Regional Explanations for XAI

Author
Pernille Matthews

Supervisor
Arthur Zimek

1st June, 2022

DEPARTMENT OF MATHEMATICS
AND COMPUTER SCIENCE

Abstract

Danish

Efterhånden som teknologien udvikler sig og mængden af data vokser, bliver kunstig intelligens en langt mere naturlig del af hverdagen. Maskinlæringsmodellerne baner sig vejen frem, og bekymringer opstår i afvejningen mellem en models kompleksitet og forståelighed. Forklarende kunstig intelligens (Explainable Artificial Intelligence - XAI) er et område inden for kunstig intelligens, som forsøger at skabe et bedre forhold mellem menneske og maskine ved at udvikle metoder, der skal gøre det nemmere at forstå beslutningerne bag modellerne. I dette speciale undersøges muligheden for at opstille regionale forklaringer ved hjælp af de nyeste XAI-metoder. I undersøgelsen anvendes lighedsmålinger og to klyngemetoder, K-means og HDBSCAN, til at undersøge mulighederne for at etablere regioner. Endelig anvendes de undersøgte metoder i forskellige sammenhænge ved hjælp af to anvendelsestilfælde for at realisere potentialet bag regionale forklaringer.

English

As technology advances and data grows, Artificial Intelligence is becoming a much more natural part of everyday life. Machine Learning models are advancing, and concern lies in the trade-off between a model's complexity and interpretability. Explainable Artificial Intelligence (XAI) is an area within AI which attempts to enable a better human-machine relationship by developing methods to help understand the decisions behind models. This thesis explores the possibility of establishing regional explanations using state-of-the-art XAI methods. The exploration uses similarity measures and two clustering methods, K-means and HDBSCAN, to explore ways to establish regions. Finally, by providing two use cases, the approaches explored are used in different contexts to actualise the potential behind regional explanations.

Acknowledgements

I want to thank all the people who have been supportive and shown guidance during my thesis writing and Data Science journey. Certain people, in particular, I would like to thank:

Professor Arthur Zimek, for going on this XAI journey with me and providing invaluable knowledge and insights. I appreciate the time and effort of each supervision.

My family, for the unconditional support in anything I set out to do.

David, for being my partner in crime. For all the fun endeavours we've been on and the encouragement and joy you always bring.

Balkonen and fellow Data Science students, for being an inspiring bunch of people, for the many lunches and fun that always takes place, thank you.

SDU, for the high quality of teaching and providing a great environment to study and learn.

Table of Contents

Chapter 1

Introduction

This chapter presents an introduction followed by the overall objective which first establishes a problem description with a clarifying problem statement. Then, two subsections follow, these include the aim, hypothesis and outline of the thesis structure in order to enable an overview

1.1 Setting the Scene

Artificial intelligence (AI) is becoming ubiquitous, and in modern times a variation of different types of AI exist, two of which are Machine Learning and Deep Learning. Machine Learning and herein Deep learning models have become a substantial part of many systems and workflows in modern society in the last century. Although both are relatively new technology, their utilisation steadily increases daily, particularly within software solutions [Holzinger et al., 2022]. The usage of such models is evident across a vast range of domains. Both high-risk domains, such as health care, military, finance and law, and low-risk domains, such as online shopping, building and the motor domain, make great use of such models [OECD, 2019, p.16].

More specific use cases include, for example, when individuals search the internet, and a predictive search produces more refined and relevant results. This outcome is made possible by an AI component that can consider factors such as an individual's previous search behaviour, previous products bought, age, gender, and much more. More examples are – chatbots, self-parking cars, email filters, smart replies, medical disease predictions and crime-solving, to name only a few [Nadikattu, 2016].

As AI moves towards ubiquitousness in its use in software, it is because AI can improve crucial areas such as productivity, efficiency, and effectiveness and, at the same time, reduce costs for businesses and individuals. However, there are ethical concerns and fears about its uses in areas such as an individual's data for exploitation and possible discrimination. Many of these concerns are due to AI models being uninterpretable, meaning that the reasoning for a model creating a decision is unknown. Not understanding why a model has produced an outcome is a massive issue, notably within high-risk domains [Molnar, 2022, ch.3.1].

In high-risk domains, such as healthcare, the usage of Machine Learning models is as much about automation as the human-machine relationship, enabling practitioners to make fewer oversights and check their beliefs against a less biased assumption to which a computer contributes.

Explainable Artificial Intelligence (XAI) comes into the picture to enable a better human-machine relationship. XAI aims to improve the relationship between humans and machines to avoid blind trust and enable an understanding of what a model's algorithm (often a black box) accurately does and how.

1.2 Problem Description

Within XAI, there are currently two well-established types of explanations used when extracting information from a given model. The first is local explanations [Molnar, 2022, ch.3.3.4], which provide explanations for a single observation. On the other end of the scale, it is possible to use global explanations [Molnar, 2022, ch.3.3.2]. Global explanations aggregate the local values and provide a generalised set of explanations for the whole model. These two types of explanations are either specific to one individual observation or generalised to all observations, leaving little insight into which information may lie between the two types of explanations. Therefore, granted local and global explanations, one might want to gain insights into regions of explanations.

Currently, no formalised method exists to enable insights into regions of explanations for a model. Subsequently, the main focus of this thesis is to explore a new area, which positions itself between local and global explanations. From this point forward, regional explanations denote this new area. The focus of the exploration explores areas and considerations that may aid the creation of a sophisticated method, similarly to local and global explanations, for regional explanations.

Problem Statement

Local and Global explanations are either specific or generalised; gaining insights into explanations in between is limited.

1.3 Aim and Hypothesis

The thesis aims to explore aspects of regional explanations and investigate whether there is valuable information within this explanation area. Additionally, the insights of this thesis could contribute to further work and valuable reflections on establishing a formal regional explanation method.

Aim

To contribute to the scientific foundations of XAI by exploring the potential of a new form of explanation, namely regional explanations.

Hypothesis

Is it possible to address the current limitations of local and global explanations by exploring regional explanations as an alternative form of explanation.

1.4 Structure of Thesis

The thesis starts with **Chapter 2** which provides insight into different theoretical methods used for exploring regional explanations. **Chapter 3** introduces the datasets and preprocessing of the data and elaborates on relevant terminology. **Chapter 4** uses the methods and datasets introduced in **Chapters 4** and **3** to aid the establishment of visualising spaces and explore concepts of regions and clustering. **Chapter 5** provides use cases for the exploration from **Chapter 4** for two distinct datasets. Finally, **chapters 6**, **7**, **8** provide a discussion, future work and conclusion of the thesis's findings.

Chapter 2

Theoretical Background

This chapter will review the essential concepts used within the thesis. First is a section outlining mathematical preliminaries. Then follows, an overview of relevant machine learning techniques. Next is an introduction to Explainable Artificial Intelligence and two SOTA post-hoc XAI methods and lastly, an introduction to specific visualisation methods.

2.1 Mathematical Preliminaries

This section provides an overview of mathematical definitions and notation within the thesis.

2.1.1 Notation

Notation	Definition
$\boldsymbol{x}_i$	A given point/observation in a dataset $\boldsymbol{D}$.
ϕ_{ij}	A SHAP value from an observation $\boldsymbol{x}_i \in \boldsymbol{D}_E$
y	The true labels of a dataset($\boldsymbol{D}$)
$\hat{y}$	The predicted labels of a dataset($\boldsymbol{D}$)
$\boldsymbol{D}$	Dataset with n observations $\boldsymbol{x}_i$, where $\boldsymbol{x}_i$ has a label $\boldsymbol{y}_i$
$\boldsymbol{D}_{\mathcal{S}}$	Data space equivalent to a dataset $\boldsymbol{D}$ excluding labels $\boldsymbol{y}_i$
$\boldsymbol{D}_E$	Explanation space with explanation values $\boldsymbol{\phi}_i$ for $\boldsymbol{x}_i \in \boldsymbol{D}_{\mathcal{S}}$
$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$	A clustering $\mathcal{C}$ consisting of k clusters(C_k)
C_i	Refers to a cluster, where $C_i \in \mathcal{C}$
T_j	Class or partition
$n_i = \lvert C_i \rvert$	The cardinality of a cluster C_k
$\mathcal{N}(i,j) = n_{ij} = \lvert C_i \cap T_j \rvert$	Cardinality of the intersection of a cluster C_i and a class T_j
$\boldsymbol{\mu}_i$	The mean value of a cluster(C_k)

Table 2.1: Overview of mathematical notation used throughout the thesis

2.1.2 Definitions

Contingency Table

For clustering validation a contingency table is often utilised both to create overviews but also to improve time complexity of e.g, pairwise measures. To provide an example and reference the contingency table for the Iris dataset is seen in Table 2.2.

	Iris-Setosa	Iris-Veriscolor	Iris-Virginica	
	T_1	T_2	T_2	n_i
C_1	0	47	14	61
C_2	50	0	0	50
C_3	0	3	36	36
m_j	50	50	50	$n = 150$

Table 2.2: Example of a contingency table as a notation reference.

Squared Euclidean Distance

As a similarity measurement, the Squared Euclidean Distance is used to put greater weight on objects farther apart. It is defined as:

$$||a - b||_2^2 = \sum_{i=1}^{m}(|(a_i - b_i)|^2) \tag{2.1}$$

where the euclidean distance between a and b is computed and squared to achieve a non-negative value and emphasise larger differences [Virtanen et al., 2020].

2.2 Machine Learning

Machine Learning (ML) is the act of a computer executing one or several algorithms to produce an output. This output depends on the type of model and learning.

There are many types of learning within ML, such as *supervised* learning, *unsupervised* learning, *semi-supervised* learning, *reinforcement* learning and more. The scope of this thesis utilises *supervised* and *unsupervised* learning. The following subsections look at *supervised* and *unsupervised* learning and their different use cases.

Supervised Learning

Supervised learning uses labelled datasets to train (or supervise) a model to classify or predict particular values using a labelled dataset. Within *supervised* learning, two types of problems occur:

1. **Classification problems** occur when wanting to split data into categories.
2. **Regression problems** occur when wanting to predict numerical values.

Unsupervised Learning

Unsupervised learning uses unlabelled datasets to analyse and cluster data, resulting in finding patterns in data. There are many different types of problems which *unsupervised* learning help solve; two of the most common include:

1. **Clusterings problems** occur when wanting to find similarity, market segmentation, and more in data.

2. **Association problems** occur when wanting to find relationships between the data; well-known methods include sequence mining and association rule mining.

The scope of this thesis investigates and utilises *supervised* learning problems for *classification* and *unsupervised* learning problems for *clustering*. The following sections take a look at specific *supervised* and *unsupervised* algorithms.

2.3 Unsupervised Clustering Algorithms

2.3.1 K-Means

K-means is an unsupervised non-parametric hard clustering algorithm that allows partitioning a dataset into k clusters. For each observation, $\boldsymbol{x}_j \in \mathcal{D}$ must belong to one k cluster. The observations assigned to a given k cluster are similar but differ significantly from neighbouring clusters. Each k cluster has a corresponding k centroid. The alternative denotation of a centroid is cluster representative or the mean($\boldsymbol{\mu}_i$) of a cluster.

To determine which observations belong to each k cluster, the minimum Euclidean distance from each $\boldsymbol{x}_j$ to each centroid determines the assignment. For the basic K-means algorithm, the initialisation of centroids occurs at random. Once all observations are assigned a centroid, the next iteration of the algorithm proceeds by adjusting the value of the centroids by calculating the mean value provided by the observations assigned to the centroids. The mean of a cluster is calculated in Equation 2.2 by:

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{x_j \in C_i} \boldsymbol{x}_j \tag{2.2}$$

where $n_i = |C_i|$, so the mean value is calculated by the summation of each observation $\boldsymbol{x}_j$ and normalising with $\frac{1}{n_i}$ [Zaki and Meira, 2020]. For each observation $\boldsymbol{x}_j \in \mathcal{D}$ the minimum distance from $\boldsymbol{x}_j$ to each mean $\boldsymbol{\mu}_i$ determines which cluster $\boldsymbol{x}_j$ is assigned to. Each iteration of K-means consists of two steps, 1) assign all the points to the closest cluster centroid, 2) recompute the centroid of newly formed clusters. [Tan et al., 2020, p. 317].

Depending on the centroids' initial placement, the K-means algorithm can converge differently given the same $\mathcal{D}$. Therefore, to find an optimal K-means clustering, it is beneficial to compare the clusterings based on the sum of squared errors(SSE). The lower the value of the SSE, the more optimal the clustering is. The definition for SSE is:

$$SSE(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in \mathcal{C}_i} ||\boldsymbol{x}_j - \boldsymbol{\mu}_i||^2 \tag{2.3}$$

where for each $\boldsymbol{x}_j \in \mathcal{C}$ the Euclidean distance to the nearest centroid is used to compute the SSE.

k	2	3	4	5	6	7	8	9
Silhouette Scores	0.681	0.553	0.498	0.489	0.365	0.35	0.357	0.339

Table 2.3: Silhouette Coefficients per k

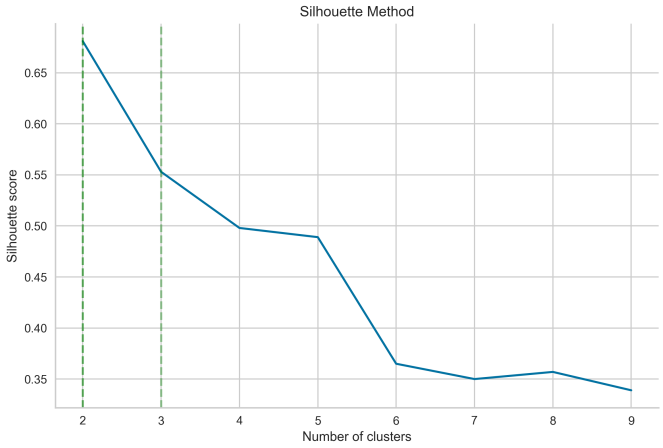


Figure 2.1: Line chart over Silhouette scores(y-axis) as k cluster(x-axis) increases.

Selecting a k value

Another critical aspect of K-means is determining a k value as input. The k value for K-means determines the separation of the data into distinct clusters. *"Eyeballing"* the k value may be a viable option when dimensions < 3, as the data structure is detectable. Once dimensions are > 3, it is no longer possible to "eyeball" the k value due to not being able to visualise dimensions > 3. Therefore, formal methods exist to help determine the k value of a dataset.

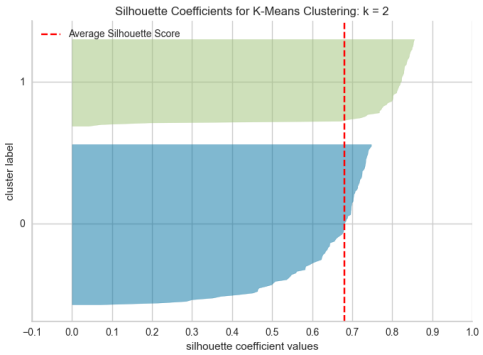
The Silhouette Method

One formal method to determine the k value for K-means granted a dataset uses the Silhouette Coefficient (described in detail in section 2.5.2). The first and more simplistic way of determining the k value is utilising the Silhouette Coefficient for a range of k values.

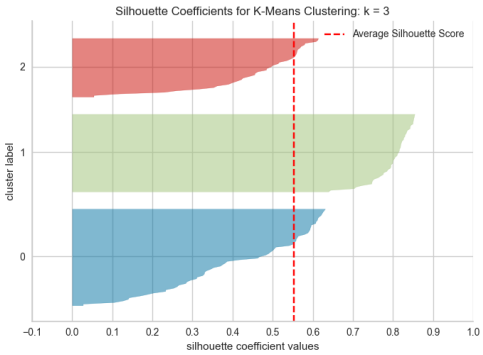
Table 2.3 below shows the Silhouette Coefficients for K-means from the dataset Iris (described in section 3.1). The Iris dataset consists of 3 classes; therefore, a common assumption may be that k=3 would be the appropriate k value. Table 2.3, Figure 2.1 shows the Silhouette Coefficients on the y-axis and each corresponding k cluster on the x-axis. It is evident that as k increases, the silhouette scores for the Iris dataset decrease. The two top silhouette scores are highlighted green on the table and correspondingly two vertical green dotted lines on the figure.

Moreover, *scikit-learn* has implemented an extended way of visualising the Silhouette Method, a graphical aid technique created by *Peter J. Rousseeuw* in 1986 [Rousseeuw, 1987]. The extension calculates each sample's silhouette score within its respective cluster. Then, the density and individual silhouette scores per sample are plottable for each cluster. The density of each cluster provides much insight when deciding on a suitable k. Figure 2.1 above shows that k= 2 or k= 3 is appropriate, and when looking at the visualisation below in Figure 2.2, the same can be derived.

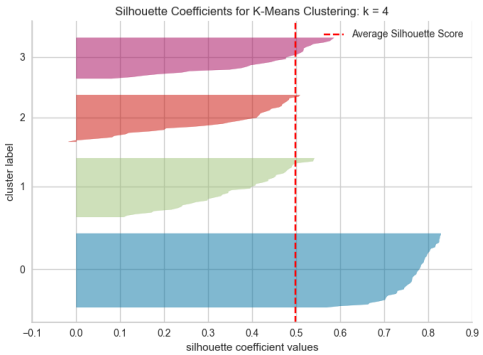
Comparably, in Figure 2.1 to Figure 2.2, the density of each cluster is visible. An ideal result is clusters of uniform shape. It is evident in sub-figure **(c)** and **(b)** in Figure 2.2 that



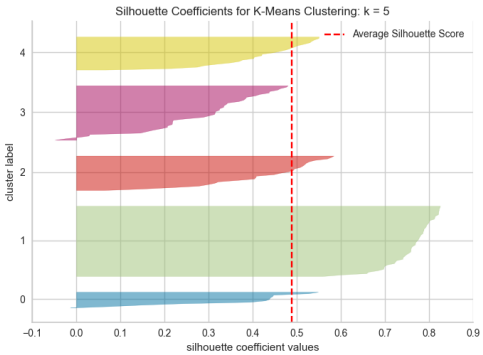
(a) Silhouette Coefficients where k= 2



(b) Silhouette Coefficients where k= 3



(c) Silhouette Coefficients where k= 4



(d) Silhouette Coefficients where k= 5

Figure 2.2: Comparison of Silhouette Coefficients for K-means as k increases.

the clusters differ significantly in thickness, and several clusters score below the average silhouette scores. However, when looking at **(a)** and **(d)**, the shape of the clusters and average silhouette scores are more comparative. Interestingly the first cluster in subplot **(a)** is very thick, suggesting that $k = 3$ would be a more satisfactory k value as the observations are more evenly distributed across clusters.

2.3.2 HDBSCAN

Hierarchical Density-Based Spatial Clustering of Applications with Noise or HDBSCAN is another unsupervised learning clustering algorithm and, unlike K-means, uses Density and Hierarchical based clustering techniques. Compared to K-means, which excels at finding ellipsoid-shaped clusters, HDBSCAN can detect non-convex clusters. This subsection will first explore DBSCAN and then HDBSCAN.

DBSCAN

DBSCAN [Ester et al., 1996] is a density-based clustering algorithm that can detect clusters of differing shapes and sizes. DBSCAN estimates the density for each observation using a centre-based approach. The algorithm requires an epsilon(ε) value and minimum points(*MinPts)* as input.

The ε value is a radius for an observation x, where the radius defines the neighbourhood for x, known as the ε-neighbourhood and formally written as:

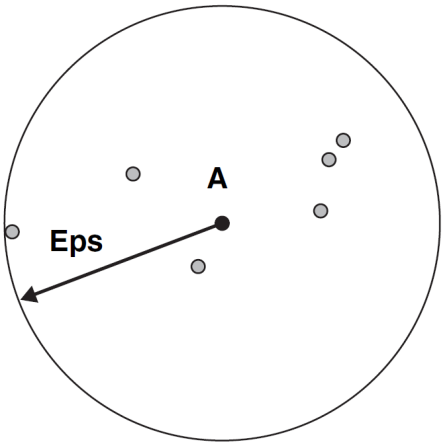


Figure 2.3: Visual representation of observation **A**, where the ε value results in **A** has 7 points in its neighbourhood [Tan et al., 2020, p. 348].

$$N_\varepsilon(\boldsymbol{x}) = \{y \mid \|\boldsymbol{x} - \boldsymbol{y}\| \leq \varepsilon\} \tag{2.4}$$

where ε declares the neighbourhood radius for x, if the euclidean distance of a point y from x is $\leq \varepsilon$, then y falls within the $N_\varepsilon(\boldsymbol{x})$ of x. Thus, each observation uses ε to determine how many points fall within the radius, which specifies x's density. As illustrated below in Figure 2.3, the neighbourhood includes the observation itself, where point A has 7 points within its neighbourhood.

Moreover, given that two points are within ε-distance of each other, they are neighbours; however, this does not necessarily mean that a cluster is formed. Forming a cluster also requires fulfilling each observation's *MinPts* argument. DBSCAN labels observations as one of three types of points described below, and depicted in Figure 2.4:

1. **Core points** are when points have $\geq$ *MinPts* within an ε radius.
2. **Border points** are when points are not core points but within the ε radius of a core point.
3. **Noise points** are when points do not qualify as core or border points. Noise points are often outliers.

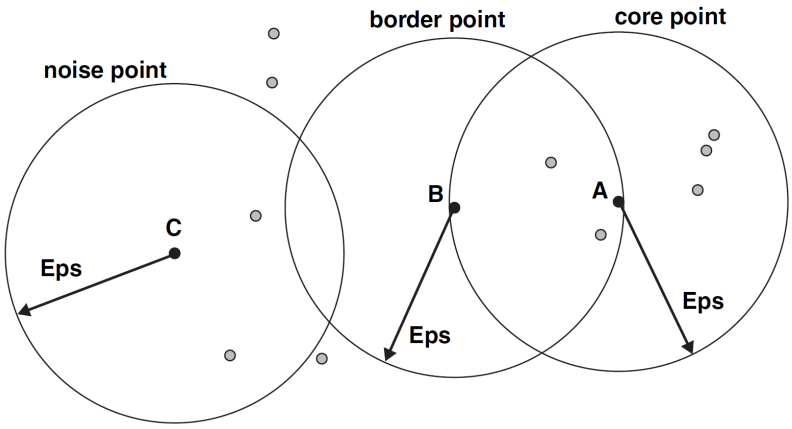


Figure 2.4: Types of labelled points, noise, border and core. [Tan et al., 2020, p. 348]

A high-level description of the DBSCAN algorithm is provided in the pseudocode 1 [Tan et al., 2020, p. 349]:

Algorithm 1 DBSCAN algorithm

1: Label all points as core, border, or noise points
2: Eliminate noise points
3: Put an edge between all core points within a distance of *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

A crucial aspect of DBSCAN is setting an appropriate ε and *MinPts*. Determining the ε value depends on the data structure. A very large ε would result in many large clusters ignoring potential smaller dense clusters. In contrast, a low value would result in many small clusters and potentially many noise points. Formal methods exist to help detect an appropriate ε, such as Ordering Points To Identify the Clustering Structure(OPTICS), where a reachability plot is able to provide insights into the structure and thus, approximate an ε value [Ankerst et al., 1999].

Determining the *MinPts* parameter, in the original DBSCAN paper [Ester et al., 1996] it is argued that when dimensions of data $= 2$ then a general $MinPts = 4$ suffices.

HDBSCAN

HBDSCAN is an extension of DBSCAN by Campello et al. [Campello et al., 2013], which only requires the minimum cluster size as a single input parameter. Parameter selection for HDBSCAN cannot be generalised and is dataset dependent. HDBSCAN is able to add the hierarchical aspect to DBSCAN and make a decision as to what level of the cluster hierarchy should be chosen to extract clusters.

Within a Python implementation of HDBSCAN [McInnes et al., 2017], it is possible to validate a given HDBSCAN clustering by using relative validity, introduced in section 2.5.2. Moreover, it is possible to project the clustering and the changes of hyperparameters using a t-SNE visualisation, introduced in section 2.8.3. The main steps for HDBSCAN are seen in algorithm 2.

Algorithm 2 HDBSCAN main steps from Tan et al.[Tan et al., 2020]

1: 1. Compute the core distance w.r.t m_{pts} for all data objects in $\boldsymbol{X}$.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster
5: **until** Centroids do not change.

2.4 Supervised Classification Algorithms

This section briefly summarises two types of classification algorithms; logistic regression and Random Forest.

2.4.1 Logistic Regression

Within this thesis, the focus is on exploring classification problems. This means that the XAI methods, introduced shortly, use logistic regression when estimating explanations for a classification dataset. The overall goal of logistic regression is to take an input and

produce an output in the form of probability by compressing the output into a range between $[0:1]$. To do this the following function is essential:

$$\theta(z) = \frac{1}{1 + exp\{-z\}} \tag{2.5}$$

The function is known as the Sigmoid function [Zaki and Meira, 2020, p.624], where $\theta(z)$ is the output probability. During training it is the dependent variable which represents the categorical value that is trying to be predicted. For binary classification it is the probability of an observation belonging to class 0 or 1.

2.4.2 Random Forest

The algorithm used for establishing models for the datasets within the thesis is Random Forest. Random Forest is known as an uninterpretable model compared to e.g. decision trees. The basic concept of the random forest algorithm is to establish an ensemble of uncorrelated decision trees with feature randomness by using bagging [Zaki and Meira, 2020, p.574]. To decide on the quality of a split in a random forest, the model typically uses either information gain or Gini Index. The approach of information gain is to split features based on the least entropy [James et al., 2013, p.336]. The alternative is to use the Gini index which uses the variance across classes to determine which class is most dominant for a given split [Tan et al., 2020, p.51].

2.5 Clustering Validation

Assessment of a clustering is important as it provides insight into how well data has been clustered. To asses or validate a clustering, Zaki et al.[Zaki and Meira, 2020, ch.17] defines three categories of validation, external, internal and relative; wherein different validation measures exist:

1. **External validation** measures uses the external true y labels to validate the how well cluster labels/predicted labels $\hat{y}$ match.
2. **Internal validation** measures the goodness of a clustering without considering the external information.
3. **Relative validation** measures look to compare two different clusterings or clusters.

2.5.1 External Validation

Within external validation methods lies the matching based measures. The thesis employs three matching based measures: Purity, Normalised Mutual Information (NMI), and Adjusted Rand Index (ARI).

Purity

Purity looks at how a clustering $\mathcal{C}$ or cluster C_i contains mixed or homogeneous class assignments. The following formula calculates the purity of a single cluster:

$$\text{purity}_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\} \tag{2.6}$$

where $n_{ij} = |C_i \cap T_j|$ corresponds to the majority class for C_i, weighted by the total observations $n_i \in C_i$. To find the purity of a whole clustering, the formula is adjusted [Zaki and Meira, 2020, ch.17]:

$$\text{purity} = \frac{1}{n}\sum_{i=1}^{r}\max_{j=1}^{k}\{n_{ij}\} \tag{2.7}$$

where the max of each class per cluster is chosen and weighted by the total amount of n.

Purity is furthermore usable when comparing different clusterings to analyse the extent of how the purity matches. One drawback of purity is that when a clustering has many small clusters, the chance of maximising purity is high. Therefore, when using purity, an important note is to consider the number and size of clusters.

Normalised Mutual Information (NMI)

Normalised Mutual Information (NMI) [Shannon, 1948] allows for comparison of two clusterings, despite having different amount of clusters. Given two clusterings C_1 and C_2 the conditional entropy of class labels for C_1 and C_2 are calculated. The conditional entropy of each clustering is then used to calculate the mutual information. The MI score becomes larger as clusters increase, and have no upper bounds. To introduce an upper bound the Normalised Mutual Information (NMI) establishes a boundary between [0-1] [Zaki and Meira, 2020, p.433]. For the exploration chapter of the thesis, NMI is used to compare two clusterings, in other words NMI provides a metric for how similar two clusterings are. A high NMI score means that the two clusterings are more similar, while a low score indicates a low similarity.

Adjusted Rand Index (ARI)

The Rand Index [Hubert and Arabie, 1985] is a measurement of the performance of a clustering algorithm. The RI score allows for measuring the performance of one clustering result to another. It compares the actual class labels and cluster labels using pairwise measures [Zaki and Meira, 2020, p.435]. The formula for the RI is:

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.8}$$

where TP+FP+FN+TN is the same as N, which is $N = \binom{n}{2}$ given n is the total number of observations in $\boldsymbol{D}$. However, differing from the confusion matrix used for performance measures of a model [Zaki and Meira, 2020, p.551], calculations for the TP, FP, FN and TN are pairwise. For two points, $x_i, x_j \in \boldsymbol{D}$, the following pairwise events are possible [Zaki and Meira, 2020, p.435]:

$$\text{TP} = |\{(x_i, x_j) : y_i = y_j \wedge \hat{y} = \hat{y}\}| \tag{2.9}$$

$$\text{FN} = |\{(x_i, x_j) : y_i = y_j \wedge \hat{y} \neq \hat{y}\}| \tag{2.10}$$

$$\text{FP} = |\{(x_i, x_j) : y_i \neq y_j \wedge \hat{y} = \hat{y}\}| \tag{2.11}$$

$$\text{TN} = |\{(x_i, x_j) : y_i \neq y_j \wedge \hat{y} \neq \hat{y}\}| \tag{2.12}$$

where a positive event is $\hat{y} = \hat{y}$ and a negative event is $\hat{y} \neq \hat{y}$. The confusion matrix for pairwise similarity can alternatively, as seen in Table 2.4, be interpreted as:

Moreover, is the Adjusted Rand Index (ARI) which is an extension of the RI which uses the expected similarity of all the permutations possible. While the RI provides a value between $[0 : 1]$ the ARI is able to provide negative values as well. A value of 1 indicates a

TP: same class + same cluster	**FN**: same class + different cluster
FP: different class + same cluster	**TN**: different class + different cluster

Table 2.4: Visual concept of how the confusion matrix is altered for pairwise measures.

perfect clustering while a negative value signals an index less than the expected similarity. In a general form the ARI is as follows [Hubert and Arabie, 1985]:

$$\text{ARI} = \frac{\text{Index} - \text{Expected Index}}{\text{Maximum Index} - \text{Expected Index}}$$

Within this thesis, the ARI metric is as another type of similarity measure when comparing two clusterings, thus, instead of class vs cluster, the formula uses cluster labels vs cluster labels to determine the similarity.

2.5.2 Internal Validation

Two internal validation methods are used for the thesis, namely the Silhouette Coefficient and Density Based Clustering Validation (DBCV).

Silhouette Coefficient

In 1986 the Silhouette method was established by Peter J. Rousseeuw [Rousseeuw, 1987] with the intention of providing a technique that indicates how well objects are clustered in regards to their own cluster (cohesion) and other clusters (separation); by producing a silhouette coefficient. In Figure 2.5 an example is seen with three clusters. When calculating the silhouette coefficient for observation $s(i)$, the average dissimilarity from i to C_A is obtained. Then, the nearest cluster is identified by calculating the average dissimilarity of distance from all observations in C_B to i and the same for C_C. Then, in this scenario, C_B is the nearest cluster and thus, is used to calculate the silhouette coefficient.

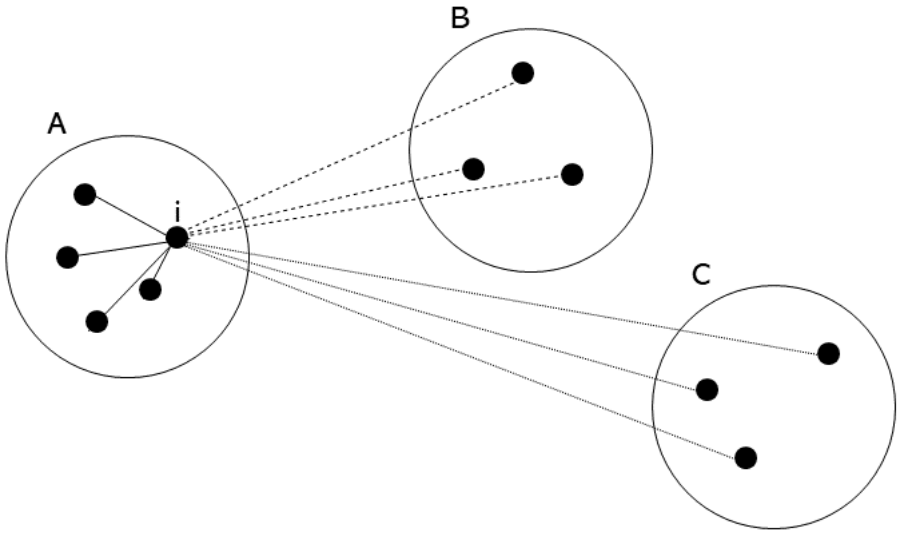


Figure 2.5: Illustration of example of calculating the silhouette coefficient for observation i from the original paper [Rousseeuw, 1987].

The formula for calculating the silhouette is:

$$S(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2.13}$$

where for observation i the average distance from C_B, denoted $b(i)$ is subtracted from the within distance for C_A, denoted $a(i)$, and divided by the maximum average distance to i of the two clusters. The silhouette coefficient provides a score between $[-1 : 1]$ with the following potential extreme scenarios:

- If $|C_A| = 1$, then the silhouette coefficient $s(i)$ is set to zero.
- If the $s(i)$ is near one, then the intra-cluster distance for C_A is significantly smaller than the inter-cluster distance for C_B
- If the $s(i)$ is near zero, then $b(i) \approx a(i)$.
- If the $s(i)$ is near minus one, then $a(i) >> b(i)$ meaning that i is closer on average to $b(i)$ than to $a(i)$, then i has been misclassified.

Density Based Clustering Validation (DBCV)

Density Based Clustering Validation (DBCV) by Davoud et al. [Moulavi et al., 2014] is a validation method for density based clustering which considers noise and works well on arbitrarily shaped clusters. In contrary to the concept for the Silhoeutte Coefficient, DBCV uses density opposed to distance to validate a clustering. The DBCV creates an index which is a score between $[-1 : 1]$. A score close to 1 indicates a better clustering solution.

For this thesis, an implementation of DBCV is used from the Python Library HDBSCAN [McInnes et al., 2017]. The method in the library is denotes DBCV as relative validity, and is implemented as a fast approximation of DBCV where instead of using all-points-core-distance described in the DBCV paper, the implementation instead utilises a mutual- reachability minimum spanning tree. The purpose of using this implementation is to achieve a score that can provide information about the quality of clusterings across a variation of different hyperparameter setups.

2.6 Introducing Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) is a field of Artificial Intelligence that focuses on creating processes and methods for humans to understand machine learning algorithms. Although XAI has existed for a long time, only recently has the term XAI existed. DARPA was the first to name the field XAI after their successful article called "DARPA's Explainable Artificial Intelligence Program", catching much traction [Gunning, 2019]. Within XAI, there are two often interchanged terms, explainability and interpretability. For simplicity's sake, this thesis uses interpretability and explainability interchangeably.

Many approaches exist when wanting to explain an ML model [Holzinger et al., 2022]. To mention some approaches, the first is the post hoc method. Using a post hoc approach means that interpretability of the model occurs after the training process. Secondly, there are model-agnostic methods which are flexible methods able to be applied to any machine learning model and explain that given model. Oppositely model-agnostic methods are those that are limited to a specific category of models. The thesis utilises model-agnostic and model-specific methods.

There are currently two well-defined types of explanations, namely local and global explanations. Local explanations are the concept of investigating a single observation

and extracting explanations locally based on the observations. Local explanations aim to examine what contributed to a given observation's prediction subject to its specific features/inputs. Thus, local explanations are specific to the observation in focus and independent of other observations' explanations.

Next are the global explanations. Global explanations can attain a holistic view of a model and the prediction, allowing an interpretation of the entire model. The overall goal of global explanations is to gain insights into how the model averagely makes decisions. These two definitions are relatively loose, and their methodology differs depending on the method used. The following section will investigate two SOTA methods more carefully and provide precise insights into how local and global explanations are extracted from a given model.

2.7 State-of-the-art XAI methods

Within XAI, there are currently several well-established methods. This section focuses on two of these, namely, LIME [Ribeiro et al., 2016] and SHAP [Lundberg and Lee, 2017b]. For the central part of the thesis, SHAP is the XAI method of choice. As SHAP is built based on LIME, the first part of this section examines LIME, after which follows an examination of the Shapley values and, in turn, SHAP transpires.

2.7.1 LIME (Local Interpretable Model-Agnostic Explanations)

Local Interpretable Model-Agnostic Explanations (LIME) is a model-agnostic explainable method by Ribeiro et al. [Ribeiro et al., 2016]. LIME can take any model and locally explain which features contributed most towards a single observation x. The way LIME can do this is by using surrogate models. A surrogate model can enable the explainability of a machine learning model by approximating the predictions of complex non-linear models, also known as black boxes.

To supply an intuitive understanding of LIME's functionality, Figure 2.6 shows a visualisation from the LIME paper [Ribeiro et al., 2016] picturing the following:

- The blue and pink areas represent a complex decision function denoted as f.
- The bold red cross is the observation in focus, for which local explanations are calculated.
- Local observations (known as perturbations) are generated around x. These are shown by the blue circles and red crosses, where the size determines the closeness of each observation to x.
- The newly established local observations make it possible to fit a linear model, as shown by the dotted black line.

LIME chooses an observation x to explain by *"zooming in"* on x's local vicinity. By looking at the local vicinity of x, it is possible to generate perturbations. Perturbations are new data points sampled from a normal distribution consisting of a mean and standard deviation based on x. Generating perturbations enables the creation of a new dataset, which can then be run through the black-box model and produce predictions. Therefore, the surrogate model approximates the complex model using the local proximity of the observation and fits a local linear model in which explanations are extractable. Mathematically this is done by using the following optimisation formula from the LIME paper:

$$\xi(x) = \underset{g \in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.14}$$

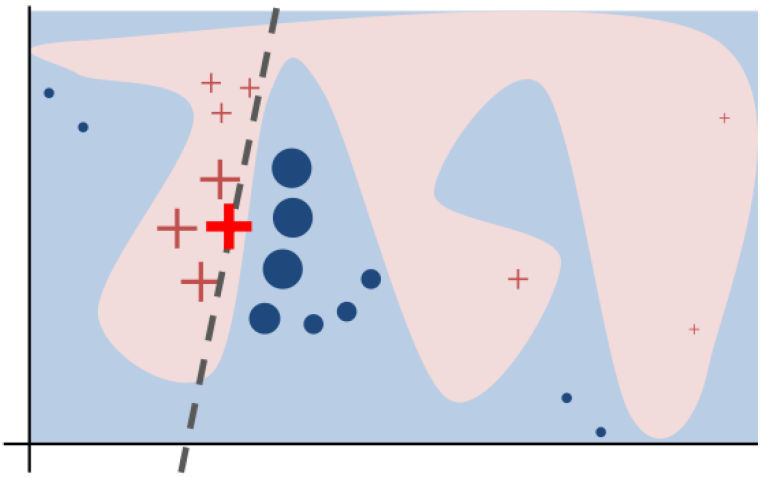


Figure 2.6: Example from LIME paper [Ribeiro et al., 2016] showing the intuition behind using a surrogate model to establish a local model.

The formula is used to create a locally optimised interpretable model, given an observation x. $\xi(x)$ is an interpretable model given input features from observation x. The goal is to minimise the two loss terms $\mathcal{L}(f, g, \pi_x)$ and $\Omega(g)$. The parameters of the first loss term includes a complex model f, local model g and proximity metric π_x. The loss term looks for an approximation of f by the simple model g in x's neighbourhood. The local model g comes from a set of interpretable models G where LIME utilises the sparse linear model by default. The LIME paper provides the following loss function as the first loss term given by a linear model. Below is the formula for the loss function for a linear model:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z,z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z'))^2 \tag{2.15}$$

The goal is to minimise the sum of squared distances given the $\hat{y}$ values from f in comparison to the $\hat{y}$ of the simple model g. Each observation is weighed by the proximity term π_x. π_x is added to the loss according to how close observations are. The proximity metric for LIME uses an exponential kernel by default, meaning that close observations to x are provided greater weights than observations with a further distance. Determining the kernel width (σ) is not clear within the paper, however, within the Python implementation of LIME, there is a default setting of number of columns·0.75 when using the Tabular Explainer [Ribeiro, 2016].

The second loss term, $\Omega(g)$, which is an argument for the complexity of the linear model. The complexity of a linear model, in this instance, is the number of features. Thus, the user determines the second loss term for a linear model by supplying a maximum number of features. Alternatively, for a different model, such as a decision tree, the max depth of the tree would define the complexity of that model.

The overall concept is to look for a simple model g, which minimises the two loss terms to approximate the complex model in the local vicinity while keeping the model as simple as possible. A last note concerning LIME is that the method fulfils what is known as local fidelity or local faithfulness [Molnar, 2022, ch.3.5]. Local fidelity is concerned with how well the model does in the vicinity of an observation. A high local fidelity means that the explanation approximates the prediction of the black model well. While LIME fulfils this property, there are many other properties which the method fails to fulfil; the upcoming section elaborates this notion.

2.7.2 Shapley Values

Before examining the SHAP method, the background concerning the Shapley values occurs to set the scene for how the Shapley values are applicable in ML. Thus, this subsection will first discuss the Shapley values, then the SHAP method.

Background of the Shapley Values

The Shapley values originate from cooperative game theory and are what is known as a solution concept. Named in honour of Lloyd Shapley, the Shapley values were introduced in 1951 and later, in 2021, Lloyd Shapley won a Nobel Prize in the field of Economics for Shapley values [Shapley, 2016].
To grasp a high-level intuition of the Shapley values, imagine the following situation: There are differently skilled players in a cooperative game and a collective reward for completing the game. Due to the players having different skill levels, the individual contribution toward the reward differs; thus, splitting the reward evenly would not result in a fair distribution. Generating a fair distribution amongst the player is possible by using the Shapley values. Establishing Shapley values requires calculations which provide a fair distribution of the collective reward amongst the players. The idea is to calculate the marginal contribution of each player and then take the average to determine how much each player contributed towards the reward.

The definition of the word "fair" is vast, and thus, the Shapley values specify the following axioms to define what a fair distribution must comply with.

Properties of the Shapley Value

- **Symmetry:** If two players contribute the same to the game, then both players contribute the same amount.
- **Null/Dummy player:** If a player contributes nothing, then their share of the reward is also nothing.
- **Additivity:** The distribution of the reward across players sums up to the total reward.

With this intuitive high-level understanding, a tangible example is provided to investigate how the Shapley values are obtained.

Calculating Shapley values

A game is defined as $G = (N, v)$ where N is the set of all players, also known as the grand coalition. v is a characteristic function meaning that for each coalition, the $v(S)$ value is the valuation of the coalition S. A non-empty subset is $S \subseteq N$, and an empty set evaluates as $v(\phi) = 0$. Given a game G, players Alice(A), Bob(B), Claire(C) and Dan(D) have the following coalition values seen in Table 2.5 and a total reward when working together = 10. To calculate each player's average marginal contribution the formula for the Shapley values is introduced:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|\mathcal{S}|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)) \tag{2.16}$$

where $\phi_i(v)$ is the marginal contribution of a given player $\in S$. The summation criterion states that within a set of players, the player in which values are calculated for is excluded. This summation occurs over all permutations of $\mathcal{P}(S)$. For $(v(S \cup \{i\}) - v(S))$ the marginal

value is calculated by adding player i to the game, enabling the comparison of the value with and without the player. Essentially, the idea is to summate the difference in the outcome for each subset with and without the player.

A simplistic example is provided, where each player's marginal contribution towards the reward in a game is calculated. The following values for each coalition are seen below in Table 2.5:

$S \subseteq N$	Coalition values
{∅}	0
{A}	5
{B}	2
{C}	2
{A,B}	3
{A,C}	2
{B,C}	2
{A,B,C}	10

Table 2.5: Example of values for each player in a game (G).

For player A, the following calculation produces the marginal contribution towards the grand coalition as follows:

$$\begin{aligned}\phi_A(V) &= \frac{1}{6}((0!(3-0-1)!)(5-0) + (1!(3-1-1)!)(3-2) \\ &\quad + (1!(3-1-1)!)(2-2) + (2!(3-2-1)!)(10-2)) \\ &= \frac{1}{6}(2\cdot 5) + 1 + 0 + (2\cdot 8) = \frac{1}{6}\cdot(27) = \frac{9}{2} = \underline{\underline{4.5}}\end{aligned}$$

for player B, the marginal contribution is:

$$\begin{aligned}\phi_B(V) &= \frac{1}{6}((0!(3-0-1)!)(2-0) + (1!(3-1-1)!)(3-5) \\ &\quad + (1!(3-1-1)!)(2-2) + (2!(3-2-1)!)(10-2)) \\ &= \frac{1}{6}\cdot 4 + (-2) + (2\cdot 8) = \frac{1}{6}\cdot(18) = \underline{\underline{3}}\end{aligned}$$

and lastly, for player C, the marginal contribution is:

$$\begin{aligned}\phi_C(V) &= \frac{1}{6}((0!(3-0-1)!)(2-0) + (1!(3-1-1)!)(2-5) \\ &\quad + (1!(3-1-1)!)(2-2) + (2!(3-2-1)!)(10-3)) \\ &= \frac{1}{6}\cdot 4 - 3 + 0 + (2\cdot 7) = \frac{1}{6}\cdot(15) = \underline{\underline{2.5}}\end{aligned}$$

It is thus, evident that summating the marginal contributions of each player, $4.5 + 3 + 2.5 = 10$, results in the reward given when all three players are in the coalition also seen in previous Table 2.5. Calculating the Shapley values comes at a great cost as each n members/players in a game/group have 2^n subsets; computationally very expensive. The upcoming SHAP method however, has a way to approximate the Shapley values, reducing the time complexity of the computations.

In the context of Machine Learning, the differently skilled players are equivalent to the features/input values, while the collective reward equates to the model prediction(s). The game can be considered the model which produces the reward or the model's outcome. With this in mind, the Shapley values in a Machine Learning setting quantify the average marginal contribution of each feature to the prediction made by the model.

2.7.3 SHAP (SHapley Additive exPlanations)

The previous subsections have led up to this moment of investigating the SHAP method. SHAP stands for SHapley Additive exPlanations, and in 2017 Scott Lundberg and Su-In Lee published a paper called: "A Unified Approach to Interpreting Model Predictions", where SHAP was introduced [Lundberg and Lee, 2017b].

The SHAP implementation uses an explainer to generate the expected value / base value for each prediction. The main explainer is the KernelSHAP which is model-agnostic [Molnar, 2022, ch.9.6.2]. KernelSHAP uses the same concept as LIME as it establishes a surrogate model to estimate a given observation. The parameters for the KernelExplainer is the predicted probability of the model and the dataset in which the explainer is based on. Downsides to the KernelExplainer is that it is an extremely slow method; particularly, when features are large or much data exists, the time complexity is significant. Another explainer is the TreeExplainer, a model-specific explainer for tree-based models. The TreeExplainer is much faster than KernelSHAP but is known to produce unintuive feature attributions [Molnar, 2022, ch.9.6.3].

To establish the explanations in this exploration, the KernelSHAP was used on all datasets because of the unstableness of the TreeExplainer. The datasets used within the thesis were not large enough to detect any not being time complexity issues when using KernelSHAP.

2.8 Visualisations

An introduction to relevant visualisations proceeds in this section. First two visualisations from the SHAP Python Library [Lundberg and Lee, 2017b] are presented to show how a local and global explanation are able to be interpreted visually. Then, a different type of visualisation is presented, namely the Parallel Coordinates Plot which is used significantly during the exploration of regional explanations in section 4.5 chapter 5. Lastly, the t-SNE plot is introduced, which is used to help visualise how clustering of HDBSCAN can be projected and interpreted, when tuning the different hyperparameters.

2.8.1 SHAP visualisations

The Python SHAP library [Lundberg and Lee, 2017a] has over time created many plots that enable visualisation of explanations. The two most notable are the Force and Waterfall plots.

Local Force Plot

The force plot in Figure 2.7 shows an individual observation from the Skin Segmentation dataset [Dua and Graff, 2017], consisting of 3 features: "B", "G", "R". The SHAP values used are towards class 2 (No Skin) which has the base value of 0.79 also illustrated in the figure. Notice how each feature values acts like forces which increase or decreases the outcome of the prediction. Positive value (red bars) means a contribution towards the prediction, while negative values (blue) push away from the prediction.

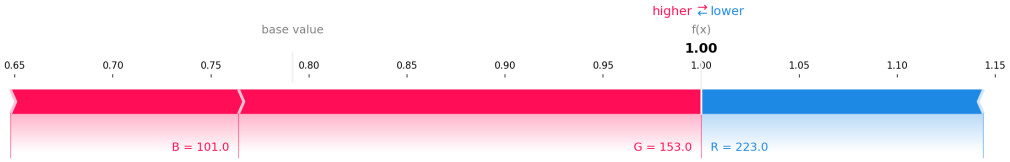


Figure 2.7: Observation 1 from Skin Segmentation dataset [Dua and Graff, 2017] showing how the force plot can depict a local explanation.

Global Bar Plot

While the force plot visualises individual predictions of observations, this next visualisation, the bar plot, shows how SHAP values can contribute globally. For SHAP, the bar plot is able to provide insight into the feature importance across the whole model by averaging the Shapley values per features, as seen in Figure 2.8.

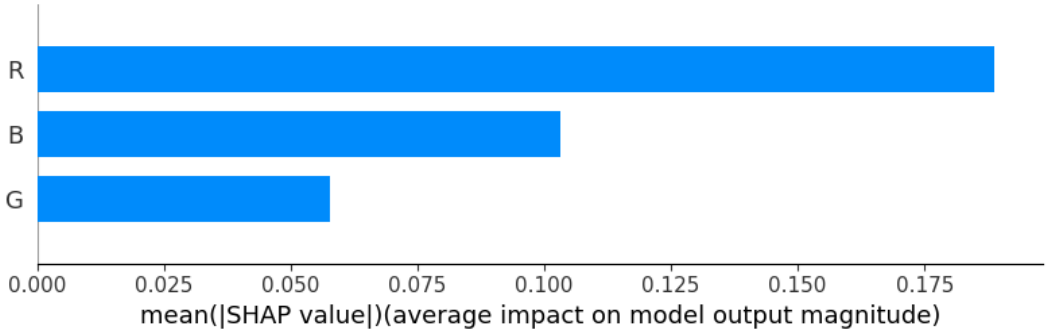


Figure 2.8: Global feature importance of Skin Segmentation dataset [Dua and Graff, 2017] showing how the each features average impact on the models output.

The SHAP Python Library also provide a lot of other types of visualisation besides these selected two, but for the scope of the thesis, these were the selected two.

2.8.2 Parallel Coordinates Plot (PCP)

A Parallel Coordinates Plot [Johansson Westberg and Forsell, 2015] from now on denoted as PCP is an interesting type of visualisation that enables a way to visualise high-dimensional data. The plot maps each observation in the dataset as a vertical line across the x-axis. Each vertical line on the x-axis is a feature and on the y-axis the values are provided. A PCP can work on features of same numeric value and scale, but also with features of different units.

The order of the axes can have great impact on how the visualisation portrays the data, and how easy the plot is to read. Reordering features can therefore impact the perception of the data. Overall the PCP is good at comparing many quantitative features and analysing patterns and relationship.

One notable downside to the PCP includes over-cluttering of lines given datasets with many observations. This issue can be solved by applying a technique called brushing which highlights lines which overlap and makes one thicker or darker line. Looking at Figure 2.9 the dataset Iris, introduced in Section 3.1, is shown. Each observation is represented through a connected line segmented across the x-axis. Every class is represented by a colour, enabling insight into any patterns across classes.

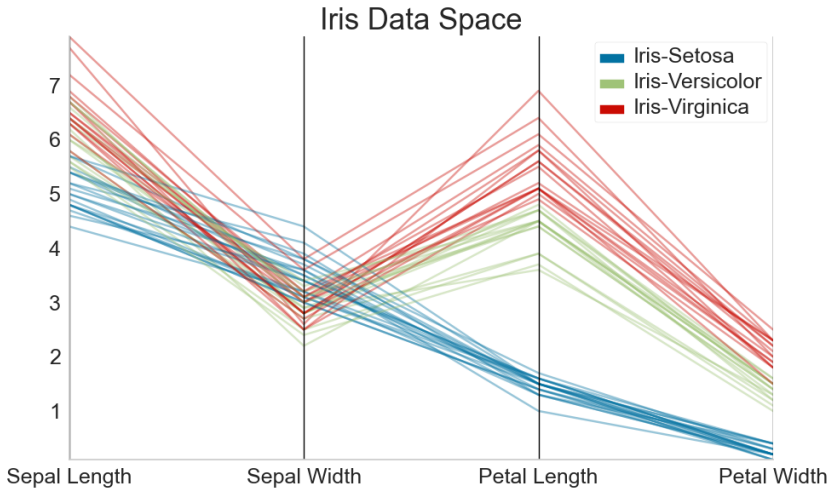
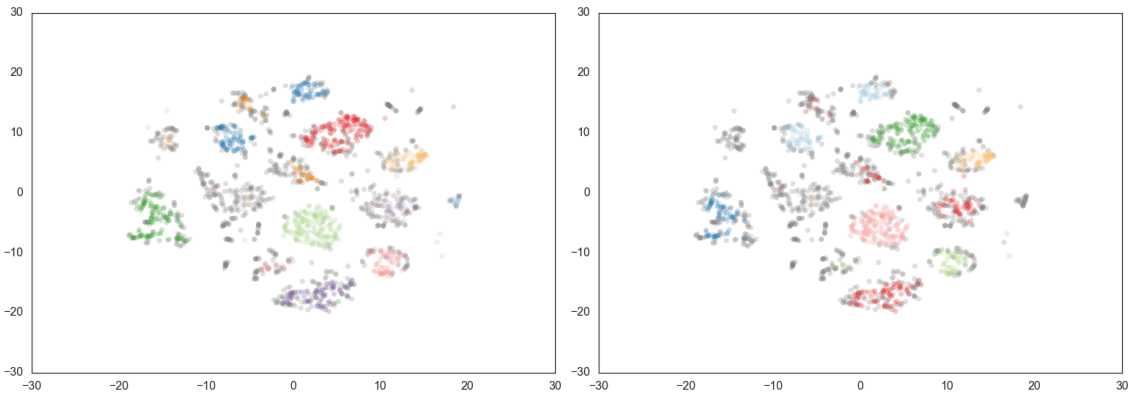


Figure 2.9: PCP over Iris dataset showing how classes separate across the test split of the dataset.

2.8.3 t-distributed stochastic neighbour embedding (t-SNE)

The final visualisation is the t-distributed stochastic neighbour embedding plot, also known as t-SNE. The t-SNE plot allows projection of a high-dimensional dataset onto a 2-dimensional representation. The core idea is to explore the data and obtain an intuition for the arrangement of the high-dimensional data. Additionally, the t-SNE can help gaining insights into the number of clusters chosen, often used to aid the choice of number of clusters and general exploration. For this thesis, the t-SNE was used to supplement the choice of hyperparameters for the HDBSCAN algorithm, explored in section 2.3.

Below in Figure 2.10 are two t-SNE plots for a HDBSCAN clustering for the Iris dataset. Figure 2.10 **(a)** shows how the data is clustered using the hyperparameters of *min_cluster_size* = 15 and Figure 2.10 **(b)** shows the same dataset but with different hyperparameters of *min_cluster_size* = 30 By plotting different hyperparameters it becomes apparent how the clusterings change, in particular in regards to the amount of clusters and noise points. More about this concept in section 2.3.



(a) HDBSCAN clustering: *min_cluster_size* = 15 (b) HDBSCAN clustering: *min_cluster_size* = 30

Figure 2.10: t-SNE example from HDBSCAN documentation showing the projection of clusterings with different hyperparameters [McInnes et al., 2017].

Chapter 3

Exploration Preparation

The upcoming sections in this chapter, first provide an overview of classification datasets and an introduction to relevant aspects of the exploration performed. An important note is that the scope of the thesis is limited to the exploration of classification datasets. Then, preprocessing of the datasets proceeds, and lastly, elaboration about the data and explanation space.

3.1 Overview of Datasets

This section introduces the different classification datasets utilised throughout the thesis. The reasoning for each choice of dataset is provided under each dataset, together with a recap at the end to regain an overview.

Skin Segmentation Dataset

The Skin Segmentation [Dua and Graff, 2017] dataset is a binary classification dataset which classifies whether a RGB color is a skin texture or not (skin:1 and no skin:2). The dataset consists of 2 classes and 245.057 observations with 3 features (RGB) as seen in Table 3.1. The reason for selecting this dataset is due to it's large amount of observations and because it's 3 dimensions, meaning it is plottable.

Index	B	G	R	y
0	74	85	123	1
1	73	84	122	1
⋮	⋮	⋮	⋮	⋮
245052	163	162	112	2
245053	163	162	112	2

Table 3.1: Snippet of Skin Segmentation dataset including index values.

Banknote Authentication Dataset

The Banknote Authentication dataset [Dua and Graff, 2017] is a binary classification dataset consisting of 1372 observations and four features. The dataset was formed based on images from real and forged banknotes. The information extracted from the banknote images resulted in the four features seen below in Table 3.2. Selecting this dataset is primarily due to the low dimensions and binary classes. The y values correspond to 0 for *Real* and 1 for *Forged* banknotes.

Index	Variance	Skewness	Curtosis	Entropy	y
0	3.62160	8.6661	-2.8073	-0.44699	0
1	4.54590	8.1674	-2.4586	-1.46210	0
⋮	⋮	⋮	...	⋮	⋮
1367	0.40614	1.34920	-1.4501	-0.55949	1
1368	-1.38870	-4.87730	6.4774	0.34179	1

Table 3.2: Snippet of Banknote Authentication dataset including index values.

Iris Dataset

The Iris dataset [Dua and Graff, 2017] is a multi-classification dataset consisting of 3 classes of 50 observations each and four features. The dataset classifies the type of specie based on the four features seen below in Tabel 3.3. This particular dataset is chosen because of the multi-class aspect, enabling exploration and experimentation on a dataset with classes > 2.

Index	Sepal Length	Sepal Width	Petal Length	Petal Width	y
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
⋮	⋮	⋮	⋮	⋮	⋮
148	6.0	2.2	4.0	1.0	Iris-versicolor
149	5.9	3.0	5.1	1.8	Iris-virginica

Table 3.3: Snippet of Breast Cancer dataset including index values.

Breast Cancer Dataset

The Breast Cancer dataset [Dua and Graff, 2017] is a binary classification dataset consisting of 569 observations and 30 features. The value of each feature comes from an image of the cell nuclei from a procedure called fine needle aspirate (FNA) of a breast mass. The y values are Benign(B) and Malignant(M), which numerically represent 1 for Benign and 0 for Malignant in the dataset.

The choice for this dataset is due to the high number of features, which enables explorations and experimentation on a dataset where features are > 10. The relevance of the dataset is also prominent as it is from a high-risk domain; healthcare.

Index	Radius Mean	Texture Mean	...	Symmetry Worst	Concavity Worst	y
0	17.99	10.38	...	0.7119	0.4601	1
1	20.57	17.77	...	0.2416	0.2750	1
⋮	⋮	⋮	...	⋮	⋮	⋮
567	20.60	29.33	...	0.9387	0.4087	1
568	7.76	24.54	...	0.0000	0.2871	0

Table 3.4: Snippet of Breast Cancer dataset showing 4/30 features and labels.

Recap of datasets

Below in Table 3.5 is an overview summarising the datasets.

Dataset	$\|\boldsymbol{D}\|$	**Features**	**Labels**
Skin Segmentation	245.057	3	Skin, No Skin
Banknote Authentication	1372	4	Real, Forged
Iris	150	4	Setosa, Versicolor & Virginica
Breast Cancer	569	30	Benign, Malignant

Table 3.5: Overview and recap of all datasets described in this section.

3.2 Preprocessing of the Datasets

Applying Machine Learning Algorithms

The Random Forest classifier is used to establish a black box model for the datasets. The aim of establishing a model for each dataset is to create a black box and not to create the best possible model for each dataset. It is, however, desirable that the accuracy is somewhat high, as the predictions for the dataset are the labels used for the exploration to mimic a realistic scenario.

Preprocessing Procedure

- Split data into X (data space) and Y (Labels)
- Apply standardisation on X split of the dataset
- Split data into X_train, X_test, Y_train, Y_test
- Train a Random Forest model using X_train and Y_train
- Evaluate model using X_test, Y_test
- Perform grid search for K-means and HDBSCAN to find the best hyperparameters
- Find SHAP values using KernelExplainer

3.3 Defining Data Spaces

This section elaborates on the previously introduced mathematical notation in Table 2.1 in Section 2.1.1, namely the Data Space and Explanation space.

- The **data space**, denoted $\boldsymbol{D}_S$, is the space in which the dataset's features exist.
- The **explanation space**, denoted $\boldsymbol{D}_E$, is the space where the SHAP values/explanations ϕ_{ij} of each $x_i \in \boldsymbol{D}$ the dataset exist.

Both spaces, when referred to, do not include any labels, clustering results or predictions. The two terms solely refer to the dataset's features and corresponding explanations. Exemplification follows for the two types of spaces based on the Banknote Authentication Dataset. Table 3.6 is the data space, and Table 3.7 is the explanation space for the dataset Banknote Authentication.

Banknote Authentication Data Space			
Variance	Skewness	Curtosis	Entropy
3.62160	8.6661	-2.8073	-0.44699
4.54590	8.1674	-2.4586	-1.46210
⋮	⋮	...	⋮
0.40614	1.34920	-1.4501	-0.55949
-1.38870	-4.87730	6.4774	0.34179

Table 3.6: Snippet of Banknote Authentication $\boldsymbol{D}_S$ consisting of original values.

Banknote Authentication Explanation Space			
Variance	Skewness	Curtosis	Entropy
0.231660	0.213273	0.003006	0.001365
4.54590	8.1674	-2.4586	-1.46210
⋮	⋮	...	⋮
0.365271	-0.022278	0.083548	0.020240
0.240185	0.253050	-0.060016	0.013562

Table 3.7: Snippet of Banknote Authentication $\boldsymbol{D}_E$ consisting of SHAP values.

Chapter 4

Exploring Regional Explanations

Regional Explanations is a concept that lies between local and global explanations, as discussed in the previous section 2.6. The upcoming chapter will explore a fraction of the potential topics within regional explanations. Other suggestions for exploration within regional explanations are provided in future work in the upcoming section 7, with a description of the more concepts untouched within this thesis.

This chapter starts with visualising the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ using a PCP. Then a definition of regions, followed by an exploration of similarity measures for regions. A section about clustering follows where two algorithms, K-means and HDBSCAN, are analysed. Next, the chapter evaluates and compares the findings and exemplifies how regional explanations are extractable using SHAP—lastly, a small section concerning a concept for identifying wrong predictions.

4.1 Visualising the Data and Explanation Space

When visualising datasets of ≤ 3 dimensions, it is possible to visually inspect the data and detect patterns by, for example, using a scatter plot. As dimensions become > 3, plotting the data on a scatter a plot is no longer a viable option. However, alternative visualisation techniques exist that enable the visualisation of a high-dimensional dataset; such techniques include the Parallel Coordinates Plot (PCP), aforementioned in subsection 2.8.2.

The PCP for investigating regional explanations provides an insightful visualisation to view the particular $\boldsymbol{D}_S$ individually or compare the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$. In particular, with classification datasets, it is possible to see how features create natural groupings based on their values and corresponding class assignment.

Two upcoming examples show how the PCP is usable when visually examining and comparing the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$. The first example uses a binary classification dataset, while the second example uses a multi-class classification dataset.

Visualising a Binary Classification Dataset

A binary classification dataset is the first and most forthright type of classification to visualise using a PCP. The dataset Banknote Authentication, introduced in section 3.1, is used for this example. In order to examine and compare the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ visually, two figures are visible in Figures 4.1 and 4.2.

The first example in Figure 4.1 visualises the $\boldsymbol{D}_S$ representing the data in its original established format and structure. The blue horizontal lines correspond to observations for class *Real (0)*, while the green lines correspond to class *Fraud (1)*. When inspecting the PCP, it is clear that many observations have overlapping features, yet there is still a distinction between the feature and class values. By removing the colouring of classes from the PCP, one may detect two classes due to the contrasting values, particularly

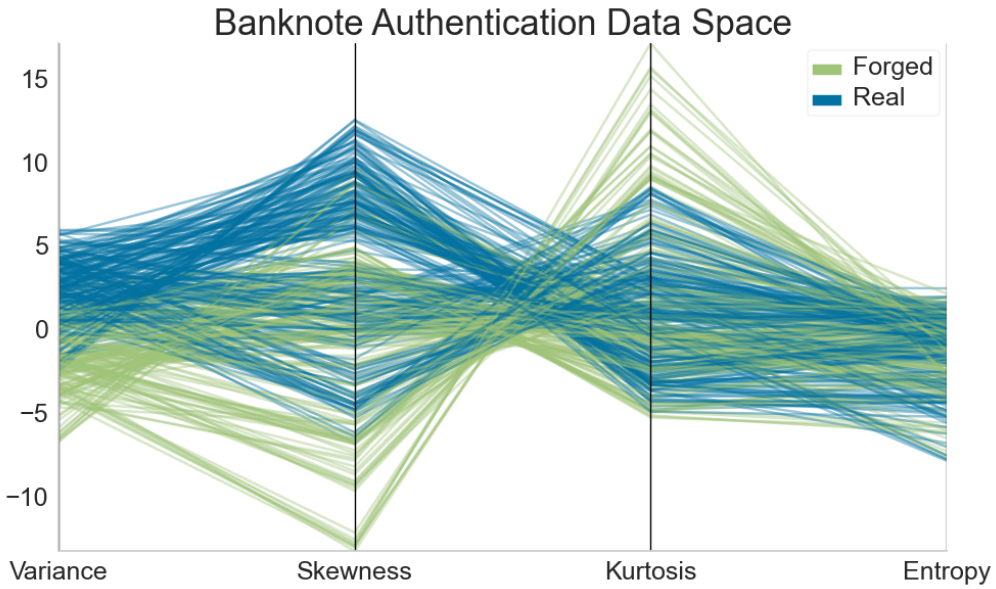
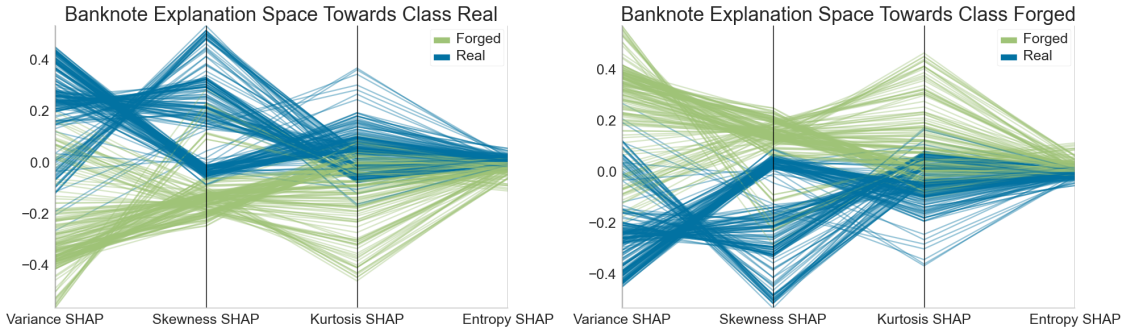


Figure 4.1: Visualisation of Banknote Authentication $\boldsymbol{D}_S$ on a PCP.



(a) $\boldsymbol{D}_E$ towards class *Real*.

(b) $\boldsymbol{D}_E$ towards class *Forged*.

Figure 4.2: PCP of Banknote Authentication for the $\boldsymbol{D}_E$, **(a)** shows explanations towards class *Real*, and **(b)** shows explanations towards class *Forged*.

for features Variance and Skewness; however, this distinction is less significant given the particular dataset.
When looking at the PCP of the $\boldsymbol{D}_S$, the following is deducible:

- *Real* images of Banknotes tend to have higher Entropy and Variance values, a mixture of negative and medium Skewness values and negative to low values of Kurtosis.
- *Forged* images of Banknotes tend to have low and negative values of Entropy and Kurtosis, many very negative values of Variance, and many medium to high values of Skewness.

The second example, in Figure 4.2, visualises the Banknote Authentication $\boldsymbol{D}_E$, where the vertical axis values are equivalent to the local SHAP values of each observation. Figure 4.2 **(a)** shows each feature's marginal contribution towards a prediction for class *Real*, while b) shows the opposite, namely each feature's marginal contribution towards a prediction for class *Forged*. Thus, based on the theory presented in section 2.7.3, the PCP visually acknowledges the calculations for SHAP values towards a particular prediction.

Moreover, when looking at Figures 4.2 **(a)** and **(b)**, there is a clear reflection on the x-axes. For instance, given class Real in **(a)**, the SHAP values are positive towards that class, while on **(b)**, the reflection is evident. Thus, it can be stated that positive SHAP values towards one prediction are negatively reflected towards another prediction by an equal amount.

To sum up the findings of visualising a binary classification dataset, it is evident that Figure 4.1 generally shows the structure and class distribution. Whereas in Figure 4.2, plotting the explanations provides different insights regarding the predictions. Generally, distinguishing between classes on the PCP depends highly on the chosen dataset. A more distinct class separation is seen in the $\boldsymbol{D}_S$ in the following example.

Visualising a Multi-Class Classification Dataset

The procedure for visualising this dataset is similar to the previous example; however, it uses the Iris multi-class classification dataset, introduced in section 3.1.

Figure 4.3 visualises the $\boldsymbol{D}_S$ in its original established format and structure. The blue horizontal lines correspond to class *Setosa*, the red lines correspond to class *Virginica*, and the green lines correspond to class *Veriscolor*. There is no significant separation between the two classes, *Virginica* and *Veriscolor*. If one removes colours from the PCP, these two classes would wrongly be identifiable as a single class. The similarity between classes is an important consideration to bear in mind when clustering the $\boldsymbol{D}_S$.
When looking at the PCP of the $\boldsymbol{D}_S$, the following is deductible:

- Species belonging to the *Setosa* species have a low Petal width and length while medium to large Sepal length and width.
- Species belonging to the *Veriscolor* species have a high Petal length and Sepal width, whereas both the Petal width and Sepal length have medium values.
- Species belonging to the *Virginica* species have a high Petal length, low Petal width, medium Sepal length and a large Sepal Width.

The second example in Figure 4.4 visualises the all three classes in the Iris $\boldsymbol{D}_E$. Figure 4.4 **(a)** shows SHAP values towards class *Setosa*, **(b)** towards class *Veriscolor* and lastly, **(c)** towards class Virginica.

When plotting the $\boldsymbol{D}_E$ for multi-class classification datasets, there is an intrinsic issue. When classes are > 2, the SHAP values calculated are no longer simplistically reflected. Instead, when looking at Figure 4.4 **(a)**, the SHAP values are calculated towards a prediction for class *Setosa*. Thus, all observations with other predictions than class *Setosa* will have low and negative SHAP values, which visually are seen on the lowest parts of the scale. Henceforth, distinguishing between classes on a PCP is only possible for the class in focus. Comparing one class to all other classes, is a commonly occurring in multi-class datasets, it is a One-Versus-Rest (OVR) situation.

Regardless, the PCP still enables insightful findings for multi-class situations, such as the variance of explanations depending on the class in focus. For example, in Figure 4.4 **(a)**, it is clear that the explanations which push against *Setosa* are uninformed. There is not much variance in the values, creating darker lines which indicate overlapping SHAP values. On the contrary, on **(b)** and **(c)**, there is a more significant variance and less stable explanations compared to **(a)**. This trend is linkable to the trends in Figure 4.3, where observations from class *Veriscolor* and *Virginica* are more similar versus class *Setosa*.

In summation, utilising the PCP for binary and multi-class classification problems provides much insight. Due to the class-wise SHAP value calculations, specific issues arise for multi-class classification datasets. Moreover, this issue goes deeper when clustering and finding the similarity in the $\boldsymbol{D}_E$, upcoming in section 4.3.

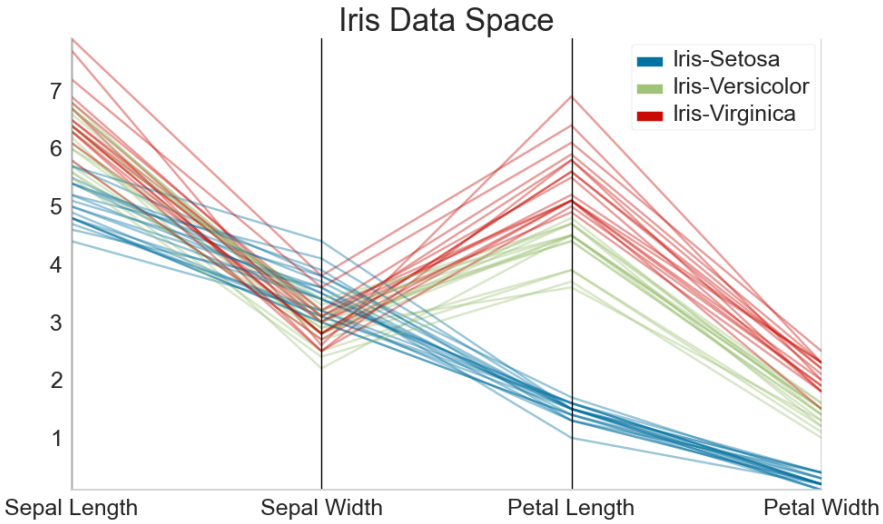
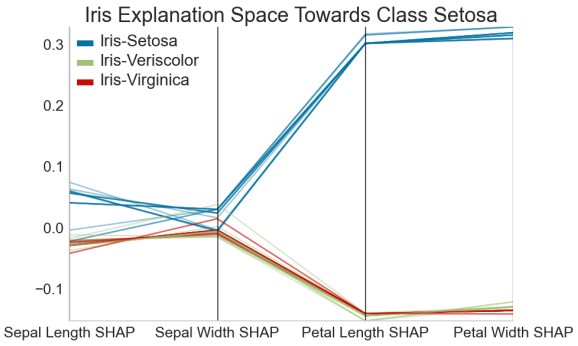
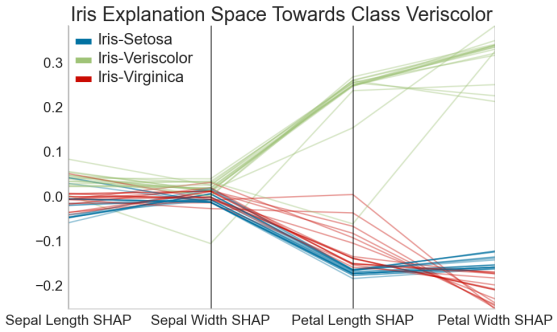


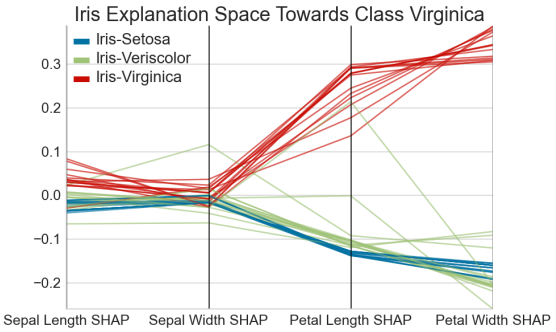
Figure 4.3: PCP of all classes for the Iris $\boldsymbol{D}_S$ showing the separation and similarity between the three classes *Setosa*, *Veriscolor* and *Virginica*.



(a) $\boldsymbol{D}_E$ towards class *Setosa*



(b) $\boldsymbol{D}_E$ towards class *Veriscolor*



(c) $\boldsymbol{D}_E$ towards class *Virginica*

Figure 4.4: PCP of all classes for the $\boldsymbol{D}_E$, **(a)** shows explanations towards *Setosa*, **(b)** shows explanations towards *Veriscolor* and **(c)** shows explanations towards *Virginica*.

4.2 Concept of Regions

While global explanations allow for interpreting the whole model, the concept for regional explanations is to interpret smaller groups of observations. By providing a set of observations (a region) and obtaining SHAP values for that set, it is possible to establish an interpretation of the average feature importance for this region. Establishing regions and explanations for a region means shifting toward more specific explanations instead of global ones, which are more generalised. This section aims to define the overall concept of a region. Furthermore, the definition should provide insights into the approaches explored in the upcoming sections.

Figure 4.5 shows a hypothetical example of a $\boldsymbol{D}_S$ split into two and eight regions. The creation of regions prompts the question: what should the number of regions for a dataset be? There is no straightforward answer to this question as the number of regions depends on various factors, such as the data structure and what the problem at hand requires.

The upcoming exploration investigates different region types and methods that may be feasible when considering formal methods for creating stable regions given a dataset. Firstly, exploration of an approach using only similarity occurs for binary and multi-class classification problems. Then, as introduced in section 2.3 the two clustering methods, K-means and HDBSCAN, are explored. The overall reasoning for choosing these two clustering methods is how they differ. The key difference to keep in mind is how K-means can detect spherical shapes; in contrast, HDBSCAN can utilise both density and hierarchical clustering, recognising many different shapes. Comparing the two methods may be particularly interesting to see how well a simple clustering algorithm such as K-means does against a more complex method like HDBSCAN.

One last mention regarding the comparison is how HDBSCAN can detect outliers/anomalies within the data. If HDBSCAN labels an observation as an outlier, it may not be similar enough to other observations to grant itself a region assignment. In this case, the most logical solution is to accept that this observation does not belong to a region given the data and existing algorithmic hyperparameters. Albeit the observation falls outside of a region, computing a local explanation is a valid option to fall back on.

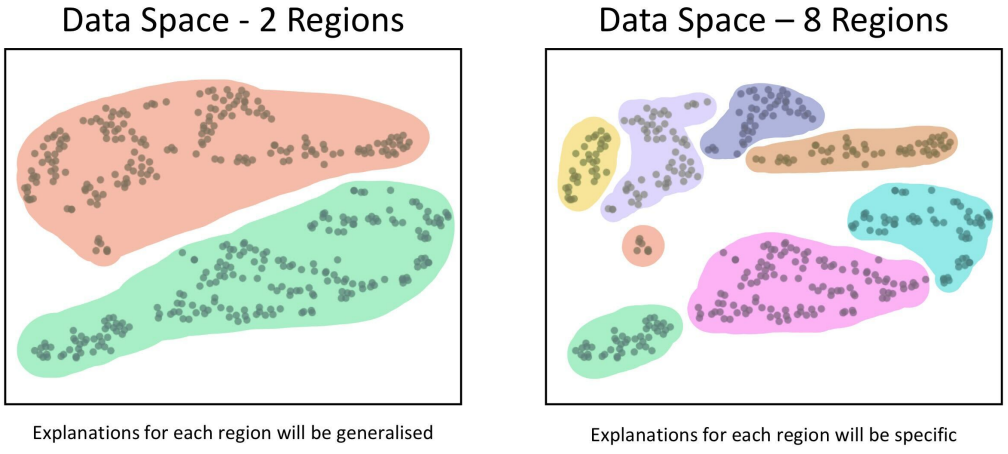


Figure 4.5: The illustration depicts the concept of creating regions. Creating two regions results in more generalised explanations than creating more regions, resulting in more specific explanations per region.

4.3 Similarity Measures for Regions

Distance measures are one way to detect similar observations. For this exploration, the squared Euclidean distance is the chosen distance metric. A distance matrix is computable using the metric containing the distance between all observations. This section will first investigate the similarity between the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$. Additionally, visualisation techniques allow a comparison of the two spaces.

Similarity Within the $\boldsymbol{D}_S$ for Binary Classification

The same binary dataset from Section 4.1, Banknote Authentication, is used as an example of exploring similarity in the $\boldsymbol{D}_S$. When computing the distance between observations, one common preprocessing step is to apply standardisation of the data [James et al., 2013, p.183]. Often, observations in the $\boldsymbol{D}_S$ differ regarding range and units. A common consequence occurs when large or broad ranges of values place an unwanted effect on the distance between observations.

When standardising the observations the mean is subtracted and the data is divided by the standard deviation, resulting in each observation having a mean of zero and a standard deviation of one. After this standardisation, all the features will be on a more comparable scale. Table 4.1 displays the distance matrix over the pairwise squared Euclidean distances for the first five observations. The diagonal is the distance to the point itself, and the matrix is symmetric.

One use of similarity here is to select one observation and find similar points, establishing a type of region for that observation. Table 4.2 shows the distance to the top 10 most similar points to observation 42 , excluding the distance from the observation itself, as the distance is 0.

	0	1	2	3	4	...
0	0.0	0.997	0.783	2.064	1.036	...
1	0.997	0.0	0.449	2.011	1.1	...
2	0.783	0.449	0.0	1.795	1.26	...
3	2.064	2.011	1.795	0.0	2.686	...
4	1.036	1.1	1.26	2.686	0.0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.1: Snippet of distance matrix showing the Euclidean distances between the 5 first observations post standardisation for test split of the dataset Banknote Authentication.

Point	55	79	266	93	11	77	38	156	196	327
Dist.	0.048	0.053	0.062	0.088	0.091	0.125	0.135	0.137	0.181	0.185

Table 4.2: Squared Euclidean distances from observation **42** to top 10 closest points in the $\boldsymbol{D}_S$.

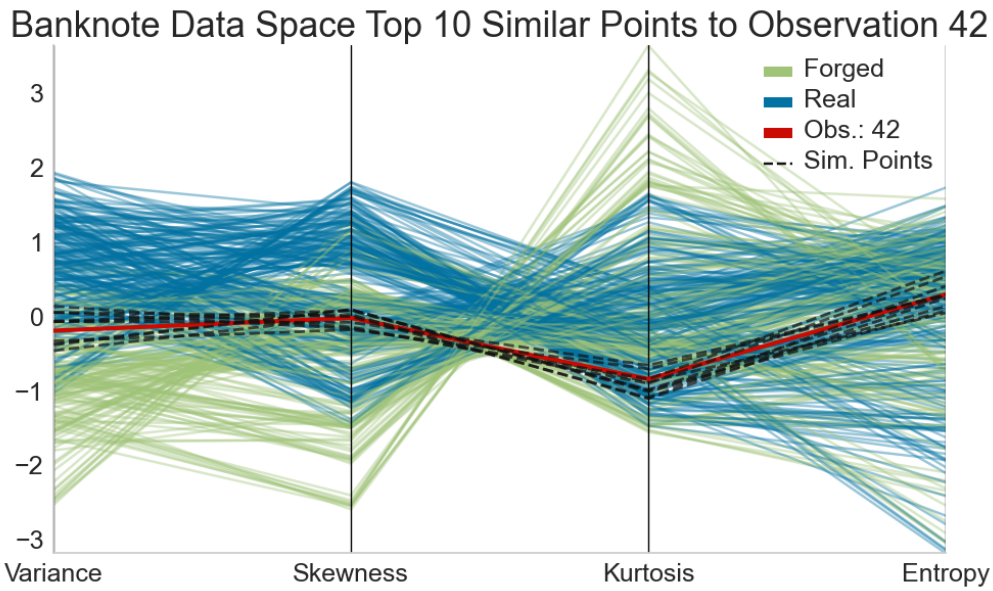


Figure 4.6: PCP of Banknote Authentication $\boldsymbol{D}_S$ with standardised values. The red dotted line is the observation 42 which is in focus, while the black dotted lines are the top 10 most similar observations to point 42.

Figure 4.6 shows this concept by highlighting the observation and its most similar point on the PCP. Together with 42 itself, these ten observations are highlighted in Figure 4.6 to demonstrate their similarity. The red line highlights the point in focus, while similar observations are highlighted by the black dotted lines. It is troublesome to distinguish between the black and red lines, as naturally, they are similar points and, thus, overlap.

Similarity Within the $\boldsymbol{D}_E$ for Binary Classification

Finding similar observations in the $\boldsymbol{D}_E$ is analogous to the $\boldsymbol{D}_S$, however, by using each observation's explanation values. The SHAP values for classification are within a given range [-1:1]; thus, scaling the explanations is unnecessary. Accordingly, as all the explanations follow the same scale, it is straightforward to calculate the pairwise distance between each explanation. Once again, as seen in Table 4.3, the pairwise distances for the first five observations is calculated.

	0	1	2	3	4	...
0	0.0	0.377	0.298	0.385	0.126	...
1	0.377	0.0	0.104	0.133	0.312	...
2	0.298	0.104	0.0	0.137	0.256	...
3	0.385	0.133	0.137	0.0	0.362	...
4	0.126	0.312	0.256	0.362	0.0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.3: Snippet of the distance matrix showing the Euclidean distances between the 5 first observation's SHAP values for the test split of the Banknote Authentication dataset.

Point	55	79	93	207	236	264	339	148	30	41
Dist.	0.001	0.002	0.003	0.003	0.004	0.004	0.004	0.004	0.005	0.006

Table 4.4: Squared Euclidean distances from observation **42** to top 10 similar points in the $\boldsymbol{D}_E$ using SHAP values towards class Forged.

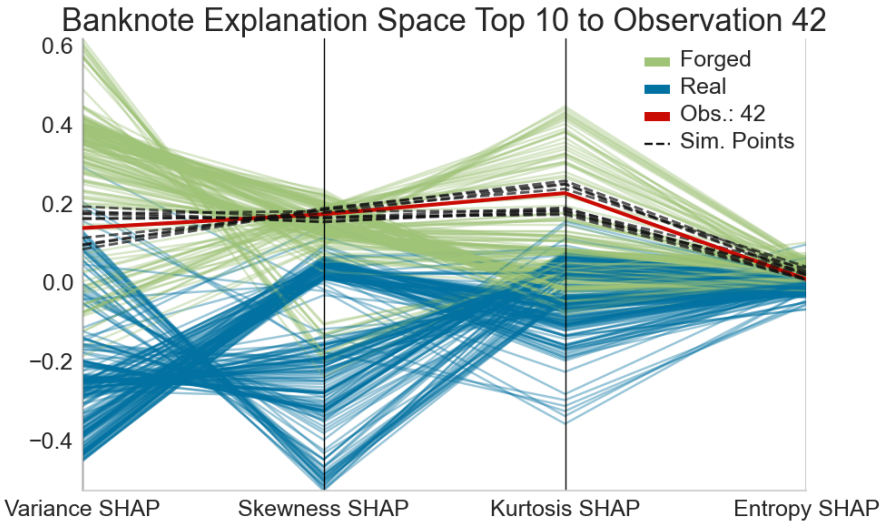


Figure 4.7: PCP of Banknote Authentication $\boldsymbol{D}_E$. The red dotted line is the observation 42 which is in focus, while the black dotted lines are the top 10 most similar observations to point 42.

An important note here is that the SHAP values utilised for calculating distances are towards a prediction for class *Forged (1)*. Because the SHAP values for binary classification are reflected, taking SHAP values towards class Real or Forged does not make a difference. However, later in multi-class scenarios, it will matter when calculating the similarity.

Likewise, as with the $\boldsymbol{D}_S$, an observation is selectable, and similar observations are identifiable. Table 4.4 shows the distances to observation 42 in the $\boldsymbol{D}_E$, while Figure 4.7 shows a PCP of observation 42 and its top ten similar observations.

Comparing Similarity Between $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ for Binary Classification

Now examples from the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ have been provided; the next step is to compare the two spaces. There are many exciting parts to look at when making the comparison, one of which is looking at the shared similar observations between the spaces. Elaborately, this means taking the similar observations from one point and space and comparing them to the same observation and its similar points from the other space. The interestingness of this comparison is to recognise the shared similar points and the extent to which they are similar and explore any other information from the comparison.

One may assume that the similarity is likely to be the same between the spaces. If this assumption holds true, then using the $\boldsymbol{D}_S$ or $\boldsymbol{D}_E$ for similarity purposes would be equally valuable. If, however, it does not hold, then the next step is to investigate to what extent it holds and why the similar shared observations are not the same.

Firstly, investigating the shared similar observations for different thresholds of similar observations, is seen in Table 4.5. The table shows a small selection of six observations providing the amount of shared similar observations per threshold. Although this is a small

sample size out of the Bank Authentications' test set, it provides sufficient information on how the shared similar observations differ from each observation. Already here, it is evident that the assumption that the shared similar observations are equal between the two spaces does not hold.

Top 5, 10, 20, 30 Shared Similar Observations

Index	Top 5	Top 10	Top 20	Top 30
0	3/5	6/10	7/20	8/30
42	3/5	3/10	11/20	16/30
50	1/5	5/10	18/20	25/30
100	0/5	1/10	9/20	11/30
200	1/5	5/10	12/20	21/30
300	4/5	6/10	12/20	21/30

Table 4.5: Six different observations showing the amount of shared similar observations for each threshold 5, 10, 20, 30 for the Banknote Authentication test split, excluding the observation itself.

Threshold	Fraction	Percentage
Top 5	1.83/5	36.6%
Top 10	4.41/10	44.2%
Top 20	9.93/20	49.7%
Top 30	16.29/30	54.3%

Table 4.6: Average Global shared similar observations for the Banknote Authentication test split for each threshold 5, 10, 20, 30

Deciding on a particular threshold is dependent on the dataset; one method would be taking all the observations in the test set and generating an average of shared similar observations for each threshold. The outcomes enable global insight into how the shared similar observations differ. Table 4.6 shows the global similar shared observations for the Banknote Authentication test split, with a length of 412. Dilution of the test split occurs as the threshold increases from 5 to 30, complementing the results found in Table 4.6—the percentage of shared similar observations increases as the threshold increases.

The most significant percentage gap is between the top 5 and top 10; prompting the choice of the top 10 moving forward. With 42 as the observation in focus, similar observations to each space are evident in Table 4.7 **(a)** and **b)**. Moreover, Table 4.7 **(c)** shows the shared similar observations between the two spaces for observation 42.

The shared similar observations are 3/10, and so the next step is to investigate the non-shared similar observations. The non-shared similar points can be taken from the $\boldsymbol{D}_S$ and plotted on the $\boldsymbol{D}_E$ to see which points are not similar more clearly and inspect why. In Figure 4.8, the PCP shows an added yellow line of similar observations found in the $\boldsymbol{D}_S$ but not in the $\boldsymbol{D}_E$. The reason for plotting the non-similar $\boldsymbol{D}_S$ observations onto the PCP of the $\boldsymbol{D}_E$ is that it intuitively makes the most sense, seeing as the SHAP values are

obtained from the $\boldsymbol{D}_S$. The figure shows that the observations from the $\boldsymbol{D}_S$ have different SHAP values once mapped to the $\boldsymbol{D}_E$ and, therefore, are not similar to 42 in the $\boldsymbol{D}_E$.

As discussed in section 2.7.3, computing the SHAP values generates a weight. This weight is not directly transferable from the $\boldsymbol{D}_S$ to the $\boldsymbol{D}_E$. Therefore, the information gained from the $\boldsymbol{D}_S$ is not equal to the information gained from the explanation space. This synergy is explored further in the next section regarding clustering and exploration of mutual information between the two spaces.

The takeaway from the comparison is that there is much information to investigate regarding using the shared similar observation as opposed to the similar observations from the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$, respectively. Later, when exploring the clustering of two spaces, the similarity investigated in this section can be used and compared. The following subsection will follow the same investigation but for multi-class classification to provide insights into the differences that occur similarity for binary vs multi-class classification.

(a) $\boldsymbol{D}_S$ Top 10

Observation	Distance
55	0.048
79	0.053
266	0.062
93	0.088
11	0.091
77	0.125
38	0.135
156	0.137
196	0.181
327	0.185

(b) $\boldsymbol{D}_E$ Top 10

Observation	Distance
55	0.001
79	0.002
93	0.003
207	0.003
236	0.004
264	0.004
339	0.004
148	0.004
30	0.005
41	0.006

(c) Shared Similar Observations

Observation
55
79
93

Table 4.7: Table **(a)** shows top 10 distances to 42 in the $\boldsymbol{D}_S$, while **(b)** shows the same for the $\boldsymbol{D}_E$ and lastly, **(c)** shows the shared similar observations between the two, also highlighted in green in table **(a)** and **(b)**.

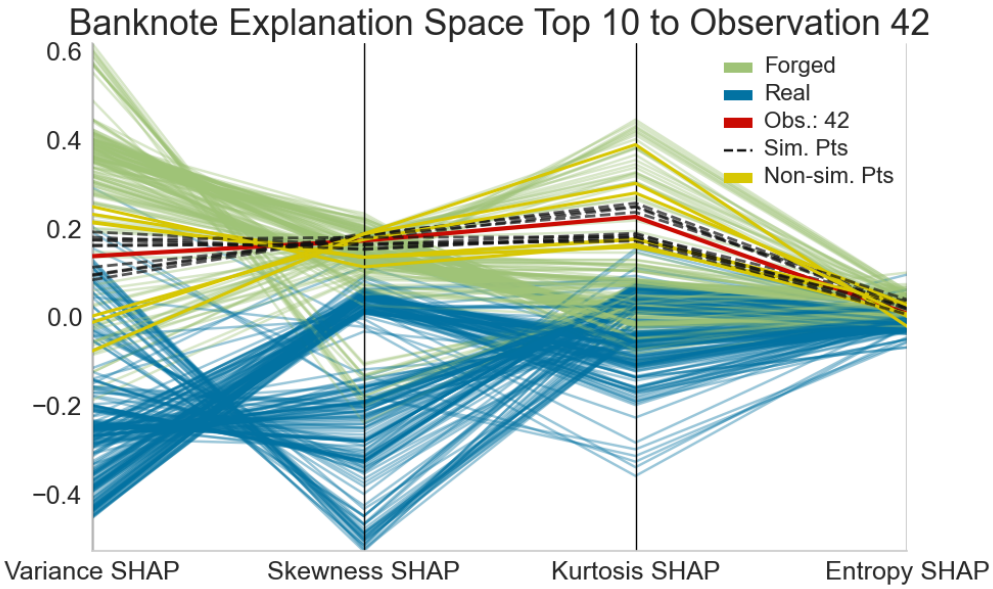


Figure 4.8: PCP of Banknote Authentication $\boldsymbol{D}_E$ for observation 42. The yellow lines show the non-similar observation from the $\boldsymbol{D}_S$ while the black dotted lines still show the top 10 similar observations to 42 in the $\boldsymbol{D}_E$.

Similarity Within the $\boldsymbol{D}_S$ for Multi-class Classification

The same multi-class dataset from subsection 4.1, Iris, is used to explore similarity in the $\boldsymbol{D}_S$. Again, as with the binary dataset, the features are standardised before calculating the squared Euclidean distance. Subsequently, Table 4.8 shows the distance matrix over the pairwise squared Euclidean for the first five observations. Choosing observation 42 again as the focus point, Table 4.9 shows the distance from 42 to the nearest ten observations.

	0	1	2	3	4	...
0	0.0	9.9	7.629	0.237	0.792	...
1	9.9	0.0	29.241	9.494	12.33	...
2	7.629	29.241	0.0	7.694	4.23	...
3	0.237	9.494	7.694	0.0	1.04	...
4	0.792	12.33	4.23	1.04	0.0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.8: Matrix showing pairwise distances between the five first observation post standardisation for test split of the dataset Iris.

Point	32	48	36	15	47	4	10	21	46	3
Dist.	0.046	0.115	0.199	0.466	0.495	0.517	0.596	0.751	0.769	0.947

Table 4.9: Squared Euclidean distances from observation 42 to top 10 similar points in the $\boldsymbol{D}_S$.

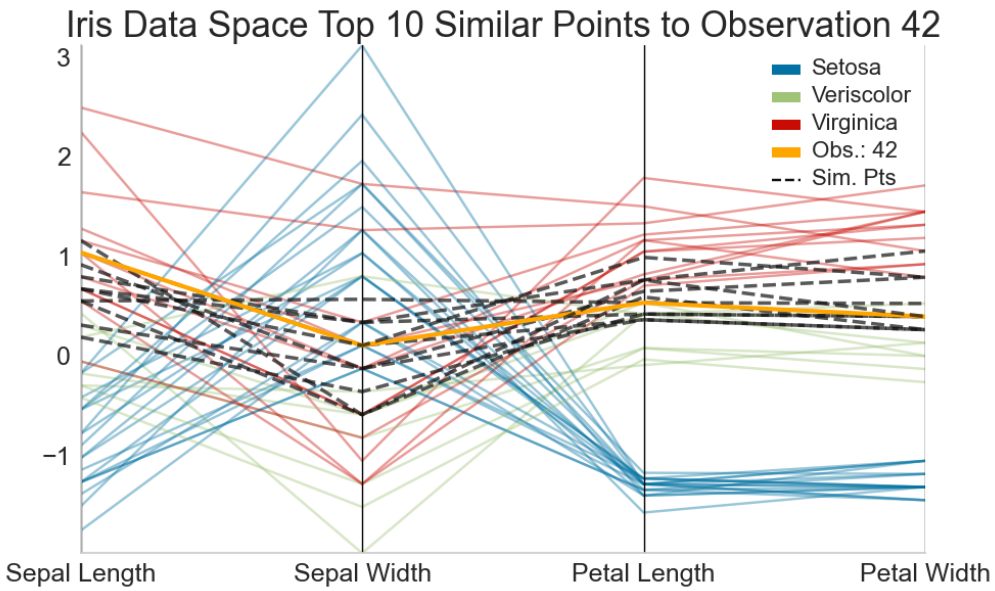


Figure 4.9: Visualisation of Iris $\boldsymbol{D}_S$ with standardised values on a PCP showing. The orange line is observation 42 which is in focus, while the black dotted lines are the top 10 most similar observations to point 42.

Moreover, Figure 4.9 shows a PCP of observation 42 and its similar ten observations. The red line is point 42, while the black dotted lines are the ten similar observations. The process is equal to the binary classification example; however, the Iris dataset has two close classes, meaning that similar points for the two classes may overlap more than the Banknote Authentication dataset. When glancing at Figure 4.9, it is evident that patterns are more challenging to recognise than the binary classification example. When standardising the features for the Iris dataset, the scaling changes make the distinguishment of classes more challenging than the PCP in Figure 4.4. To some degree, after standardisation, *Virginica* and *Veriscolor* (red and green lines) are still visually close. As observation 42 belongs to class *Veriscolor*, it is evident that it has similar observations from class *Virginica*, simply due to the closeness of the two classes.

Similarity Within the $\boldsymbol{D}_E$ for Multi-class Classification

The exciting and differing aspects of the multi-class example lie within the similarity for the $\boldsymbol{D}_E$ because, as aforementioned, the SHAP values are not reflected and create a One Versus Rest situation.

For Iris's $\boldsymbol{D}_E$, there are distinct differences to the binary example. The difference occurs as the Iris dataset is a multi-class classification dataset, meaning that the reflection of SHAP values does not occur. Iris has three sets of SHAP values, each towards a different class. Therefore a different approach must be taken when investigating the similarity between observations. Instead of simply using one set of SHAP values, all three sets play a role depending on the class to which the observation in focus belongs.

An issue arises when deciding which set of SHAP values are best suited for determining similarity. Much future work is available within the aspect of multi-class classification for explanations; thus, for now, a naive approach proceeds. The naive approach assumes that an observation is likely to be more similar to observations with the same class assignment. Therefore, when deciding the set of SHAP values to use for analysing similarity for observation 42, it is the SHAP values towards class *Veriscolor*.

The One Versus Rest situation is apparent from the three Table 4.10, 4.11 and 4.12,

with the distance between each observation; and previously demonstrated in Figure 4.4. The tables include labels to demonstrate that when looking at Table 4.10, where the SHAP values for class Setosa are used, there is a considerable distance difference from Setosa to the other classes. Row 1, column 0 shows the distance from a Setosa observation to the Veriscolor. Row 1, column 0 shows the distance from Virginica to Veriscolor, 0, meaning these two observations are identical. While the distances support the One Versus Rest class situation, the next step is to consider how to deal with this situation.

	0 (Veris).	1 (Seto.)	2 (Virg.)	3 (Veri.)	4 (Veri.)	
0 (Veris.)	0.0	0.422	0.0	0.0	0.0	...
1 (Seto.)	0.422	0.0	0.424	0.422	0.422	...
2 (Virg.)	0.0	0.424	0.0	0.0	0.0	...
3 (Veri.)	0.0	0.422	0.0	0.0	0.0	...
4 (Veri.)	0.0	0.422	0.0	0.0	0.0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.10: Pairwise distances in the $\boldsymbol{D}_E$ with SHAP values towards class *Setosa*

	0 (Veris).	1 (Seto.)	2 (Virg.)	3 (Veri.)	4 (Veri.)	
0 (Veris).	0.0	0.443	0.425	0.001	0.01	...
1 (Seto.)	0.443	0.0	0.008	0.417	0.405	...
2 (Virg.)	0.425	0.008	0.0	0.4	0.398	...
3 (Veri.)	0.001	0.417	0.4	0.0	0.015	...
4 (Veri.)	0.01	0.405	0.398	0.015	0.0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.11: Pairwise distances in the $\boldsymbol{D}_E$ with SHAP values towards class *Veriscolor*

	0 (Veris).	1 (Seto.)	2 (Virg.)	3 (Veri.)	4 (Veri.)	
0 (Veris.)	0.0	0.004	0.428	0.001	0.012	...
1 (Seto.)	0.004	0.0	0.404	0.001	0.028	...
2 (Virg.)	0.428	0.404	0.0	0.402	0.406	...
3 (Veri.)	0.001	0.001	0.402	0.0	0.018	...
4 (Veri.)	0.012	0.028	0.406	0.018	0.0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.12: Pairwise distances in the $\boldsymbol{D}_E$ with SHAP values towards class *Virginica*

Point	32	48	36	0	6	3	18	45	9	17
Dist.	0.0	0.0	0.0	0.0	0.001	0.001	0.001	0.001	0.001	0.001

Table 4.13: Pairwise distances from observation *42* to top 10 similar points in the $\boldsymbol{D}_E$.

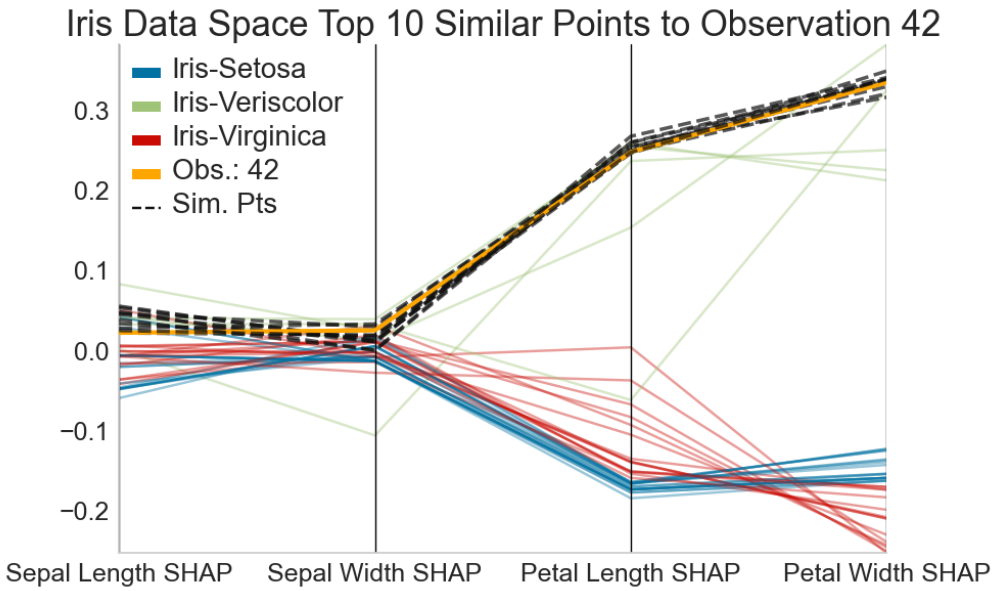


Figure 4.10: Iris $\boldsymbol{D}_E$ on a PCP showing the top 10 most similar observations to 42. The orange line is observation 42 which is in focus, while the black dotted lines are the top 10 most similar observations to point 42. As 42 is from class Veriscolor, the SHAP values used towards Veriscolor are used.

When investigating observation 42 in the $\boldsymbol{D}_E$, the distance from the point given the SHAP values array for class *Veriscolor* is evident in Table 4.13. Moreover, the PCP for observation 42 in the $\boldsymbol{D}_E$ granted the SHAP values for class *Veriscolor* is visible in Figure 4.10. Point 42 lies close to the observations from class *Virginica* when looking at the PCP. Naturally, selecting the correct SHAP values depending on the observation in focus prompts much extra work. There is difficulty when finding similarities, but later, when clustering, it also becomes apparent that multi-class datasets in this situation bring many difficulties. Table 4.13 also shows that the distances are minimal because the SHAP values are towards the same class.

Similarity Between $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ for Multi-class Classification

Similarily to the binary classification example, the similarity between the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ is done by looking at the shared similar observations for the Iris dataset. First, because the Iris is significantly smaller than the Banknote Authentication, the test set is also smaller, meaning that a new threshold set is used instead. Investigation of six observations from the Iris test set and their shared similar points between the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ are seen below in Table 4.14. The new thresholds are top 5, 10, 15 and 20. Moreover, the similarity depends on the class due to the multi-class aspect. When including the class label, a new column is added to display the class of each observation.

Next, the global shared similar observations are found, in Table 4.15 to gain insight into how the shared similar observations look across the whole test split. In the same

manner as the binary dataset, the biggest gap is between top 5 and top 10, meaning top 10 seems to be a reasonable choice moving forward.

Top 5, 10, 15, 20 Shared Similar Observations

Index	Class	Top 5	Top 10	Top 15	Top 20
0	Veriscolor	2/5	6/10	11/15	14/20
10	Veriscolor	2/5	7/10	12/15	18/20
20	Virginica	2/5	7/10	12/15	18/20
30	Setosa	4/5	6/10	14/15	19/20
40	Setosa	1/5	5/10	11/15	16/20
42	Veriscolor	3/5	4/10	9/15	10/20

Table 4.14: Six different observations which shows how many are shared similar observations out of the top 5, 10, 15, 20 for the Iris test split, excluding the observation itself.

Threshold	Fraction	Percentage
Top 5	1.56/5	31.2%
Top 10	4.92/10	49.2%
Top 15	8.28/15	55.2%
Top 30	11.38/20	56.9%

Table 4.15: Average Global shared similar observations for the Iris test split for each threshold 5, 10, 15, 20

4.4 Exploration of Clustering

This section explores clustering of the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$. The previous section used similarity as a means to potentially to identify a region surrounding an observation. Alternatively clustering proves another way of exploring the creation of regions. This section investigates firstly clustering and creation of regions within the $\boldsymbol{D}_S$. Next, the same is done for the $\boldsymbol{D}_E$. Moreover, evaluation and comparison of the clusterings is performed together with alternative use cases.

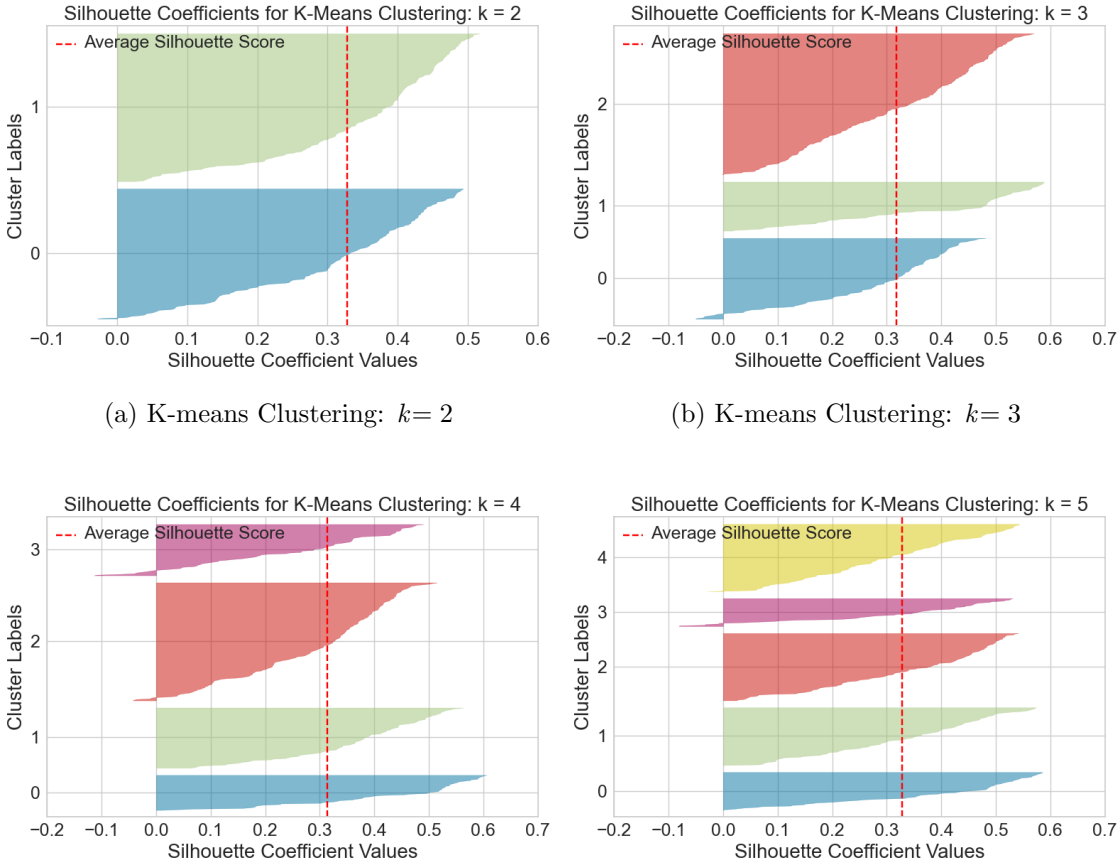
A key note moving forward is that only binary classification is explored further. Previous section gave some insights into difficulties that come with multi-class classification, and time constraints did not allow for further exploration.

4.4.1 Clustering the Data Space

The Banknote Authentication dataset is used for this section as an example. First K-means is used to cluster the standardised test split. The silhouette coefficients for $2 \leq k \leq 9$ are in Table 4.16, where there is only a small difference between the silhouette coefficients. Recall from section 2.3 the silhouette method. Figure 4.11 shows the fit of K-means using $k = 2, 3, 4, 5$.

k	2	3	4	5	6	7	8	9
Silhouette Scores	0.328	0.317	0.314	0.328	0.316	0.312	0.313	0.307

Table 4.16: Silhouette Coefficients per k for Banknote Authentication $\boldsymbol{D}_S$



(a) K-means Clustering: k= 2

(b) K-means Clustering: k= 3

(c) K-means Clustering: k= 4

(d) K-means Clustering: k= 5

Figure 4.11: Comparison of Silhouette Coefficients for K-means as k increases.

Based on the different K-means clusterings, the decision for a k value cannot be decided purely from the Silhouette Scores. Therefore, by visually inspecting the density and separation of the clusters, $k = 2$ seems like an optimal choice. It is also evident that for all clusterings there are some negative silhouette coefficients, meaning that observations may have been assigned to a wrong clusters. Choosing $k = 2$, a contingency table seen in Table 4.17 is established.

Following the contingency table, the purity of the clustering $\mathcal{C}$ and clusters $\{C_0, C_1\} \in \mathcal{C}$ is calculated and shown in Table 4.18. Recall section 2.5.1, where a high purity signals the extent a clustering or cluster has observations from the same class.

The creation of regions will depend on the dataset; for specific datasets, a high k may be appropriate, meaning the data is split into many regions. Given the data structure, a low value is most suitable for other datasets. The central question is how is the number of clusters determined. Are they dependent on the problem at hand? On the naturally best clustering outcome? Or other requirements? If two large clusters are created, do they differ enough from the global explanation? Moving forward, the focus will lie on creating the best possible clusters by validation scores as a means of purpose for region

establishment.

Given that the clustering validation measures point towards a clustering being a good fit or superior to other clusterings, it makes sense to select that clustering as the basis for regions for the given dataset. Given the best clustering, the next step is to explore similarities within the clusters. The exploration of similarity from the previous Section 4.3 is comparable and usable for that.

Onwards, Table 4.19 shows the distances from observation 42 to the top 10 observations based on squared Euclidean Distance to point 42; however, within the assigned K-means cluster C_0.

Table 4.19 shows the top 10 distances once again to point 42, however, within the assigned K-means cluster namely, class C_0. When comparing with the previous Table 4.2 the two approaches share observations: $[55, 79, 266, 93, 11, 77, 38, 156, 196]$ so 7/10 of the nearest observations to 42 within the $\boldsymbol{D}_S$ using the similarity approach are the same closest points to the K-means clustering with $k = 2$ and selecting the similarity from point 42 to its nearest 10 points.

	T_1: Real	T_2: Forged	n_i
C_0	117	112	229
C_1	76	107	183
m_j	193	219	$n = 412$

Table 4.17: Contingency table after K-means clustering given $k = 2$.

	Majority Class Count	$\lvert C_i \rvert$	Purity
C_0	117	193	0.606
C_1	112	219	0.511
	Sums of Majority Classes	$\lvert \mathcal{C} \rvert$	
$\mathcal{C}$	229	412	0.556

Table 4.18: Purity table over the K-means clustering $\mathcal{C}$ and $\{C_0, C_1\} \in \mathcal{C}$

Point	55	79	93	11	77	38	156	187	111	395
Dist.	0.048	0.053	0.088	0.091	0.125	0.135	0.137	0.194	0.216	0.244

Table 4.19: Squared Euclidean distances from observation *42* to top 10 similar points in the K-means region in $\boldsymbol{D}_S$.

Continually, HDBSCAN is applied to the same data set with the same goal in mind of creating meaningful regions. Using the procedure for finding the best hyperparameters as prescribed in Section 2.3 for HDBSCAN, and the visualisation technique described in section 2.8.3, the relative validity for the clustering is 0.311 with $min\_cluster\_size = 5$ and $min\_samples = 7$. Moreover, the clustering consists of 7 clusters and noise, as seen in Figure 4.12. Observation 42 is identified by the black dot in the middle of the blue lower cluster with class 1. For each of the clusters a contingency and purity table is established in Table 4.20 and Table 4.21 respectively.

Following the contingency table, the purity of the clustering $\mathcal{C}$ and clusters $\{C_0, C_1\} \in \mathcal{C}$ is calculated and shown in Table 4.21. Recall section 2.5.1, where a high purity signals the extent a clustering or cluster consists of one class. Next the similar points for observation 42 are identified in Table 4.22.

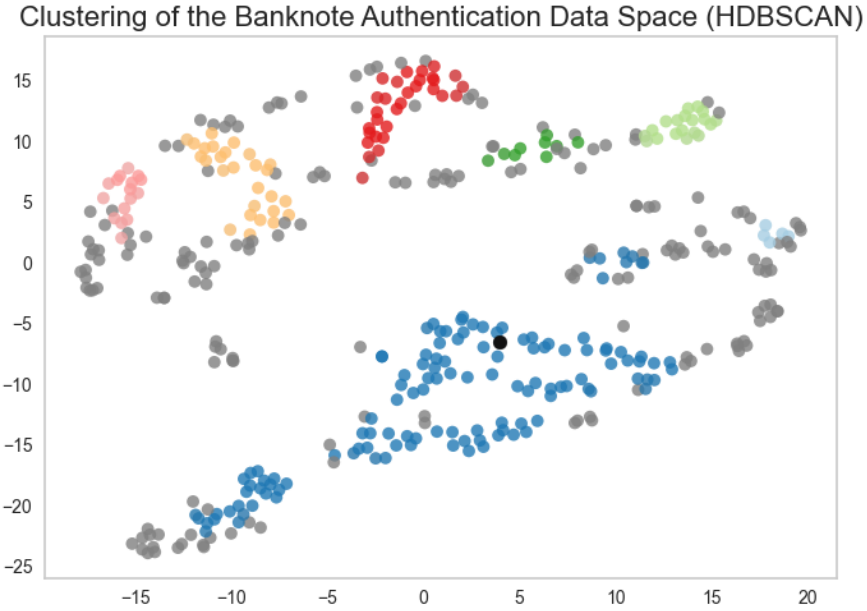


Figure 4.12: t-SNE projection of HDBSCAN to supply insight into noise vs labelled points using the hyperparameters $min\_cluster\_size = 5$ and $min\_samples = 7$, resulting in 7 clusters and noise points. The one black point is the placement of observation within the t-SNE projection.

While it is evident that many points are declared noise, it is arguable that forcing larger clusters at the risk of reducing the quality of the clustering is not the desired outcome. As previously mentioned, if a point is declared noise, then the natural solution for that observation is to find the local explanation as no regional explanation would seemingly be sufficient. To compare the similar shared observations from HBDSCAN clustering to K-means and the basic similarity approach the following was evident. When comparing the top 10 observations from 42 the distances had zero in common with the similar or K-means. When comparing to the whole region in which 42 belongs to, it is evident that the basic similar approach all 10/10 points are within the c_1. For K-means, 8/10 exist within c_1, thus while not in the top 10 to 42, the observations are almost all still within the c_1 cluster.

To sum up the exploration, K-means can create two larger uniform regions, but with much impurity, smaller regions are likely to produce a higher purity than larger regions. HDBSCAN does well at identifying smaller regions but much noise. Noise is not necessarily

a problem, but it does also not contribute towards a region meaning local explanations for noise points must be used. The next subsection similarly investigates the $\boldsymbol{D}_E$.

	T_1: Real	T_2: Forged	n_i
Noise	121	59	180
C_0	5	0	5
C_1	3	123	126
C_2	19	0	19
C_3	9	0	9
C_4	16	0	16
C_5	30	0	30
C_6	27	0	27
m_j	109	123	$n = 412$

Table 4.20: Contingency table after HDBSCAN clustering given *min_cluster_size* $= 5$ and *min_samples* $= 7$

	Majority Class Count	$\lvert C_i \rvert$	Purity
C_0	5	5	1.0
C_1	123	126	0.976
C_2	19	19	1.0
C_3	9	9	1.0
C_4	16	16	1.0
C_5	30	30	1.0
C_6	27	27	1.0
	Sums of Majority Classes	$\lvert \mathcal{C} \rvert \setminus$ noise	
$\mathcal{C}$	$106 + 123 = 229$	232	0.987

Table 4.21: Purity table over the HDBSCAN clustering $\mathcal{C}$ and $\{C_0, \ldots C_5\} \in \mathcal{C}$

Point	55	79	266	93	11	77	38	156	196	327
Dist.	0.048	0.053	0.062	0.088	0.091	0.125	0.135	0.137	0.181	0.185

Table 4.22: HDBSCAN clustering identification of region for observation 42, and the Squared Euclidean distances from observation **42** to top 10 similar points in the $\boldsymbol{D}_S$.

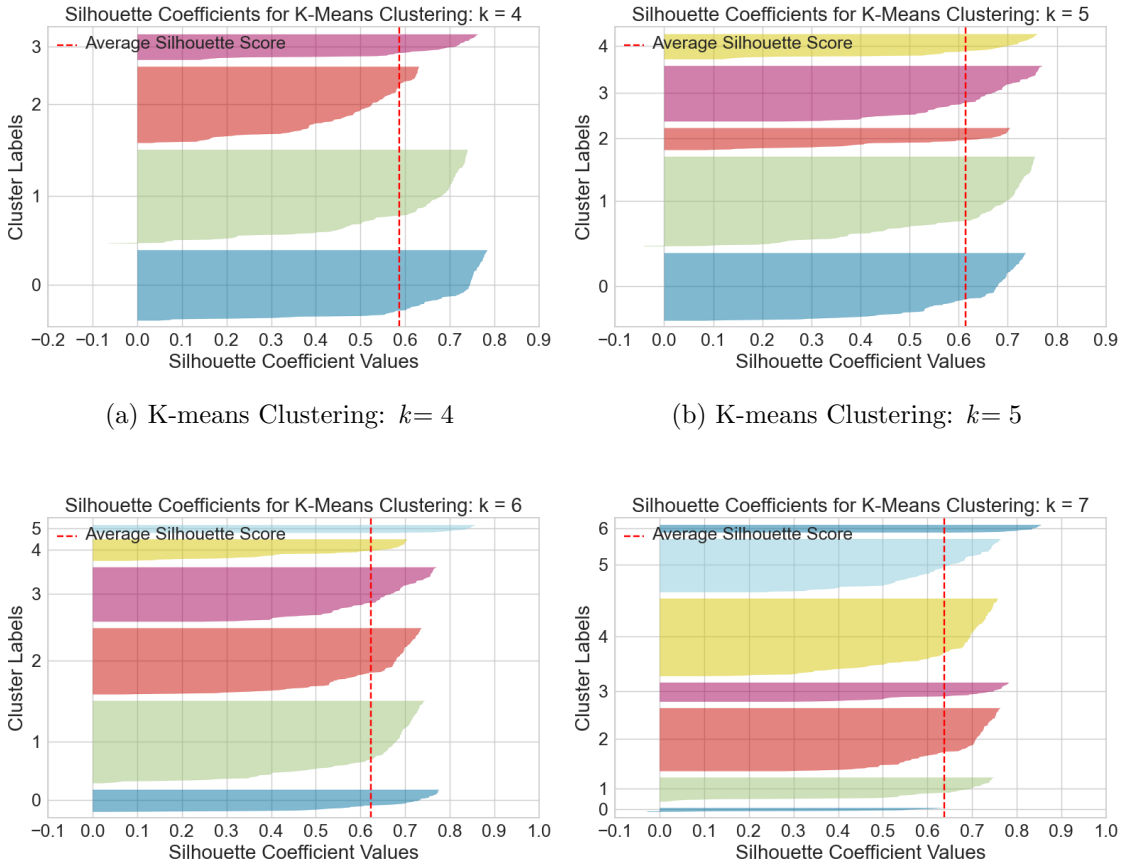
4.4.2 Clustering the Explanation Space

Moving to the $\boldsymbol{D}_E$ a similar analysis is performed on the same dataset. Firstly for K-means, Table 4.23 shows the Silhouette scores where the best scores are $k = 5, 6, 7$, then a comparison of the highest scoring silhouette scores are seen in Figure 4.13.

Based on the silhouette images in Figure 4.13 a $k = 6, 7$ may be an appropriate k value, however, with $k = 5$ the clusters are more uniform and placement of the average silhouette score line is better. Therefore, $k = 5$ is used moving forward and a contingency table is established in Table 4.24 together with a purity table in Table 4.25. Already at this point it is evident that clustering using a higher k in the $\boldsymbol{D}_E$ is possible and purities suggest the same. Following the contingency table, the purity of the clustering $\mathcal{C}$ and clusters $\{C_0, \ldots, C_4\} \in \mathcal{C}$ is calculated. Recall section 2.5.1, where a high purity signals the extent a clustering or cluster consists of one class.

k	2	3	4	5	6	7	8	9
Silhouette Scores	0.564	0.562	0.587	0.614	0.624	0.638	0.596	0.548

Table 4.23: Silhouette Coefficients per k for Banknote Authentication $\boldsymbol{D}_E$



(a) K-means Clustering: k= 4

(b) K-means Clustering: k= 5

(c) K-means Clustering: k= 6

(d) K-means Clustering: k= 7

Figure 4.13: Comparison of Silhouette Coefficients for K-means as k increases.

	T_1: Real	T_2: Forged	n_i
C_0	107	0	107
C_1	1	141	142
C_2	35	0	35
C_3	88	0	88
C_4	0	40	40
m_j	231	181	$n = 412$

Table 4.24: Contingency table after K-means clustering given $k = 5$.

	Majority Class Count	$\lvert C_i \rvert$	Purity
C_0	107	107	1.0
C_1	141	142	0.993
C_2	35	35	1.0
C_3	88	88	1.0
C_4	40	40	1.0
	Sums of Majority Classes	$\lvert \mathcal{C} \rvert$	
$\mathcal{C}$	411	412	0.997

Table 4.25: Purity table over the K-means clustering with $\mathcal{C}$ and $\{C_0, \ldots, C_4\} \in \mathcal{C}$

Looking within cluster C_4 where observation 42 lies, the following top 10 similar points within the region are seen in Table 4.26.

When comparing to the similarity approach within the $\boldsymbol{D}_E$ the top 10 observations differ only by a single point. Next, is a look at how HDBSCAN fairs for the $\boldsymbol{D}_E$. Figure 4.14 shows the t-SNE visualisation of HDBSCAN with the parameters of *min_cluster_size* = 11 and *min_samples* = 4 and a relative validity of 0.456. Given these hyperparameters, HDBSCAN creates 6 clusters and noise points.

Point	55	79	93	236	264	339	77	11	80	395
Dist.	0.001	0.002	0.003	0.004	0.004	0.004	0.01	0.01	0.013	0.014

Table 4.26: Squared Euclidean distances from observation *42* to top 10 similar points for K-means in the $\boldsymbol{D}_E$.

Following the contingency Table 4.27, the purity of the clustering $\mathcal{C}$ and clusters $\{C_0, \ldots, C_5\} \in \mathcal{C}$ is calculated and shown in Table 4.28. Recall section 2.5.1, where a high purity signals the extent to which a clustering or cluster consists of one class. The purity for the clustering (excluding noise points) is 1, meaning each cluster consist only of one class, which is an excellent result. The HDBSCAN clustering in the $\boldsymbol{D}_E$ produces high purity and not a whole lot of noise. The next subsection will compare the different clusterings.

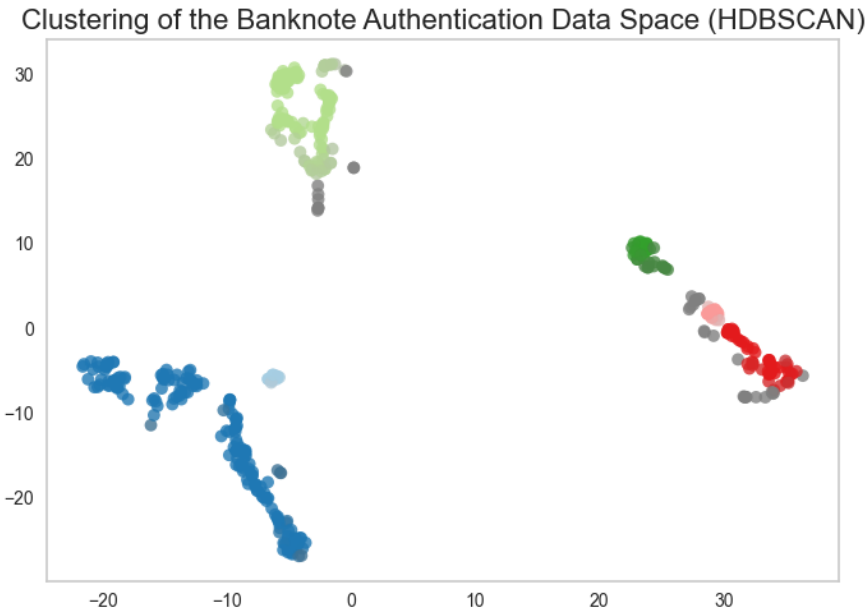


Figure 4.14: HDBSCAN t-SNE visualisation using the hyperparameters *min_cluster_size* = 11 and *min_samples* = 4, resulting in 6 clusters and noise points.

	T_1: Real	T_2: Forged	n_i
Noise	32	1	33
C_0	0	13	13
C_1	0	168	168
C_2	101	0	101
C_3	32	0	32
C_4	15	0	15
C_5	50	0	50
m_j	230	182	$n = 412$

Table 4.27: Contingency table after HDBSCAN clustering in the $\boldsymbol{D}_E$, given *min_cluster_size* = 11 and *min_samples* = 4

	Majority Class Count	$\lvert C_i \rvert$	Purity
C_0	13	13	1.0
C_1	168	168	1.0
C_2	101	101	1.0
C_3	32	32	1.0
C_4	15	15	1.0
C_5	50	50	1.0
	Sums of Majority Classes	$\lvert \mathcal{C} \rvert \setminus$ noise	
$\mathcal{C}$	230	232	1.0

Table 4.28: Purity table over the HDBSCAN clustering $\mathcal{C}$ and $\{C_0, \dots C_5\} \in \mathcal{C}$

4.4.3 Evaluation and Comparison

First Table 4.29 proceeds, showing a summation of the points for each space and method towards observation 42. The similarity approach uses the squared Euclidean distance in the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ for the top 10 observations. For K-means and HDBSCAN the regions are firstly found and then dependent on the region in which observation 42 belongs, the top 10 similar points are found, based off the squared Euclidean distance. There are some definite trends between the approaches. For the $\boldsymbol{D}_S$, the first two observations are identical, while moving down the list many observations are similar but at different positions; meaning at different distances for the respective approach. The $\boldsymbol{D}_E$ has the first three points identical to 42. After the first three points, the list continues but some points differ and/or are staggered differently.

To provide a better way to identify which approaches have the same points in common, Table 4.30 shows a matrix of the shared similar points of each possible combination of spaces and methods. Looking at Table 4.30 it is interesting to see that the similarity no clustering approach within the $\boldsymbol{D}_S$ has all observations in common with the HDBSCAN region's top 10 observation for 42. Furthermore, the similarity with no clustering approach in the $\boldsymbol{D}_E$ also have all observations in common to HDBSCAN in the $\boldsymbol{D}_E$. While this is simply one example with one observation, a general overview is provided later to provide holistic insight to all points.

Comparing Clusterings

Next, the clusterings are compared, so the similarity approach explored in section 4.3 is not used. Using the NMI and ARI validation measures, introduced in 2.3, the comparison of each clustering algorithm in the respective space is visible in Table 4.31.

While these clusterings are evaluated independently, it is possible to look at the mutual information between the clusterings. By doing so, one can gain insight into how much information the clusterings share between space, and identify the two clusters that share the most information, this metric may be used to selecting a good set of clusterings for regional explanations.

Data Space - Top 10 Observations to 42										
Top 10	1	2	3	4	5	6	7	8	9	10
Simimlarity	55	79	266	93	11	77	38	156	196	327
K-means	55	79	93	11	77	38	156	187	111	395
HDBSCAN	55	79	266	93	11	77	38	156	196	327
Explanation Space - Top 10 Observations to 42										
Top 10	1	2	3	4	5	6	7	8	9	10
Simimlarity	55	79	93	207	236	264	339	148	30	41
K-means	55	79	93	236	264	339	77	11	80	395
HDBSCAN	55	79	93	207	236	264	339	148	30	41

Table 4.29: Recap of all three methods and the top 10 observations based on distance from point 42.

	Simi. $\boldsymbol{D}_S$	Simi. $\boldsymbol{D}_E$	K-means $\boldsymbol{D}_S$	K-means $\boldsymbol{D}_E$	HDBSCAN $\boldsymbol{D}_S$	HDBSCAN $\boldsymbol{D}_E$
Simi. $\boldsymbol{D}_S$	10	3	7	5	10	3
Simi. $\boldsymbol{D}_E$	3	10	3	6	3	10
K-means $\boldsymbol{D}_S$	7	3	10	6	7	3
K-means $\boldsymbol{D}_E$	5	6	6	10	5	6
HDBSCAN $\boldsymbol{D}_S$	10	3	7	5	10	4
HDBSCAN $\boldsymbol{D}_E$	3	10	3	6	4	10

Table 4.30: Matrix over the similar observations between the different spaces and approaches.

The two main comparison are:

- K-Means $\boldsymbol{D}_S$ vs. K-Means $\boldsymbol{D}_E$
- HDBSCAN $\boldsymbol{D}_S$ vs. HDBSCAN $\boldsymbol{D}_E$

as it makes most sense to compare same clustering methods and between spaces. However, the following two comparison below are also provided:

- K-Means $\boldsymbol{D}_S$ vs. HDBSCAN $\boldsymbol{D}_E$
- HDBSCAN $\boldsymbol{D}_S$ vs. K-means $\boldsymbol{D}_E$

with the intention of providing alternative combinations, and to concretize that such combinations produce worse outcomes. Table 4.31 shows the NMI scores of the clustering comparisons. Table 4.31 shows NMI scores of the clustering comparisons and furthermore, that the NMI between clustering of the same method between spaces is higher than NMI between clusterings of differing methods. Based on the results in Table 4.31 the best combination of clusterings is: HDBSCAN in $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$.

With that clustering combination questions appear, such as: would using the HDBSCAN $\boldsymbol{D}_E$ clustering be the best for establishing regional explanations? Is it possible to use the information between two clusterings for anything valuable? The first question is rather easy to answer. As HDBSCAN $\boldsymbol{D}_E$ provides the best validation measure, less noise and a good amount of region, it would seem like a obvious choice to use for this particular dataset with the goal of creating many regions. However, what is to be gained from comparing clusterings between spaces? Upcoming section 4.6 comes with a suggestion for this topic.

	NMI	ARI
K-Means $\boldsymbol{D}_S$ vs. K-Means $\boldsymbol{D}_E$	0.373	0.245
HDBSCAN $\boldsymbol{D}_S$ vs. K-Means $\boldsymbol{D}_E$	0.360	0.185
HDBSCAN $\boldsymbol{D}_S$ vs. K-Means $\boldsymbol{D}_S$	0.309	0.177
HDBSCAN $\boldsymbol{D}_S$ vs. HDBSCAN $\boldsymbol{D}_E$	0.466	0.313

Table 4.31: Normalized Mutual Information Score across different combinations of clusterings to detect how much NMI are between clusterings.

4.5 Generating Explanations for Regions Using SHAP

Once the identification of regions is complete, it is possible to generate explanations using the regions as an isolated dataset using the same method as one would for the whole model (global explanations) but on smaller datasets.

The clustering results of Banknote Authentication HDBSCAN $\boldsymbol{D}_E$ clustering are used for establishing regional explanations. The idea is to take a cluster/region and its assigned observations and use the SHAP global explanation method for that region.

Utilising SHAP's feature importance bar plot, the overall feature importance is plotted in Figure 4.15. Then, the region in which observation 42 is isolated and a complete dataset with the observations within that region is created. To compare with the global feature importance in Figure 4.15, a similar plot is created for region 1, the region in which 42 is assigned. Seen in Figure 4.16 is the feature importance for that particular region. A comparison can be made by looking at the regional feature importances vs. the global feature importances. It becomes clear that some value lies within establishing regional explanations as they become more specific to the points in focus, making it possible to pinpoint which features have the most significant impact.

Furthermore, it is interesting to see to what extent a region differs from the global view. To provide another example, a smaller region is chosen. Figure 4.17 shows how feature importance for region 4 which consists of only 15 observations. The comparison shows that dependent on the region, there is significant change in the feature importance per region.

While these visualisations provide some insight, one could imagine more detailed visualisations that allow more significant insights into the features' values within regions. For example, it would be informative to see how one observation is similar or dissimilar to the region by comparing the features of an observation to the region.

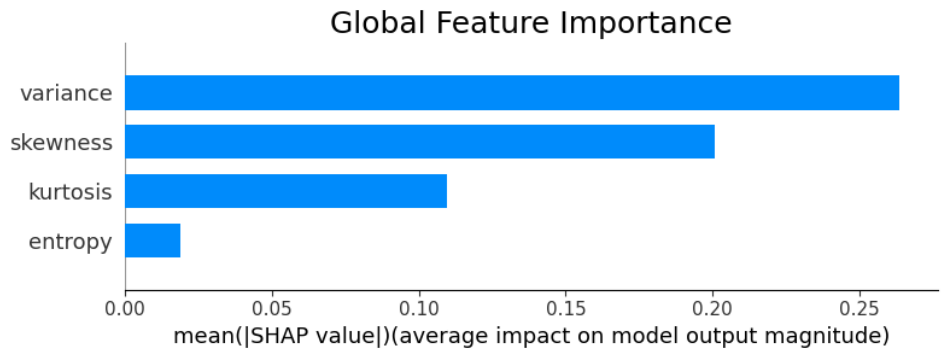


Figure 4.15: Global feature importance.

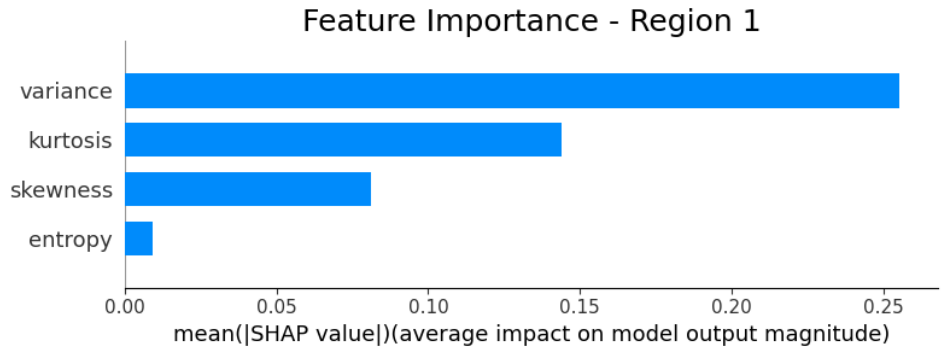


Figure 4.16: Feature importance for region 1, with 168 observations.

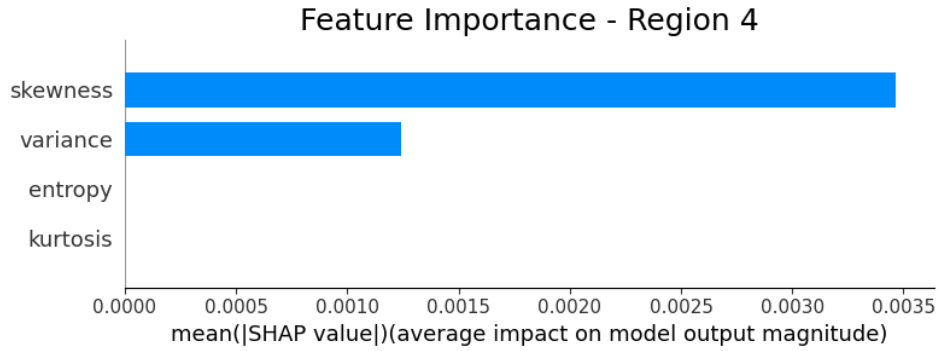


Figure 4.17: Feature importance for region 4, with 15 observations.

4.6 Concept for Identification of Wrong Predictions

A final concept that will be explored is using the similarity approach and clustering of the data to determine "good" or "bad" predictions. The idea behind this concept is that when a new observation is obtained and predicted, the clustering may be able to provide information about the extent to which a prediction may be regarded as a "good" or "bad" prediction. One could imagine a scenario within healthcare where a model has given a prediction, but beyond the prediction, the model provides an indication/threshold for how precise the prediction is. If a patient has a high certainty, then the patient may not need a check up before many months. On the contrary if there is an indication that the patients prediction may be a "bad" prediction, this information is conveyed and provided to the clinicians. The outcome of providing such information to the clinician may be that a the

patients should have a check up sooner. This is simply one scenario, but many other scenarios within different domains may make use of such an indicator for the validity of a prediction. So, how would such a concept become realised?

By using the previous exploration, the purity and similarity tools are can to be used. When adding a new sample to the dataset, it is possible to determine which HDBSCAN cluster the sample will fall within. Once the sample is assigned a cluster, it is possible to investigate the observations predictions in the region it has been placed within. One way may be to look at the region and the predictions within the region and compare the predictions of the observations to the a new observation. Essentially by using the region, the class purity can be used. If a region is entirely pure, meaning all the observations have the same prediction, then if the new sample does not have the same class, questions can be raised to whether this prediction is good prediction.

An alternative perspective on this may be to predict the new sample and within both the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$, find the top 10 similar observations to the new sample, without any clustering involved. After doing so, the top 10 from the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ can be analysed. For example, by taking the top 10 points to the new sample in the $\boldsymbol{D}_S$, the purity of the top 10 can be investigated. If there is a mix of $\hat{y}$ values for those observations, then not much can be said. However, if the new sample has a prediction of 1 and the most similar points all have a prediction of 0, then it would be interesting to dig deeper and see if this new sample has been wrongly predicted. Additionally, as the data has been clustered, the underlying clustering of the top 10 points can be used. By taking each of the top 10 observations, seeing their cluster assignment and the purity of said cluster, are there detectable trends? How many of the top 10 similar points are in the same cluster?

Such questions need elaborate testing and refinement of methods, which may be an interesting concept for future work. To sum up the concept, the overall idea is to investigate whether there is potential to produce an indicator for new samples to gain insight into whether new samples seem valid or have potential to be wrong predictions.

The next Chapter 5 will provide two different use cases where the exploration approaches are used and analysed. Lastly, the chapter will provide a small discussion and comparison of the two cases and methods applied.

Chapter 5

Experiments and Use Cases

This chapter includes use cases conducted with two different datasets than those used in the exploration, namely the Breast Cancer and Skin Segmentation datasets. The decision for these two data sets is to provide two distinct use cases, both a high and a low-risk domain dataset. This chapter's end is a short section that recaps and discusses the result.

5.1 High-Risk Domain Use Case

The high-risk domain case uses the Breast Cancer dataset, introduced in section 3.1. Recall that the dataset's binary labels are Benign/B (1) and Malignant/M (0).

The data in the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ are visible in the PCP; in Figures 5.1 and 5.2. As this dataset has 30 features, the PCP quickly becomes chaotic. Despite being chaotic, overall trends are extractable from the PCP. In the $\boldsymbol{D}_S$, generally, values greater than 0 belong to class Malignant (0), while low and negative feature values typically belong to class Benign (1). There are also clear trends when examining the PCP in the $\boldsymbol{D}_E$. Comparing features, it is evident which contribute more towards a prediction of 1.

After visualising the data in the respective spaces, the next step is to cluster the data using K-means and HDBSCAN to detect how well the data fairs on both algorithms; and which would be the better choice for generating regions. Table 5.1 summarises the results, showing the best scores for each clustering algorithm and the number of clusters each produces. Within the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$, the results show that K-means is superior to HDBSCAN. Based on these results and other previously described ways to evaluate the clusterings, it would make sense to use K-means to create regions, as K-means clusters better in both spaces.

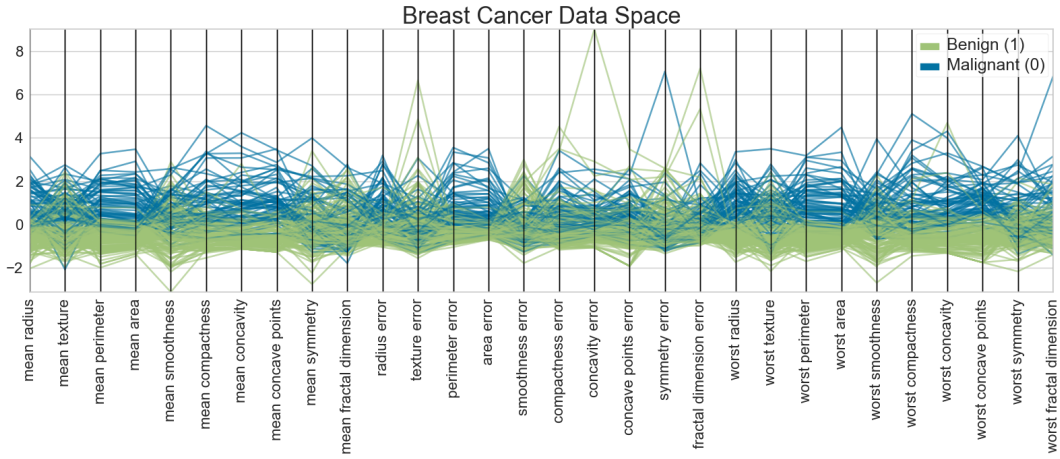


Figure 5.1: PCP over Data Space of Breast Cancer Dataset

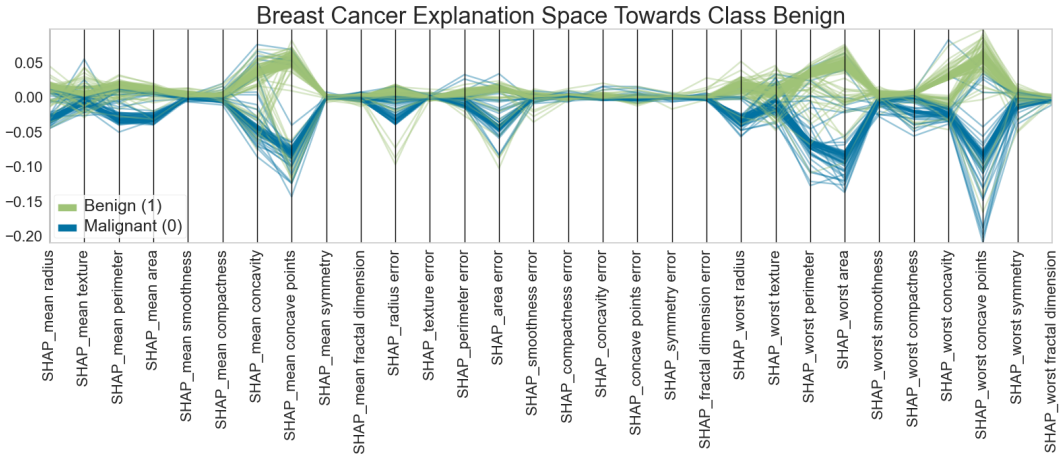


Figure 5.2: PCP over Explanation Space of Breast Cancer Dataset

Data Space - Results			
Algorithm	Validation Score	Hyperparameters	Clusters
K-means	Silhouetter Score: 0.361	$k = 2$	2
HBDSCAN	Relative Validity: 0.075	min_cluster_size = 2	2
		min_samples = 1	
Explanation Space - Results			
Algorithm	Validation Score	Hyperparameters	Clusters
K-means	Silhouette Score: 0.636	$k = 2$	2
HBDSCAN	Relative Validity: 0.308	min_cluster_size = 6	5
		min_samples = 2	

Table 5.1: Best clustering results for K-means and HDBSCAN in $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$.

Tables 5.2 and 5.3 show a contingency table of the results when using K-means to cluster in the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$ space; together with a purity table. These clusters are relatively pure in both spaces. However, one keynote is that observations are not evenly distributed, which is not necessarily a desire if the clusters are pure despite an uneven distribution of observations.

The chosen K-means clustering separates the data into two clusters, which are not many regions. It may be more beneficial for a health dataset to want more specific regions to use. To investigate whether it is possible to divide the clusters into more regions, the established clusters are subset and k-means is run one more time.

The goal is to see whether the data is clusterable into additional meaningful clusters. Figure 5.3 show the result of running K-means on the clusters. Each cluster then splits into two additional clusters using a $k = 2$, which results in four regions. By splitting into four regions instead of keeping the 2, more specific regional explanations are extractable. Moreover, it is interesting to detect if the minority class observations are spread across regions or grouped.

Contingency Table: Data Space			
	T_1: Malignant	T_2: Benign	n_i
C_0	50	3	53
C_1	17	118	135
m_j	67	121	$n = 188$
Contingency Table: Explanation Space			
	T_1: Malignant	T_2: Benign	n_i
C_0	2	122	124
C_1	62	2	64
m_j	64	124	$n = 188$

Table 5.2: Contingency table for K-means clustering $k = 2$ in $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$.

Purity Table: Data Space			
	Majority Class Count	$\lvert C_i \rvert$	Purity
C_0	50	53	0.943
C_1	118	135	0.874
	Sums of Majority Classes	$\lvert \mathcal{C} \rvert$	
$\mathcal{C}$	168	188	0.894
Purity Table: Explanation Space			
	Majority Class Count	$\lvert C_i \rvert$	Purity
C_0	122	124	0.984
C_1	62	64	0.969
	Sums of Majority Classes	$\lvert \mathcal{C} \rvert$	
$\mathcal{C}$	184	188	0.979

Table 5.3: Purity table for K-means clustering $\mathcal{C}$ and $\{C_0, C_1\} \in \mathcal{C}$ for $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$

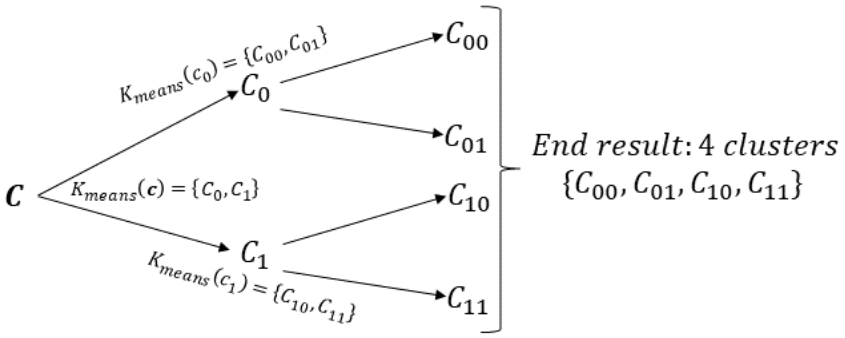


Figure 5.3: Visual representation of creating 4 regions using K-means.

Tables 5.4 and 5.5 show a contingency table these clearly show how the new clusters are separated and how the purity within the clusters look. It is important to note the different sizes of regions. A critical notice is the different sizes of regions. One issue with creating more and more regions is that regions may at some point only consist of 1 observation. With HDBSCAN, the algorithm can detect noise points. Alternatively, noise points can be considered observations that lack enough similarity with other points to establish a region. Naturally, this is one downside to K-means unless the problem requires hard-partitioning of all observations. However, forcing observations to be part of regions does not make much intuitive sense for regional explanations. These observations will "skew" the regional explanations as they are potentially not as similar as other observations.

Given the final four regions, it is possible to use SHAP to calculate the regional feature importance. Figure 5.4 shows the feature importance for the top 20 features in each region. The bar plots show which features have the highest average contribution per region. Creating regions makes it easier to pinpoint which features have the most significant impact from region to region, which can be helpful when receiving a new sample. Comparing an observation to its region can be helpful as the individual feature values are comparable, and the similarity and dissimilarities are detectable, allowing for in-depth analysis of observations.

The last chosen step of analysis for this use case is investigating wrong predictions. Given the random forest model for this dataset, there are seven wrong predictions; 8, 20, 77, 82, 108, 164 and 176. By picking observation 8, $y = 1$ while the predicted $\hat{y} = 0$. The wrong observation belongs to region 4, a relatively small cluster. When looking at region 4, it is evident that observation 108 is also included in region 4, and has the same y and $\hat{y}$ labels. When looking at patterns of wrong predictions, many scenarios can be recognised and dependent on the dataset. For region 4, given observation eight, other observations are also wrongly predicted, so an indicator could be put on the region to communicate that new samples falling within this region may have a higher likelihood of being a wrong prediction.

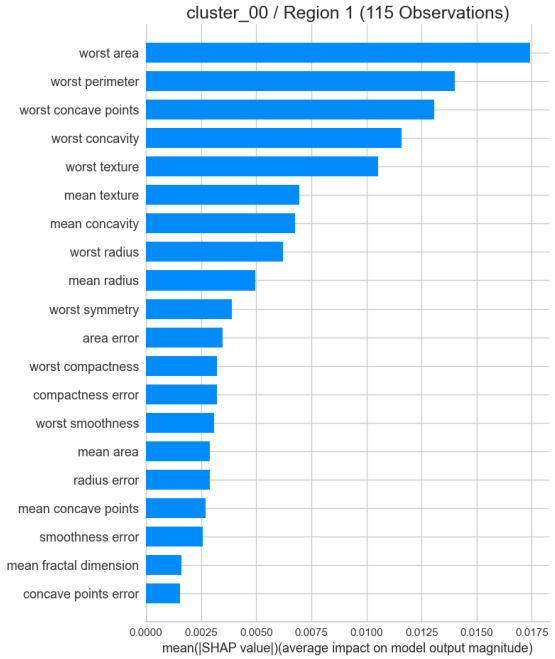
Observation 20 is another wrong prediction which falls within another region, namely region one. The observation has a $y = 0$ and $\hat{y} = 1$, where the majority class is, in fact, one within the region, making it notoriously hard to determine anything "wrong" about this observation. Looking at the rest of the observations, it is evident that the last wrong predictions 77, 82, 108, 164, and 176 all fall within this region; which makes sense since the region is relatively large, consisting of 114 observations. However, it is also noticeable how similar wrong predictions fall within the same regions. Generally identifying distinctness about wrong predictions is complex but a fascinating topic to investigate in potential future work.

Contingency Table: Data Space			
	T_1: Malignant	T_2: Benign	n_i
C_{00}	30	0	30
C_{01}	20	3	23
C_{10}	0	57	57
C_{11}	14	64	78
m_j	64	124	$n = 188$
Contingency Table: Explanation Space			
	T_0: Malignant	T_1: Benign	n_i
C_{00}	2	112	114
C_{01}	0	10	10
C_{10}	55	1	56
C_{11}	7	1	8
m_j	64	124	$n = 188$

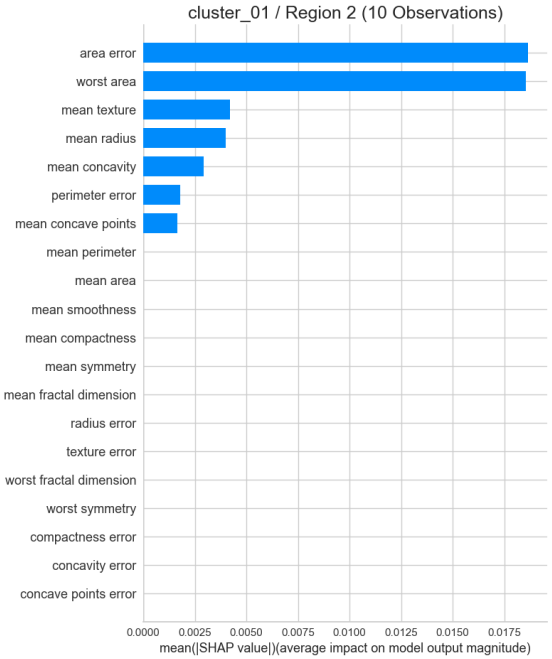
Table 5.4: Combined contingency table for the four regions in for $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$

Purity Table: Data Space			
	Majority Class Count	$\lvert C_i \rvert$	Purity
C_{00}	30	30	1.0
C_{01}	20	23	0.87
C_{10}	57	57	1.0
C_{11}	64	78	0.821
	Sum of Majority Classes	$\lvert \mathcal{C} \rvert$	
$\mathcal{C}$	171	188	0.910
Purity Table: Explanation Space			
	Majority Class Count	$\lvert C_i \rvert$	Purity
C_{00}	112	114	1.0
C_{01}	10	10	1.0
C_{10}	55	56	0.982
C_{11}	7	8	0.875
	Sum of Majority Classes	$\lvert \mathcal{C} \rvert$	
$\mathcal{C}$	184	188	0.979

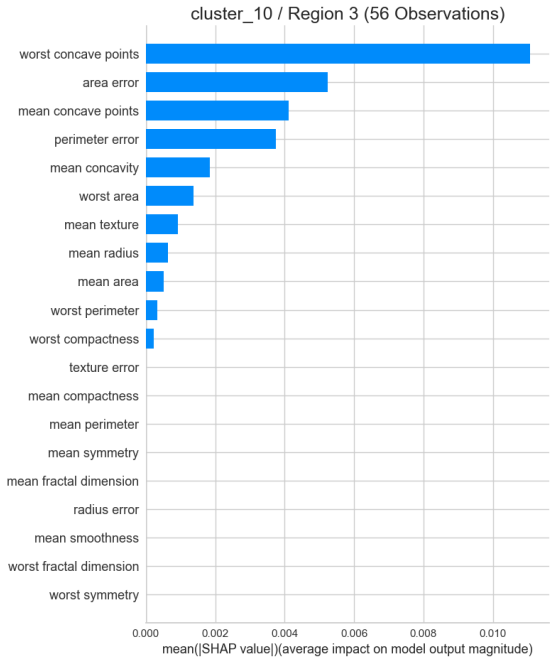
Table 5.5: Purity table for K-means clustering: $\{C_{00}, C_{01}, C_{10}, C_{11}\} \in \mathcal{C}$ in $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$



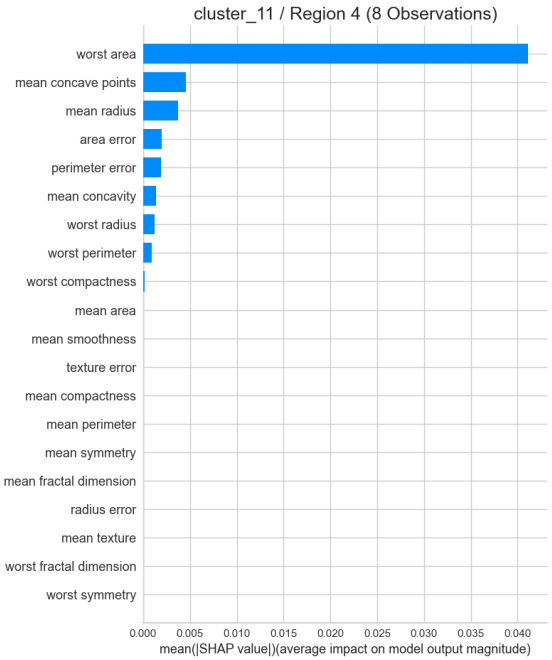
(a) Feature importance for C_{00}/Region 1



(b) Feature importance for C_{01}/Region 2



(c) Feature importance for C_{10}/Region 3



(d) Feature importance for C_{11}/Region 4

Figure 5.4: Feature Importance for top 20 in clusters: $\{C_{00}, C_{01}, C_{10}, C_{11} \in \mathcal{C}\}$ in $\boldsymbol{D}_E$

5.2 Low-Risk Domain Use Case

The Skin Segmentation dataset with class Skin (1) and No Skin (2), introduced in section 3.1, is now investigated. This dataset has 245.057 observations, which makes it an exciting case to analyse when looking at creating regions. Figures 5.5 and 5.6 show a PCP of the $\boldsymbol{D}_S$ and $\boldsymbol{D}_E$. Recall the PCP for the Breast Cancer dataset where the 30 features made the PCP chaotic to look at. For the Skin Segmentation dataset, there are a large number of observations, making it hard to distinguish between observations and detect patterns on the PCP. The first Figure 5.5 is tough to determine anything from; even when standardising the data, no help is gained. Because the range of the features is from $[0 : 255]$, there are many overlapping observations and combinations. The only detectable pattern is that class Skin, the blue lines, are more central within the data. The PCP for the explanation space in Figure5.2 shows a more apparent separation but still many overlapping values around 0.

When applying HDBSCAN and K-means on such large datasets, data sampling to determine hyperparameters is a must as otherwise the time complexity is very high when performed on a lot of data. This dataset is notoriously hard to cluster for many reasons. Using HDBSCAN with the best parameters from the grid search results in a 11320 clusters; a very large number of clusters. For K-meas a partition of two provides the best silhouette score. This dataset is a much different case to the Breast Cancer dataset as there is a lot of observations and few patterns to detect.

Additionally, as the purpose of the dataset is to distinguish between the Skin and No Skin classes, what would be the ideal knowledge that one would want to know? For this dataset, the wrong predictions would most likely be the most interesting factor to gain information about. It would be plausible to cluster via HDBSCAN and extract the wrong predictions out and then compare with the region in which they are in, unless they are declared noise points. For such a low-risk domain dataset, setting a single-pixel wrong within an image may not be as severe as wrongly predicting breast cancer. Generating regional explanations would still be beneficial in analysing where the model takes a wrong turn and comparing close samples.

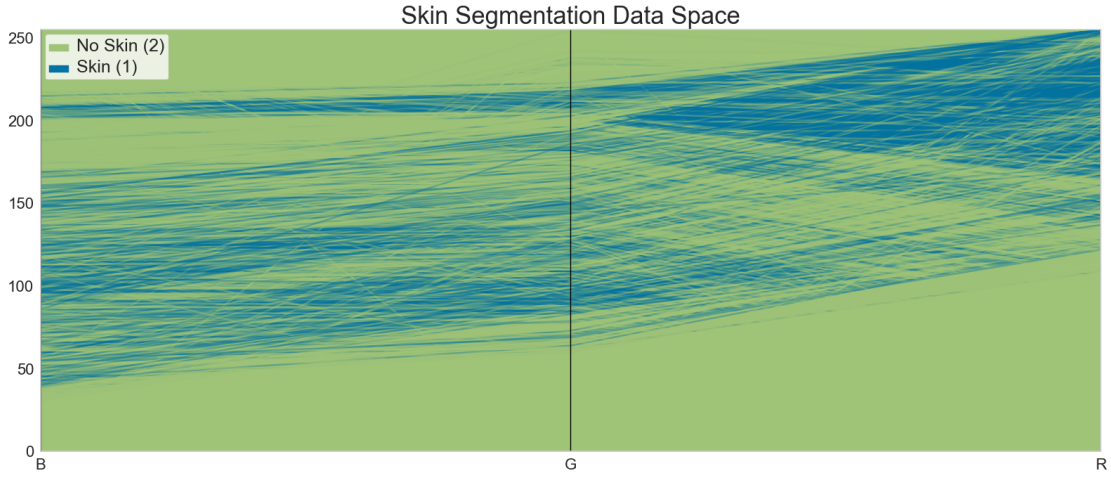


Figure 5.5: PCP over Data Space of Skin Segmentation Dataset

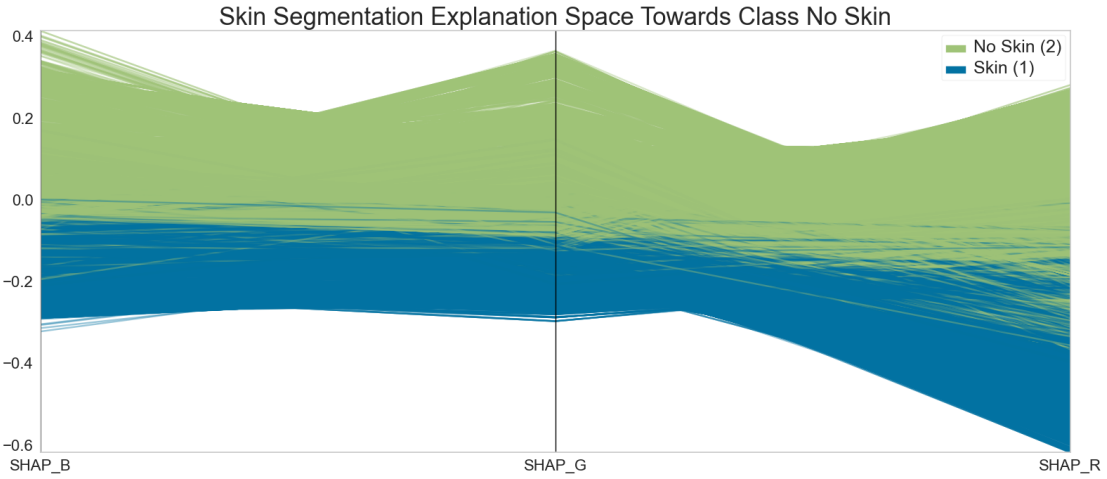


Figure 5.6: PCP over Explanation Space of Skin Segmentation Dataset

5.3 Recap and Discussion of Use Cases

To summarise the experiments and use cases, the Breast Cancer datasets have better results using K-means than HDBSCAN. However, only two clusters were generated for K-means. Therefore, nested clustering occurred, ending in a result of four clusters. After creating regions, the feature importance using SHAP was created where it is evident how the feature importance changes dependent on the datasets of the regions. This dataset proves to be an excellent example of how generating regional explanations can enable more significant insights into the data.

On the contrary, the Skin Segmentation dataset proved to be much more difficult as many observations exist. Secondly, using HDBSCAN either results in many small clusters or a lot of noise and larger clusters. While noise is not an issue, too much noise is often not necessarily desirable.

Chapter 6

Discussion

Regional explanations are not a well-defined concept within XAI. No particular method or definition exists, meaning that there was very little related work to base this thesis upon. Therefore, the main focus was to identify methods as a starting point to explore the concept of establishing regional explanations. Obtaining a profound understanding of the methods enabled insights into the interactions of applying the methods and the corresponding result.

Exploring visualisations was a significant aspect of the thesis, and utilising the PCP enabled a better understanding of the different datasets. The ability to plot high dimensional data proves to be helpful as it enables visual inspection of how the values of the features alternate. Especially between classes, it is interesting to see which features are more alike than others. While the PCP worked well at visualising specific datasets, other datasets, such as the Breast Cancer dataset, were hard to interpret. The Breast Cancer dataset, in particular, contains many features, meaning that the visualisation fast became complex and, therefore, required a lot of attention to detect distinctions between features. Moreover, the Skin Segmentation dataset within the $\boldsymbol{D}_S$ was almost entirely incomprehensible as the amount of data spanned the whole range of values on the y-axis. While the PCP works well overall, certain edge cases fail to provide meaningful information.

Using K-means and HDBSCAN for exploration enables two methods for clustering the dataset, suggesting different ways to establish regions. In Chapter 4, when clustering the Banknote Authentication dataset, it was clear that HDBSCAN was superior to K-means in both spaces. However, in Chapter 5, it was apparent that K-means did well at separating into two clusters for the Breast Cancer dataset. Rarely would a clustering of two clusters provide a great deal of insight; therefore, it made sense to use the same algorithm again to achieve a result of four clusters in total.

In addition to K-means and HDBSCAN, other methods may have been interesting to explore. For example, it would have been interesting to use clustering algorithms or sampling methods that deal well with large amounts of data on the Skin Segmentation dataset. While K-means does well at partitioning a large amount of data, during the exploration, the best k value was often simply 2. It may be plausible to perform nested K-means until reaching a desirable outcome or until the silhouette coefficient is no longer significant. Alternatively, it may be interesting to use a purely hierarchical clustering algorithm to see how the hierarchical clustering results compare to K-means and HDBSCAN.

Besides clustering algorithms, a dozen other distance measures for similarity exploration and clustering validation metrics could also be chosen depending on the dataset and its structure. The same goes for deciding which XAI method to use. To focus on simply one XAI method, LIME was firstly investigated. As SHAP takes inspiration from LIME, and the mathematical foundation of SHAP is superior, the intuitive choice was to explore the usage of SHAP for establishing regional explanations. Other XAI methods are explorable; however, they may not produce the same outcomes as SHAP in the same analysis and context.

Creating a refined and generalisable method for regional explanations is very complex as each dataset is unique and has a distinct data structure. What might work for some dataset is not necessarily applicable to another. It would be interesting to work with a specific goal of what the regional explanations should accomplish. What information is sought after once regions are created? And how many regions would be a suitable amount for the given dataset?

Following an approach like that may make generalising a process or methodology later more easy, than performing many tests on arbitrary datasets with little knowledge towards if it would be better to cluster the data into 2, 5, 10 or 100 regions or something else.

The next chapter will outline potential future work for regional explanations.

Chapter 7

Future Work

During the exploration many ideas appeared for alternative concepts and areas to investigate regarding regional explanations and generally within XAI. This section provides a brief description of potential future work.

Regions for Multi-Class Classification within the D_E

Due to time constraints of the thesis it was not possible to explore how regions are able to be created within the explanation space. A naive approach taken was regarding similarity within the explanation space, but much work can be done to identify how the SHAP values are able to reflect similarity of multi-class datasets.

Regions for Regression Problems

The scope of this thesis did not include regression datasets. The analysis could be extended to include regression problems.

Explore and Compare Other XAI Methods

The chosen XAI method SHAP is simply one of many XAI methods, but grounded it mathematical background and superior visualisations it made a lot of sense to use SHAP. However, it would be interesting to explore other XAI methods and consider taking a new approach to establishing regional explanation give that XAI method.

Explore Visualisation Possibilities of Regional Explanations

During the thesis, a dashboard was established to help analyse different datasets and dynamically highlight and change the ordering of the axes on PCP plot. An interesting extension for regional explanations would be to identify or build new visualisations that can help show an observation and it's similar observations within the regions.

Testing of the Concept of Identifying Wrong Predictions

In section 4.6, a concept for identifying wrong predictions was introduced. Future work of this would test whether it is possible to create some indicator for new samples to determine to what extent they are good or bad predictions.

Utilising Semi-Supervised Learning

Semi-supervised learning could be used to improve regions by using some labels for the clustering. Can Semi-supervised learning help achieve more precise regions?

Chapter 8

Conclusion

Exploring regional explanations has given many directions for future work. This thesis can conclude that attaining regional explanations provided regions is achievable using SHAP. However, the actual establishment of the regions is a complex process, and much more research and testing are still required to establish a formal method.

Additionally, the exploration determines that regional explanations create alternative insights to local and global explanations. While there is a big challenge in creating a formal method for regional explanations, there are many exciting use cases where regional explanations can prove helpful. Regional explanations address the limitations of local and global explanations by identifying similar points to a particular observation. Identifying similar observations allows for comparisons to other similar observations, enabling deep analysis of any sample.

Furthermore, given an observation, it is possible to use the clusterings to gather a similarity threshold, e.g., top 10 similar points and investigate the shared similar observations for each space and combined. While exploring similarity, no significant patterns were found, but the concept may be interesting to explore further, together with the concept introduced about identifying wrong predictions.

Often XAI is a way to analyse a model and see whether the model is performing as intended. By creating regional explanations, even more, concise analysis is possible. By using HDBSCAN, outliers are detectable, making it possible to produce a local explanation for that outlier. Given the outlier, it is then comparable to nearby regions of observations to see more informatively how the outlier differs and why it is deemed an outlier. K-means on the other hand enables hard-partitioning of a dataset, not producing outliers. Dependent on the desired outcome of producing regional explanations, having no noise points may be wanted.

This exploration covered a variety of concepts, some concepts, such as clustering and validation, have more quantitative results. In contrast, other concepts were merely ideas and brief explorations, such as the concept of identifying wrong predictions. In conclusion, a foundation has been established for exploring regional explanations, and the future work ahead signals a challenging but exciting journey.

Bibliography

[Ankerst et al., 1999] Ankerst, M., Breunig, M., Kriegel, H.-P., and Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. volume 28 of *Sigmod Record*, pages 49–60. url: https://dl.acm.org/doi/10.1145/304181.304187.

[Campello et al., 2013] Campello, R. J. G. B., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In Pei, J., Tseng, V. S., Cao, L., Motoda, H., and Xu, G., editors, *Advances in knowledge discovery and data mining*, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Dua and Graff, 2017] Dua, D. and Graff, C. (2017). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining*, KDD'96, pages 226–231. AAAI Press. Place: Portland, Oregon Number of pages: 6.

[Gunning, 2019] Gunning, D. (2019). DARPA's explainable artificial intelligence (XAI) program. IUI '19: Proceedings of the 24th International Conference on Intelligent User Interfaces, pages ii–ii.

[Holzinger et al., 2022] Holzinger, A., Saranti, A., Molnar, C., Biecek, P., and Samek, W. (2022). Explainable AI Methods - A Brief Overview. In Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, K.-R., and Samek, W., editors, *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, Lecture Notes in Computer Science, pages 13–38. Springer International Publishing, Cham.

[Hubert and Arabie, 1985] Hubert, L. J. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

[James et al., 2013] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R*. Springer.

[Johansson Westberg and Forsell, 2015] Johansson Westberg, J. and Forsell, C. (2015). Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. *IEEE Transactions on Visualization and Computer Graphics*, 22:1–1.

[Lundberg and Lee, 2017a] Lundberg, S. M. and Lee, S.-I. (2017a). SHAP documentation. https://shap.readthedocs.io/en/latest/index.html.

[Lundberg and Lee, 2017b] Lundberg, S. M. and Lee, S.-I. (2017b). A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[McInnes et al., 2017] McInnes, L., Healy, J., and Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205. https://hdbscan.readthedocs.io/en/latest/index.html.

[Molnar, 2022] Molnar, C. (2022). *Interpretable machine learning. A Guide for Making Black Box Models Explainable*. Independently published, 2 edition.

[Moulavi et al., 2014] Moulavi, D., A Jaskowiak, P., Campello, R., Zimek, A., and Sander, J. (2014). Density-based clustering validation.

[Nadikattu, 2016] Nadikattu, R. R. (2016). THE EMERGING ROLE OF ARTIFICIAL INTELLIGENCE IN MODERN SOCIETY. 4:906–911.

[OECD, 2019] OECD (2019). *Artificial intelligence in society*. OECD.

[Ribeiro, 2016] Ribeiro, M. T. (2016). Lime Python Package — v.0.1 documentation. https://lime-ml.readthedocs.io/en/latest/index.html.

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the Demonstrations Session, NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 97–101. The Association for Computational Linguistics.

[Rousseeuw, 1987] Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

[Shapley, 2016] Shapley, L. S. (2016). 17. A Value for n-Person Games. In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games (AM-28), Volume II*, pages 307–318. Princeton University Press.

[Tan et al., 2020] Tan, P.-N., Steinbach, M., Karpatne, A., and Kumar, V. (2020). *Introduction to Data Mining*. Pearson, second edition edition.

[Virtanen et al., 2020] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, , Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272.

[Zaki and Meira, 2020] Zaki, M. J. and Meira, Jr, W. (2020). *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge University Press, 2 edition.