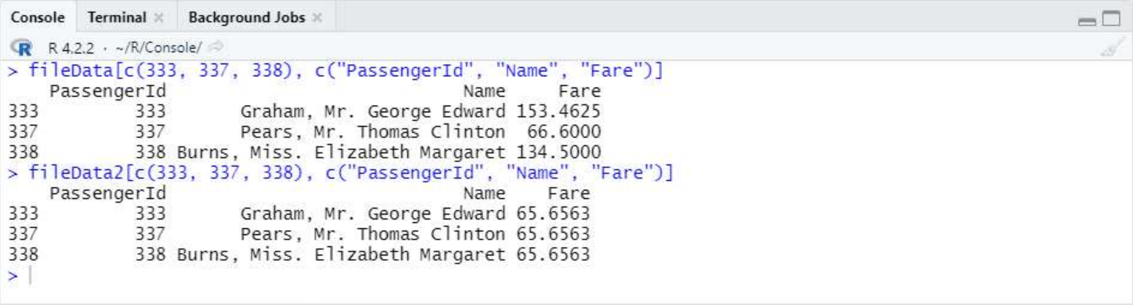


```
> fileData[c(32, 56, 62, 830, 1, 10, 31), c("PassengerId", "Name", "Pclass", "Age", "Embarked", "Cabin")]
    PassengerId                                          Name Pclass Age Embarked Cabin
32           32 Spencer, Mrs. William Augustus (Marie Eugenie)      1  NA        C   B78
56           56                             Woolner, Mr. Hugh      1  NA        S   C52
62           62                           Icard, Miss. Amelie      1  38     <NA>   B28
830         830     Stone, Mrs. George Nelson (Martha Evelyn)      1  62     <NA>   B28
1             1                       Braund, Mr. Owen Harris      3  22        S  <NA>
10           10           Nasser, Mrs. Nicholas (Adele Achem)      2  14        C  <NA>
31           31                      Uruchurtu, Don. Manuel E      1  40        C  <NA>
> fileData2[c(32, 56, 62, 830, 24, 18, 8), c("PassengerId", "Name", "Pclass", "Age", "Embarked", "Deck")]
    PassengerId                                          Name Pclass Age Embarked Deck
32           32 Spencer, Mrs. William Augustus (Marie Eugenie)      1  28        C    B
56           56                             Woolner, Mr. Hugh      1  28        S    C
62           62                           Icard, Miss. Amelie      1  38        S    B
830         830     Stone, Mrs. George Nelson (Martha Evelyn)      1  62        S    B
24           24                  Sloper, Mr. William Thompson      1  28        S    A
18           18                  Williams, Mr. Charles Eugene      2  28        S    E
8             8                Palsson, Master. Gosta Leonard      3   2        S    G
> |
```

Figure 4. Dataset extract before and after handling Missing Data

Outliers. The addressing of outliers falls on the *Fare* feature considering its dispersed distribution with several extreme values in the upper region.

Fare. For this feature, a high percentage of the values is concentrated between 0 and 100 pounds, while the rest is scattered up until around 512 pounds. To deal with the extreme values, the IQR Limits Capping technique can be used, replacing the outliers with the lower and upper limits accordingly. For this feature, there are 116 values above the upper limit (65.6563) and no values below the lower limit (-26.7605). Figure 6 presents a portion of the resulting dataset, where the *Fare* outliers were capped with the upper IQR limit.



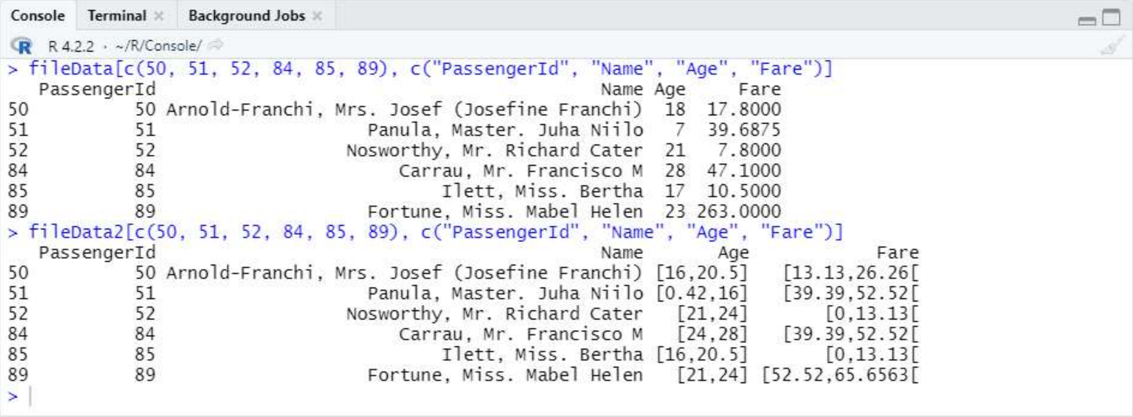
```
> fileData[c(333, 337, 338), c("PassengerId", "Name", "Fare")]
    PassengerId                            Name     Fare
333         333       Graham, Mr. George Edward 153.4625
337         337       Pears, Mr. Thomas Clinton  66.6000
338         338 Burns, Miss. Elizabeth Margaret 134.5000
> fileData2[c(333, 337, 338), c("PassengerId", "Name", "Fare")]
    PassengerId                            Name    Fare
333         333       Graham, Mr. George Edward 65.6563
337         337       Pears, Mr. Thomas Clinton 65.6563
338         338 Burns, Miss. Elizabeth Margaret 65.6563
> 
```

Figure 5. Dataset extract before and after handling Outliers

Continuous Data. Regarding continuous information, *Age* and *Fare* are the two features with ranges that can benefit from continuous data handling techniques.

Age. A common strategy to handle age features is to generate age groups, through Binning. The Binning approach to follow can be either Equal-Frequency or Equal-Width. For this variable, the Equal-Frequency Binning technique should be used in order to ensure that every bin is relevant to the model. Figure 7 shows the application of the Equal-Frequency Binning technique to the *Age* feature, using 10 bins.

Fare. After removing the outliers in the previous subsection, the range of the *Fare* variable decreased significantly. The Equal-Width approach is adopted considering that there is a relatively significant number of observations along the range of values. Figure 7 shows the application of the Equal-Width Binning technique with 5 bins.



```
> fileData[c(50, 51, 52, 84, 85, 89), c("PassengerId", "Name", "Age", "Fare")]
   PassengerId                                          Name Age     Fare
50          50 Arnold-Franchi, Mrs. Josef (Josefine Franchi)  18  17.8000
51          51                    Panula, Master. Juha Niilo   7  39.6875
52          52                  Nosworthy, Mr. Richard Cater  21   7.8000
84          84                       Carrau, Mr. Francisco M  28  47.1000
85          85                           Ilett, Miss. Bertha  17  10.5000
89          89                    Fortune, Miss. Mabel Helen  23 263.0000
> fileData2[c(50, 51, 52, 84, 85, 89), c("PassengerId", "Name", "Age", "Fare")]
   PassengerId                                          Name       Age            Fare
50          50 Arnold-Franchi, Mrs. Josef (Josefine Franchi) [16,20.5]   [13.13,26.26[
51          51                    Panula, Master. Juha Niilo [0.42,16]   [39.39,52.52[
52          52                  Nosworthy, Mr. Richard Cater   [21,24]       [0,13.13[
84          84                       Carrau, Mr. Francisco M   [24,28]   [39.39,52.52[
85          85                           Ilett, Miss. Bertha [16,20.5]       [0,13.13[
89          89                    Fortune, Miss. Mabel Helen   [21,24] [52.52,65.6563[
> 
```

Figure 6. Dataset extract before and after handling Continuous Data

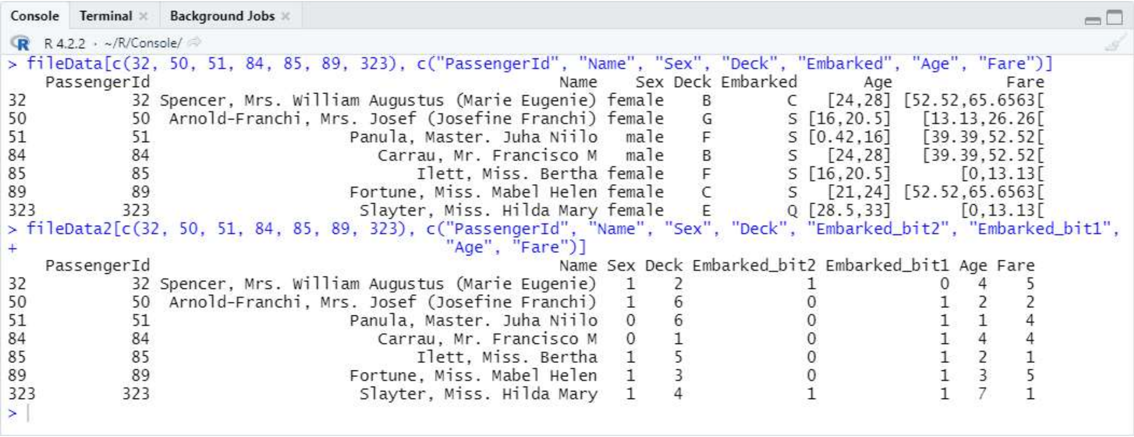
Categorical Data. Categorical data plays an important part in the dataset. *Sex*, *Embarked* and the newly created *Deck* are relevant features that need to be encoded as quantitative data in order to be of use to the Machine Learning model. The discretized features, *Age* and *Fare* are no longer numeric and need to be encoded as well.

Sex. The *Sex* feature has only two possible values – male or female. To encode it, the Label Encoding technique is used. This technique assigns a numerical label to each of the possible values, in a way that "male" becomes represented by 0 and "female" by 1. As shown in Figure 8, after applying the function, *Sex* becomes a binary feature.

Deck. This is the feature generated from *Cabin* when handling missing data. Unlike "male" and "female", the *Deck* values have a meaningful order since they correspond to consecutive sections of the ship. Hence, for this attribute, Ordinal Encoding, a variant of Label Encoding, would be a better fit. This technique converts each value into a number while preserving the natural order of the original sequence. As demonstrated in Figure 8, a label was associated with each of the possible values, such that 'A' = 1, 'B' = 2, 'C' = 3, 'D' = 4, 'E' = 5, and 'F' = 6.

Embarked. To encode the *Embarked* feature, the Binary Encoding technique was chosen as a way of assigning a numeric label to each of the possible values and, after converting this number into binary notation, splitting it into individual binary features. Hence, 'S' would correspond to 01, 'C' to 10, and 'Q' to 11. Since, the three new values consist of two digits (or bits), the number of new features generated is two. Figure 8 shows the application of Binary Encoding to a portion of the dataset.

Age and *Fare*. *Age* and *Fare* are the two features to which Binning was applied, converting them from numerical to categorical. Therefore, encoding the new values, now corresponding to intervals, is necessary. In order to retain the implicit order of the sequenced groups, Ordinal Encoding will be used on both features. Figure 8 shows the application of the Ordinal Encoding technique to both features, where the different groups were replaced by the corresponding sequenced labels.



```
> fileData[c(32, 50, 51, 84, 85, 89, 323), c("PassengerId", "Name", "Sex", "Deck", "Embarked", "Age", "Fare")]
    PassengerId                                           Name    Sex Deck Embarked       Age            Fare
32           32 Spencer, Mrs. William Augustus (Marie Eugenie) female    B        C   [24,28] [52.52,65.6563[
50           50  Arnold-Franchi, Mrs. Josef (Josefine Franchi) female    G        S [16,20.5]   [13.13,26.26[
51           51                     Panula, Master. Juha Niilo   male    F        S [0.42,16]   [39.39,52.52[
84           84                        Carrau, Mr. Francisco M   male    B        S   [24,28]   [39.39,52.52[
85           85                            Ilett, Miss. Bertha female    F        S [16,20.5]       [0,13.13[
89           89                     Fortune, Miss. Mabel Helen female    C        S   [21,24] [52.52,65.6563[
323         323                      Slayter, Miss. Hilda Mary female    E        Q [28.5,33]       [0,13.13[
> fileData2[c(32, 50, 51, 84, 85, 89, 323), c("PassengerId", "Name", "Sex", "Deck", "Embarked_bit2", "Embarked_bit1",
+                                              "Age", "Fare")]
    PassengerId                                           Name Sex Deck Embarked_bit2 Embarked_bit1 Age Fare
32           32 Spencer, Mrs. William Augustus (Marie Eugenie)   1    2             1             0   4    5
50           50  Arnold-Franchi, Mrs. Josef (Josefine Franchi)   1    6             0             1   2    2
51           51                     Panula, Master. Juha Niilo   0    6             0             1   1    4
84           84                        Carrau, Mr. Francisco M   0    1             0             1   4    4
85           85                            Ilett, Miss. Bertha   1    5             0             1   2    1
89           89                     Fortune, Miss. Mabel Helen   1    3             0             1   3    5
323         323                      Slayter, Miss. Hilda Mary   1    4             1             1   7    1
> |
```

Figure 7. Dataset extract before and after handling Categorical Data

High Dimensional Data. The Titanic dataset does not display high dimensionality issues. Nevertheless, checking the data for duplicate features is simple and valuable.

Duplicate Information. A useful test is to ensure that there are no features providing related information, i.e., information that can be inferred from other attributes. To test for this issue, the *duplicateInformationFeatureRemoval* function was executed to all pairs of features (excluding *PassengerId* and *Name*), returning the sets below.

- *Ticket* and *Fare* – All records sharing the ticket number also share the fare value, considering that one ticket should only have one price. Hence, in the context of ML, both provide the same information and one of them can be safely discarded.

- *Cabin* and *Deck* – *Deck* is a feature created to handle *Cabin* missing data. Since *Deck* derives directly from the *Cabin* value (it corresponds to the first letter), it is expected that they act as duplicates. Given that the *Deck* feature was created with the purpose of replacing *Cabin*, the latter can be dropped.
- *Pclass* and *Ticket*/*Pclass* and *Cabin* – All tickets with the same number should correspond to only one class and the same goes for *Cabin* since each cabin belongs to only one class as well. As was already established in the previous points, both *Ticket* and *Cabin* should be dropped, therefore, eliminating the duplications.

Considering the reflections above, Figure 10 shows the resulting dataset features.

Drop Unnecessary Features. Dropping features does not have to always be based on the application of Data Reduction techniques. A feature can be discarded because it is simply not useful to the prediction goal. In the Titanic dataset, there are two identifier features, *PassengerId* and *Name*, that serve only as differentiators for the rows. Hence, having one identifier is enough and the *Name* feature can be safely dropped. Figure 10 shows the resulting dataset features.

3.3 Discussion

The complete pipeline of the techniques employed in the previous subsection is described in Figure 9.

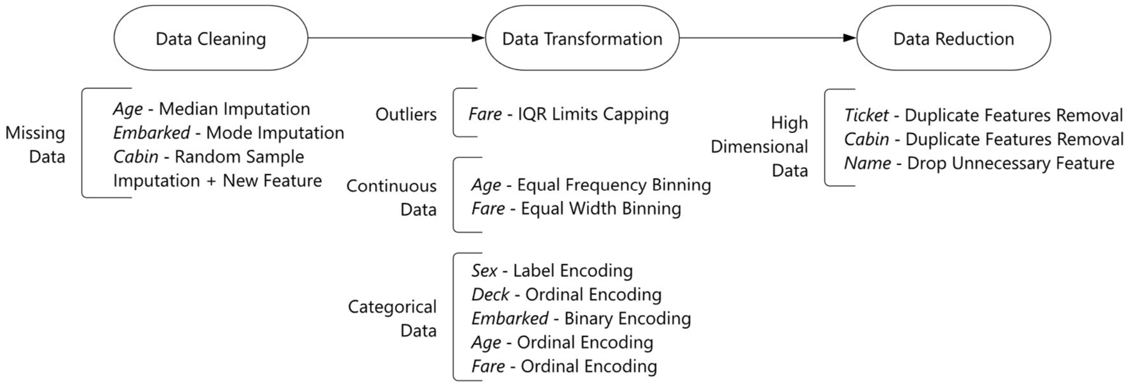

Data Cleaning
Data Transformation
Data Reduction
Missing Data
Age - Median Imputation
Embarked - Mode Imputation
Cabin - Random Sample Imputation + New Feature
Outliers
Fare - IQR Limits Capping
Continuous Data
Age - Equal Frequency Binning
Fare - Equal Width Binning
Categorical Data
Sex - Label Encoding
Deck - Ordinal Encoding
Embarked - Binary Encoding
Age - Ordinal Encoding
Fare - Ordinal Encoding
High Dimensional Data
Ticket - Duplicate Features Removal
Cabin - Duplicate Features Removal
Name - Drop Unnecessary Feature

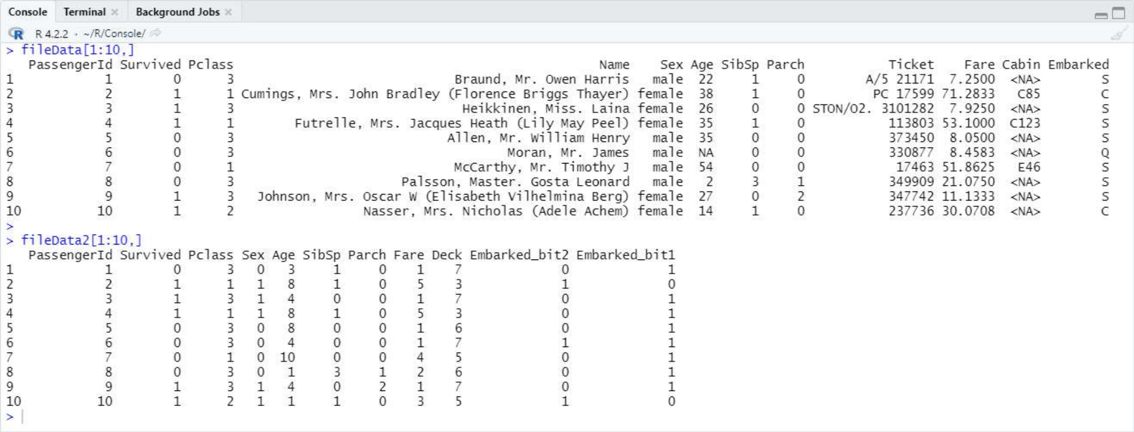
Figure 8. Feature Engineering pipeline applied to the Titanic dataset

Table 3 consolidates the Feature Engineering process for each feature, including their types before and after applying all transformations.

Table 3. Dataset features description after applying Feature Engineering

#	Attribute Name	Type before Feature Engineering	Type after Feature Engineering	Problem – Feature Engineering Technique
1	PassengerId	Integer	-	-
2	Survived	Integer (0..1)	-	-
3	PClass	Integer (1..3)	-	-
-	Name	Categorical	-	High Dimensional Data – Drop Unnecessary Features
4	Sex	Categorical	Binary	Categorical Data – Label Encoding
5	Age	Numeric	Integer	Missing Data – Median Imputation Continuous Data – Binning Categorical Data – Ordinal Encoding
6	SibSp	Integer	-	-
7	Parch	Integer	-	-
-	Ticket	Categorical	Dropped	High Dimensional Data – Duplicate Features Removal
8	Fare	Numerical	Integer	Outliers – IQR Limits Capping Continuous Data – Binning Categorical Data – Ordinal Encoding
-	Cabin	Categorical	Dropped	Missing Data – Random Sample Imputation + New Feature (#9) High Dimensional Data – Duplicate Features Removal
9	Deck	-	Integer	Categorical Data – Ordinal Encoding
-	Embarked	Categorical	Integer (Replaced by #12 and #13)	Missing Data – Mode Imputation Categorical Data – Binary Encoding (#10 and #11)
10	Embarked_bit2	-	Integer	-
11	Embarked_bit1	-	Integer	-

There is no universal set of Feature Engineering techniques that should be applied to a dataset. The proposed pipeline was created with the purpose of fitting all datasets, but the specific set of techniques to employ should always be defined based on the data in question and, potentially, the Machine Learning model to be used. This last topic is not considered for the present work, but it is closely related to Feature Engineering since some techniques work better with some models than others. Hence, the techniques applied to the Titanic dataset in Section 3.2 served as an example of a possible Feature Engineering workflow, which does not imply that there couldn't be others with equal, better, or worse results. Figure 10 shows the first 10 records of the Titanic dataset before and after going through the Feature Engineering pipeline.



```
> fileData[1:10,]
   PassengerId Survived Pclass                                                Name    Sex Age SibSp Parch           Ticket    Fare Cabin Embarked
1            1        0      3                             Braund, Mr. Owen Harris   male  22     1     0        A/5 21171  7.2500  <NA>        S
2            2        1      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0         PC 17599 71.2833   C85        C
3            3        1      3                              Heikkinen, Miss. Laina female  26     0     0 STON/O2. 3101282  7.9250  <NA>        S
4            4        1      1        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0           113803 53.1000  C123        S
5            5        0      3                            Allen, Mr. William Henry   male  35     0     0           373450  8.0500  <NA>        S
6            6        0      3                                    Moran, Mr. James   male  NA     0     0           330877  8.4583  <NA>        Q
7            7        0      1                             McCarthy, Mr. Timothy J   male  54     0     0            17463 51.8625   E46        S
8            8        0      3                      Palsson, Master. Gosta Leonard   male   2     3     1           349909 21.0750  <NA>        S
9            9        1      3   Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) female  27     0     2           347742 11.1333  <NA>        S
10          10        1      2                 Nasser, Mrs. Nicholas (Adele Achem) female  14     1     0           237736 30.0708  <NA>        C
>
> fileData2[1:10,]
   PassengerId Survived Pclass Sex Age SibSp Parch Fare Deck Embarked_bit2 Embarked_bit1
1            1        0      3   0   3     1     0    1    7             0             1
2            2        1      1   1   8     1     0    5    3             1             0
3            3        1      3   1   4     0     0    1    7             0             1
4            4        1      1   1   8     1     0    5    3             0             1
5            5        0      3   0   8     0     0    1    6             0             1
6            6        0      3   0   4     0     0    1    7             1             1
7            7        0      1   0  10     0     0    4    5             0             1
8            8        0      3   0   1     3     1    2    6             0             1
9            9        1      3   1   4     0     2    1    7             0             1
10          10        1      2   1   1     1     0    3    5             1             0
> 
```

Figure 9. First 10 records of the Titanic dataset before and after applying Feature Engineering

4. Conclusion

The current work aimed to create a baseline for Feature Engineering and demonstrate its implementation. Firstly, an analysis of the many different concepts regarding Feature Engineering was performed. Based on this set of notions, a pipeline and taxonomy of the most relevant Feature Engineering techniques were created. The techniques were segregated by dataset issue. A simple R language implementation was developed for each technique and applied to a real-world dataset. The resulting dataset was lacking the initially identified problems, validating the developed techniques and the Feature Engineering process. Altogether, this work was successful in accomplishing the goals established early on.

References

Bastian, N., Maxwell, P., & Alhajjar, E. (2019). Intelligent Feature Engineering for Cybersecurity.

Das, S., & Cakmak, U., M. (2018). Hands-On Automated Machine Learning: A beginner's guide to building automated machine learning systems using AutoML and Python. Packt Publishing Ltd.

Domingos, P. (2012). A few useful things to know about machine learning. Commun. ACM 55, 10 (October 2012), 78–87.
https://doi.org/10.1145/2347736.2347755

Gardner, S. A. (1992). Spelling Errors in Online Databases: What the Technical Communicator Should Know. Technical Communication, 39(1), 50–53. http://www.jstor.org/stable/43095181

Gupta, R. (2020). 8 Clutch Ways to Impute Missing Data. Towards Data Science.

Kersting, K. (2018). Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behavior in Machines, Frontiers in Big Data 1.

Pombinho, P., Cavique, L., & Correia, L. (2021). Qualidade de Dados em Bases de Dados Anonimizadas: Uma Abordagem de Avaliação Mista. Boletim SPE - Sociedade Portuguesa de Estatística

Roh, Y., Heo, G., & Whang, S., E. (2021). A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328-1347. https://doi.org/10.1109/TKDE.2019.2946162

SocialCops. The Ultimate Guide to Basic Data Cleaning. Atlan.

Zheng, A., & Casari A. (2018). Feature Engineering for Machine Learning. O'Reilly Media, Inc.

https://www.kaggle.com/competitions/titanic/data

http://campus.lakeforest.edu/frank/FILES/MLFfiles/Bio150/Titanic/TitanicMETA.pdf

Mariana Teixeira é licenciada em Informática e Gestão de Empresas pelo ISCTE-IUL (2020) e obteve o grau de Mestre em Informação e Sistemas Empresariais pelo Instituto Superior Técnico (2023). Atualmente, trabalha na área da integração de sistemas, com especialização em webMethods. Os seus interesses recaem na exploração de diferentes tecnologias e das diversas possibilidades que lhes estão associadas.

Luís Cavique, Professor Auxiliar no Departamento de Ciências e Tecnologia (DCeT), Secção de Informática, Física e Tecnologia (SIFT). Licenciado em Engenharia Informática pela FCT-UNL. Obteve o grau Mestre em Investigação Operacional e Eng. Sistemas pelo IST-UTL. Obteve o grau de Doutor em Eng. Sistemas pelo IST-UTL em 2002. Tem como áreas de interesse, a intersecção da Informática com a Engenharia de Sistemas designadamente a área de Data Science.