CS5014 Machine Learning

Practical 1 Report

University of St Andrews – School of Computer Science

Student ID: 150014151

12th March, 2020

A close up of a logo

Description automatically generated

Table of Contents

[1 Introduction 3](#_Toc34868497)

[1.1 Usage instructions 3](#_Toc34868498)

[1.2 Tools used 3](#_Toc34868499)

[1.3 Project organisation 3](#_Toc34868500)

[2 Methodology & Design Decisions 4](#_Toc34868501)

[2.1 Initial data loading and data train/test split 4](#_Toc34868502)

[2.2 Data visualisation and analysis 5](#_Toc34868503)

[3 Evaluation 8](#_Toc34868504)

[4 Conclusion 9](#_Toc34868505)

[References 10](#_Toc34868506)

[Appendix A: Random scatter plot for stratified example 11](#_Toc34868507)

1. Introduction

todo

* 1. Usage instructions

Before running the program, create a new virtual environment before installing the Python libraries used by running the following command:

1. pip install -r requirements.txt

To run the program, move to the “*src*” directory and run the following command.

1. python main.py

where:

* 1. Tools used
* Programming language: Python 3.7.
* Python libraries used: Scikit-Learn[[1]](#footnote-1), Pandas, Matplotlib[[2]](#footnote-2) and Numpy.
* Editor: PyCharm[[3]](#footnote-3).
* Version controlling: Git and GitHub (private repository).
  1. Project organisation

The project is organised into the following python modules: todo

The program runs in six distinct steps:

<https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f>

<https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4>

<https://towardsdatascience.com/data-preparation-for-machine-learning-cleansing-transformation-feature-engineering-d2334079b06d>

<https://towardsdatascience.com/polynomial-regression-bbe8b9d97491>

1. Methodology & Design Decisions
   1. Initial data loading and data train/test split

The dataset used is provided by the CSV file related to the journal paper “A data-driven statistical model for predicting the critical temperature of a superconductor” [1]. This file is loaded into a Pandas DataFrame[[4]](#footnote-4) and immediately split into a training and a testing set.

Before even viewing the data’s features and values, the dataset is split in a training and a testing set to avoid any form of data snooping. Data snooping corresponds to the practice of making (either voluntary or involuntary) design decisions after viewing the data and detecting patterns that could lead to favouring certain models or hyperparameters above others. The opening sentence Halbert White’s abstract in his paper on data snooping reveals how any form of data snooping could cause the final results to be questionable, thus reiterating the importance of splitting the data into two separate sets and forgetting about the testing data until the final evaluation [2]:

﻿ “*Data snooping occurs when a given set of data is used more than once for purposes of inference or model selection. When such data reuse occurs, there is always the possibility that any satisfactory results obtained may simply be due to chance rather than to any merit inherent in the method yielding the results*”.

After considering various options to split the dataset, including random splits and stratified sampling. Stratified sampling consists in maintaining representative samples from the data in both the training and the testing sets to avoid sampling bias. To demonstrate sampling bias, an example using the random numbers generated in Figure 1 below (see Appendix A for code used to generate chart), under the assumption that there are four important sections in the data, the split needs to extract samples from each of the four sections to be considered as representative. If the training set contains samples from three sections and the testing data contains samples for the fourth one, then the model will not generalise well to the unseen data as it was not trained on representative data.

A screenshot of a cell phone

Description automatically generated

Figure 1: section separation and test sampling colouring performed in painting application.

However, stratified sampling is not always necessary, as random sampling is considered to be sufficient when the data is large enough. Aurélien Géron mentions that stratified sampling is necessary for small datasets of about 1000 rows [3], and because the dataset used in this practical contains 21,263 rows, random sampling is therefore used instead. Additionally, stratified sampling requires strong knowledge of the dataset [3], which is not the case with this dataset as there are 81 features in total, and no unique feature is considered more important than the other (8 features are mentioned to be important) [1], thus rendering the task of performing a representative split of the data based on the most important feature impossible.

The dataset is therefore split using a randomised 80%/20% split, with the random number generator’s seed set to a constant to ensure that it always generates the same shuffle indices when running the code multiple times to ultimately ensure reproducibility. The *shape* attribute of the Pandas DataFrame is used to double-check that the data was correctly split in a 80%/20% split (the training set size should be and the testing set size should be , which is the case).

1. from sklearn.model\_selection import train\_test\_split
2. train\_set, test\_set = train\_test\_split(df, test\_size=0.2, random\_state=42)
3. print("Train set size = {}, Test set size = {}".format(train\_set.shape, test\_set.shape))
4. #Output: *Train set size = (17010, 82), Test set size = (4253, 82)*

At this stage, with the data is separated into a training set and a testing set, only the training set is considered until the final evaluation, while the testing set is set aside and forgotten about.

* 1. Data visualisation and analysis

Before deciding how to fit a regression model to the training set, it must first be visualised and analysed to gain a better understanding of the data. To secure the training set from any undesired modifications, a copy of the training set, called the exploration set, is created and used for data visualisation.

**Data overview**

Storing the data in Pandas DataFrames provides functions to quickly visualise the data and gain some high-level insights. Using *info* function to receive a summary of the data reveals some information, such as:

* There are no empty values in the entire dataset (each feature has 17,010 non-null values, the size of the training set)
* All values are numbers (either floats or integers)

1. exploration\_set = train\_set.copy()
2. exploration\_set.info()

A close up of text on a white background

Description automatically generated

…

A screenshot of a cell phone

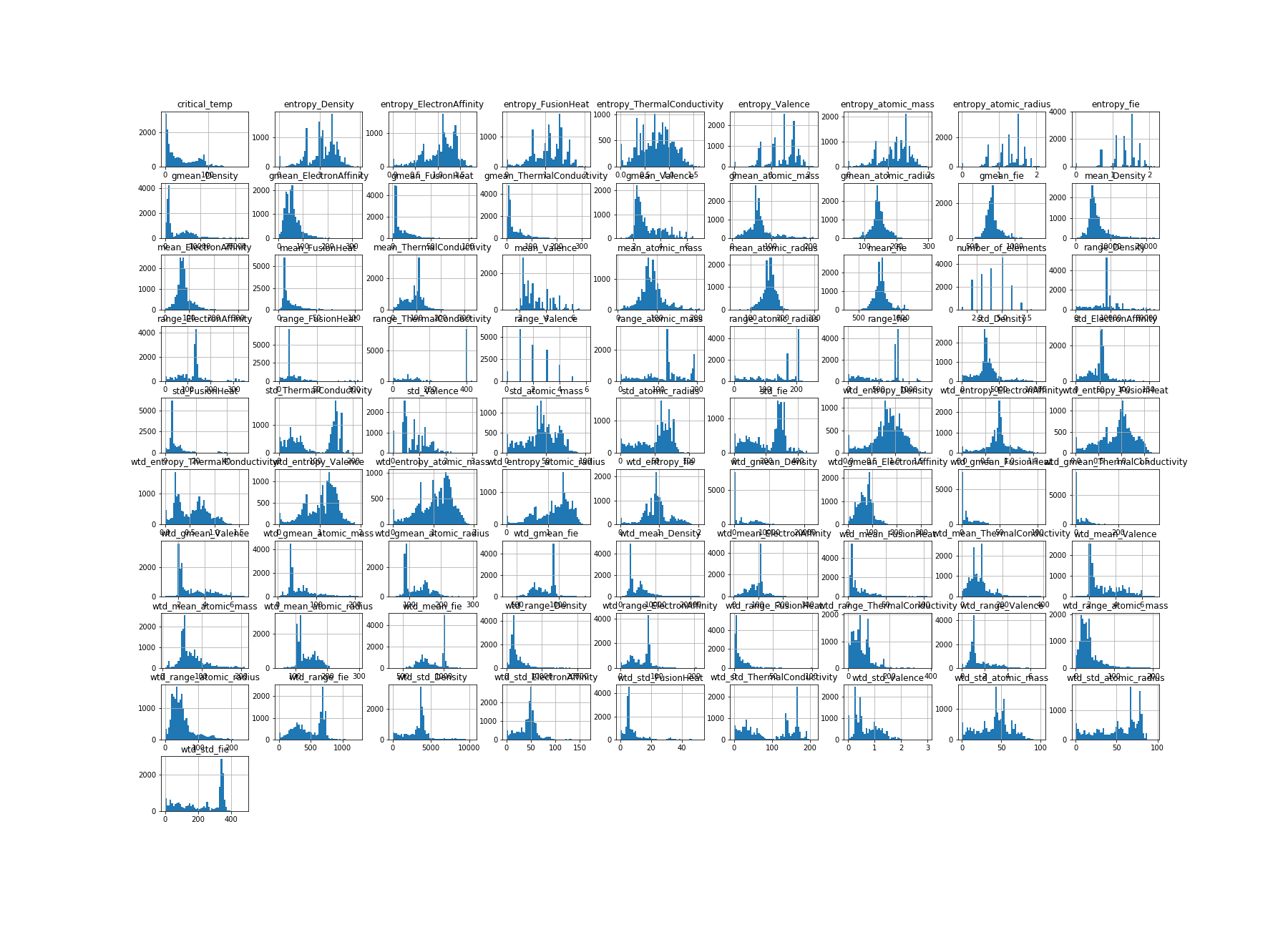
Description automatically generated

These pieces of information mean that no data encodings for non-numerical inputs are required, and no data cleaning operations are required as the data is already clean (no missing numbers).

**Data visualisation**

Now that a general understanding of the data is acquired, more detail can be gathered from visualising the spatial distribution of each of the 81 features and the target column. This can be done by plotting each feature into individual small histograms (see Figure X). This visualisation reveals two interesting characteristics about the data:

* Not all of the features’ distributions are bell-shaped, some have a concentration of data points around specific regions (either for low values or high values at the edges of the distribution bins). This could indicate that feature transformation would be required to achieve more bell-shaped distributions.
* Observing the y-axes of the histograms, the features seem to have different scales, which may suggest that feature scaling is required.



* 1. Feature selection

The data contains 81 different features. Although the paper mentions the eight most important

The correlation matrix is computed to identify the correlation between each feature and the target value. The closer it is to 1, the stronger the correlation is, and vice versa when the correlation is close to -1. A correlation of 0 can be translated as a lack of linear correlation

* 1. Input preparation

Todo

* 1. Training

Todo

* 1. Model fine-tuning

Todo

1. Evaluation

Evaluation of the final model

1. Conclusion

todo

References

[1] K. Hamidieh, “A data-driven statistical model for predicting the critical temperature of a superconductor,” *Computational Materials Science*, vol. 154, pp. 346–354, Nov. 2018.

[2] H. White, “A reality check for data snooping,” *Econometrica*, vol. 68, no. 5, pp. 1097–1126, Sep. 2000.

[3] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, 2nd ed. O’Reilly Media, 2019.

Appendix A: Random scatter plot for stratified example

1. import random
2. def random\_plot():
3. random.seed(0)
4. x = [random.randint(0,100) for i in range(0, 100)]
5. y = [random.randint(0,100) for i in range(0, 100)]
6. plt.scatter(x,y)
7. plt.savefig("plot\_stratified\_example.png")
8. plt.show()
9. random\_plot()

Generated image:

A screenshot of a cell phone

Description automatically generated

1. Scikit-learn: <https://scikit-learn.org> [↑](#footnote-ref-1)
2. Matplotlib: <https://matplotlib.org/> [↑](#footnote-ref-2)
3. PyCharm: <https://www.jetbrains.com/pycharm/> [↑](#footnote-ref-3)
4. Pandas DataFrame: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html> [↑](#footnote-ref-4)