UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

André Correia Lacerda Mafra

Truth or Utility: Transductive Regularizer for Feature Selection

Belo Horizonte
2023

André Correia Lacerda Mafra

Truth or Utility: Transductive Regularizer for Feature Selection

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Nivio Ziviani
Co-Advisor: Adriano Veloso

Belo Horizonte
2023

© 2023, André Correia Lacerda Mafra.
Todos os direitos reservados

Mafra,André Correia Lacerda

M187t Truth or utility [recurso eletrônico]: transductive regularizer for feature selection / André Correia Lacerda Mafra — 2023.
1 recurso online (62 f. il, color.)

Orientador: Nivio Ziviani.
Coorientador: Adriano Alonso Veloso.
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Departamento de Ciência da Computação, Instituto de Ciências Exatas.
Referências: f. 56-62.

1. Computação – Teses. 2. Aprendizado de máquina – Teses. 3. Transdução - Teses. I. Ziviani, Nivio II. Veloso, Adriano Alonso III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. IV. Título.

CDU 519.6*32(043)

Ficha Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa CRB 6/1510 Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

TRUTH OR UTILITY: TRANSDUCTIVE REGULARIZER FOR FEATURE SELECTION

ANDRÉ CORREIA LACERDA MAFRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Nivio Ziviani - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Adriano Alonso Veloso - Coorientador
Departamento de Ciência da Computação - UFMG

Prof. Heitor Soares Ramos Filho
Departamento de Ciência da Computação - UFMG

Prof. Anderson da Silva Soares
Instituto de Informática - UFG

Belo Horizonte, 01 de março de 2023.

Documento assinado eletronicamente por **Nivio Ziviani**, **Professor Magistério Superior - Voluntário**, em 19/07/2023, às 17:18, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Adriano Alonso Veloso**, **Professor do Magistério Superior**, em 19/07/2023, às 17:52, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Anderson da Silva Soares**, **Usuário Externo**, em 18/09/2023, às 10:29, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Heitor Soares Ramos Filho**, **Professor do Magistério Superior**, em 18/09/2023, às 11:20, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2474147** e o código CRC **7044354D**.

I Dedicate this work to my parents. Without you I would have never made it.

Acknowledgments

First, I thank my family, parents and siblings for all the investment made for my education and for the support given during this time of my life.

Then my girlfriend Thaís for her partnership and words of encouragement, so I could finish this work.

Lastly, I thank Nivio and Adriano for the mentoring and CNPQ, CAPES and Fapemig.

"*Not all those who wander are lost*"
(J.R.R Tolkien)

Resumo

Em Machine Learning, a generalização é muito desejável, pois evita que o desempenho do modelo decaia consideravelmente com novos dados. No entanto, encontrar regras gerais é muito difícil, uma vez que visa modelar uma solução que funcione em qualquer conjunto de dados possível. Portanto é muito comum que um modelo não alcance resultados satisfatórios em um conjunto de teste, mesmo que diferentes técnicas de generalização sejam aplicadas. Dito isto, a transdução é uma técnica que, ao contrário da indução, visa inferir uma solução aplicável a um conjunto alvo específico B, a partir de um conjunto de fontes A. Desta forma, a transdução não tenta resolver um problema geral, ela foca em dar uma solução para um conjunto de dados específico, o que reduz drasticamente a complexidade de encontrar uma boa solução. A motivação para este trabalho é o fato de que os modelos de aprendizado de máquina têm dificuldade para generalizar, por outro lado, alguns cenários estão interessados apenas em uma solução que seja boa o suficiente para um conjunto específico de dados. Este cenário se beneficia mais da transdução do que da indução e reduz consideravelmente o nível de dificuldade para encontrar uma boa solução para o conjunto de dados específico. O objetivo deste trabalho é conseguir selecionar características que otimizem os resultados em um conjunto alvo específico, partindo de um conjunto fonte, onde não temos rótulos para o conjunto alvo B, que são cenários onde não é possível retreinar o modelo para B ou aplicar a aprendizagem indutiva. O método proposto pode ser aplicado a qualquer algoritmo de seleção de características existente, pois se propõe a continuar otimizando o algoritmo por qualquer uma das métricas que ele já utiliza, além de uma nova métrica que mede a correlação da importância de um conjunto de atributos tanto no treinamento quanto no conjunto de teste.

Palavras-chave: aprendizado de máquna; transdução.

Abstract

In Machine Learning, generalization is very desirable, because it prevents that model performance decays considerably with new data. However, finding general rules is very hard, once it aims to model a solution that works in any possible dataset. Therefore is very usual that a model does not achieve satisfactory results in a test set, even though different techniques of generalization are applied. That said, transduction is a technique, that in contrast with induction, aims to infer an applicable solution to a specific target set B, starting from a source set A. In this way, transduction does not try to solve a general problem, it focuses on giving a solution to a specific dataset, which drastically reduces the complexity of finding a good solution. The motivation for this work is the fact that machine learning models struggle to generalize, on the other hand, some scenarios are only interested in a solution that is good enough for a specific set of data. This scenario benefits more from transduction than induction and reduces considerably the level of difficulty to find a good solution to the specific dataset. The goal of this work is to be able to select features that optimize the results in a specific target set, starting from a source set, where we do not have labels for target set B, which are scenarios where can not retrain the model for B or apply inductive learning. The proposed method can be applied to any existing feature selection algorithm, because it proposes to continue to optimize the algorithm by any of the metrics it already uses, plus a new metric that measures the correlation of the importance of a feature set in both training and test set.

Keywords: machine learning; transduction.

List of Figures

List of Tables

List of Algorithms

Contents

Chapter 1

Introduction

In this work, we use a transduction learning approach instead of generalization to select features, because generalization is a difficult problem, especially in the scenarios described in this work, where we applied our methodology to two very complex problems of diagnosing Alzheimer's and Heart Disease. Therefore, transduction is a good alternative to generalization, because it assumes a problem with less complexity.

The objective of this study is to identify features that maximize performance specifically in a target set B, given a source set A. In scenarios where retraining the model for the target set or employing inductive learning is not feasible, this becomes particularly relevant. The proposed approach can be utilized in conjunction with any existing feature selection algorithm. It aims to further enhance the algorithm by incorporating an additional metric that evaluates the correlation of feature set importance between the training and test sets. By integrating this new metric alongside the existing optimization metrics, the proposed method enables improved feature selection in a wide range of applications.

1.1 Motivation

Suppose two hospitals, A and B. Hospital A developed a model for diagnosing its patients with Alzheimer's using its own data. Given the success, hospital A tries to sell the model to hospital B, however, the model performance was not as expected for hospital B.

Typically we train a model from A to be used in A as well, but what if we need to use it in B? One alternative would be retraining the model with hospital B data, for that we typically need labels for B as well. In this work, we will work on scenarios where we do not have labels for B.

In this work, we propose a transductive regularizer for feature selection. This approach would allow us to select different set of features for hospital B and this new set of features is potentially better for hospital B scenario. We could then retrain the model

with the new features using the labeled hospital A dataset and, in this way transduction does not need for the hospital B dataset to have labels.

Sometimes transduction can be confused with Transfer Learning because usually Transfer Learning is referred to as Inductive Transfer Learning, which involves the usage of large models trained on heavy datasets on smaller ones. Transfer Learning relies on Fine Tuning of Hyperparameters and will work only if the target dataset has labels. On the other hand, as presented in the paper *A Survey on Transfer Learning* [48] the Transductive Transfer Learning is the scenario where the source and target tasks are the same, while the source and target data domains are different. Transduction does not use fine-tuning and does not need labels, because Transduction is all about domain adaptation, by generating a model from a source dataset that can be used on an unlabeled target dataset.

1.2 What is Transduction?

The word transduction is always associated with converting a signal to another form. In biology, transduction refers to the process of a microorganism transferring genetic material to another microorganism [45], and in the field of electronics, a *transducer* takes the input signal and converts it to an electrical signal.

In Machine Learning, Transduction is a learning technique, as well as induction, also known as generalization. Induction seeks to find a general model for any set of test data by inferring general rules from a specific set of training, whilst transduction seeks a specialist model for a specific set of target data learned from different sets of training, called source. Usually, generalization is a much more difficult problem than transduction. It is harder for a model to generalize because induction requests a large amount of data to be used as training, in some cases the amount of available data is not enough.

Transduction algorithms can be generally classified into two main groups: those aimed at assigning discrete labels to unlabeled data points, and those aimed at predicting continuous labels for unlabeled data points. The algorithms that focus on predicting discrete labels often involve incorporating partial supervision into clustering algorithms. There are two classes of algorithms commonly employed: flat clustering and hierarchical clustering. The latter can be further divided into two subcategories: partitioning-based clustering and agglomerative clustering. On the other hand, algorithms that aim to predict continuous labels are typically developed by incorporating partial supervision into manifold learning algorithms.

An advantage of transduction is to potentially make superior predictions using

a smaller number of labeled points, as it leverages the inherent patterns present in the unlabeled data. However, one drawback of transduction is the, usual, absence of a predictive model. When a previously unknown point is introduced, the entire transductive algorithm must be rerun with all the points to predict its label. This process can be computationally demanding, especially when dealing with incremental data streams. Additionally, this could potentially lead to changes in the predictions of existing points, which may have positive or negative implications depending on the specific application. In contrast, supervised learning algorithms can promptly label new points with minimal computational overhead.

1.3 Truth or Utility

Induction seeks the truth, which means, for example, finding a solution to correctly diagnose every Alzheimer's patient in every existing hospital. On the other hand, transduction seeks utility, for example, finding a solution to diagnose Alzheimer patients from a specific hospital. We can conclude that induction is meant to find a universal truth, that is a model that was able to generalize beyond the training data and model the phenomenon behind it so well, to a point where the solution could be used in any sample of new data and it would never degrade in performance. That is why generalization is so difficult, and transduction is a promising alternative, once it aims to find a solution that is *useful* to only one specific scenario, which drastically reduces the scope of the problem, thus decreasing complexity.

Then, when to use each technique? It depends on the coverage of the solution, whether it is desired to have a model robust enough to be used in different scenarios, such as an Alzheimer diagnosis model that could be deployed in many different hospitals, or to have a model useful enough to a specific scenario, such as an Alzheimer diagnose model that needs to be deployed in only one hospital.

Usually, models are trained and tested to work in one specific set of data, however, to optimize it, generalization techniques are used, such as regularization, ensembles, and early stopping [38] which turn it to a more difficult problem. The idea is to use techniques that seek to solve the problem for that one specific set only, which is the case for transduction techniques.

1.4 Objectives and Contributions of the Work

In this work, we propose to adapt feature selection methods to optimize the transduction of the model, usually, feature selection is associated with induction, once it aims to reduce dimensionality to ease the generalization. In this case, transduction is being used to select the best set of features to be used in the target dataset, meanwhile, we only use the source dataset for training and the target dataset does not need to have labels. Then we train a model with these selected features using the labeled source dataset and evaluate it in the target.

The hypothesis behind our methodology is that if a feature is important in the source dataset and it is also important in the target dataset, then it is likely capable of contributing to the target dataset too. For this we measure the overall score of importance of a set of features in the source and target datasets, then we calculate the similarity between the scores in both datasets. We call this value *feature similarity* and it indicates if the features are equally important in both datasets, besides that, we also calculate the AUC of the feature set in the source dataset, then we sum the AUC and the feature similarity for all the possible combinations of feature sets, and we choose the one that maximizes AUC + *feature similarity* because we want features that can be important in both datasets but are also good predictors. As explained in the methodology chapter, we needed to add a transduction regularizer, called c to the equation to control the weight of the feature similarity. It is called a transduction regularizer because the feature similarity represents the transduction learning factor. After all, it is a way to evaluate a model trained on the source dataset on the target dataset. So the final equation is AUC + c* *feature similarity*.

The data used in this work are derived from different hospitals regarding Alzheimer's disease and Heart diseases. All of these datasets are small in size and have labels, but in our experiments, the dataset selected to be the target has its targets removed. Therefore one of the future works is to apply our approach to datasets with more samples.

The main contributions of this work are:

- the proposal of a transduction regularizer variable for feature selection algorithms using the explainability of features and similarity of matrices.
- the usage of models on target datasets that do not have labeled data. To have labeled data can be very expensive, especially for health data scenarios, where you need certified professionals to label the data for you.
- to build a strategy that achieves results with +0.166 points in AUC, an increase of 28% compared to a generalization method.

Chapter 2

Related Work

In this work we use the feature importance tool called SHAP (SHapley Additive exPlanations) [44] in our framework to select features, which is widely used today [24] as a manner to rank features by their importance and pick only the top ones in the ranking. However, our approach is different because we combine SHAP with Mantel test [46] to create an equation that measures a model performance in the source dataset and the model usefulness in the target dataset, which is to transduce learning from the source to the target. Transduction was introduced by Vladimir Vapnik around 1990, According to Vapnik, when solving a problem of interest, one should not try to find a solution to a more general problem first, but rather try to find the answer to the specific problem you want to solve [62].

2.1 Induction Learning Models

The lifecycle of a machine learning model does not end when the training and validation stage is finished, as it is after deployment in an external environment that the model is exposed to new data all the time, at this point many effectiveness problems emerge [6, 43, 57, 59]. The reason behind these problems is that models are created when trying to optimize an objective function for a database, which is just a sample of the real world and does not have all the characteristics of the phenomenon, in addition to noise. On the other hand, modeling the phenomenon is a very general and therefore very complex problem.

When a machine learning model is trained, it is expected to be able to generalize to any input data and not just the ones it was trained for. However, the data used in training are only an observed sample of a phenomenon. Such a phenomenon is what every scientist would like to be able to accurately reproduce in a model, but the sample data is not capable of storing all the characteristics of the phenomenon and often brings with it other noises that make it difficult to identify what is most relevant, and therefore that is,

the best models are just approximations of the phenomenon with a bias from the sample data.

The difference between the phenomenon and the sample data is perceived when comparing the metrics in the different stages of the evaluation of a model, the results with sample data tend to decline at each stage from the training with cross-validation to the evaluation in a validation sample and finally in a test sample of type *out of sample*. This model degradation effect becomes even clearer when a model is exposed to real-world data, outside the scientist's controlled environment.

Usually, the generalization capacity of a model is measured through a subset of the sample data that is not used in training, such subset is usually called holdout test set or test set. If the performance of the model declines too much on the test set concerning the training data, it can be said that the model fails to generalize. When this happens, the scientist will have to go back to the model conception stage and re-evaluate several possible causes of the problem, one of the possibilities is that the model is not managing to absorb the most important features, as some features can bring the noise to the model and not contribute to generalization. Machine Learning and Data Mining techniques have already made important advancements in various domains of knowledge engineering, including tasks such as classification, regression, and clustering. (e.g., [65, 66]).

2.2 Transfer Learning Models

Several induction-based learning algorithms exhibit optimal performance solely when operating under the assumption that both the training and test data originate from an identical data distribution. Should there be any alterations to the distribution, the majority of statistical models necessitate a complete reconstruction using freshly acquired training data. In numerous practical scenarios, the cost or feasibility of reacquiring the training data is prohibitive. An illustration of this challenge is found is the healthcare data domain, where labeled data require expensive employees to manually review exams and label a diagnosis for each patient. In such cases, transfer learning between data domains would be desirable.

The study of Transfer learning is driven by the recognition that humans possess the ability to effectively utilize previously acquired knowledge to solve novel problems more efficiently or with improved solutions. Transfer learning can prove highly advantageous in various instances within the realm of knowledge engineering. One example is Web-document classification [1, 25, 55], where the objective is to categorize a given Web document into predetermined classes. For instance, in the domain of Web-document

classification, the labeled examples may consist of journalistic web pages associated with category information derived from previous manual labeling efforts. However, when faced with a classification task involving a newly created website, the availability of labeled training data may be scarce due to differing data features or distributions. Consequently, directly applying web page classifiers trained on the journal website becomes impracticable. In such scenarios, the ability to transfer classification knowledge to the new data domain would be helpful.

Transfer learning techniques have demonstrated successful application in numerous practical scenarios. For instance, [51, 14, 15] proposed to use transfer learning techniques to learn text data across different domains, respectively. [5] suggested the use of structural correspondence learning (SCL) to address NLP problems, and an extension of SCL was later proposed by [4] for solving sentiment classification problems. [63] put forward a methodology that leverages both limited target domain data and abundant but low-quality source domain data for image classification problems. [2] introduced transductive transfer learning methods specifically designed to tackle name-entity recognition problems.

In the inductive transfer learning, the target task is different from the source task. In this case, some labeled data in the target domain are required to *induce* learning for use in the target domain. On the other hand, the transductive transfer learning, the source and target have different tasks, but come from the same data domain.

2.3 Transduction Learning Models

Transduction learning models are different from transductive transfer learning models, because for transfer learning you still need the target to have labeled data, meanwhile, it is not necessary for transduction learning models. Transduction learning algorithms started to become popular with the Transductive Support Vector Machines (TSVM) [36], which are a variation of the popular SVM, but now the test data is unlabeled, which makes it possible to learn on the test data and get better classifications, on the other hand optimizing TSVMs is very hard. Transduction is still very used in many segments of Machine Learning today, in the field of Neural Networks and Natural Language Processing, transduction is very used in architectures of the type Encoder-Decoder, such as the Recurrent Neural Networks (RNN). In general, Transductive Learning Algorithms are usually classified based on the instance, and maybe the most famous algorithm is the *k-Nearest Neighbors* [13, 27] which does not model a generalist function, given a set of training, but directly uses it for each instance where the prediction is requested.

In linguistics, transduction has been used when referring to natural language. For

example, there is the idea of a "transduction grammar" that refers to a set of rules for transforming examples of one language into another. There is also the concept of a *finite state transducer* (FST) from the computation theory that is commonly related to translation tasks for mapping one set of symbols to another [37, 39].

Transduction is also very commonly used for sequence prediction tasks, Yoav Goldberg defines a transducer as a specific network model for NLP tasks [29]. Goldberg suggests using this particular model for both sequence tagging and language modeling purposes. Furthermore, they suggest that conditioned generation, exemplified by the Encoder-Decoder architecture, can be viewed as a specific instance of the RNN transducer. This aspect is unexpected, as the Decoder in the Encoder-Decoder model architecture allows for a flexible number of outputs for a given input sequence, deviating from the traditional "one output per input" definition.

In the realm of NLP sequence prediction tasks, particularly in translation, transduction is commonly employed. The definitions in this context appear to be less stringent compared to the rigid requirement of one output per input as presented by Goldberg and the FST approach. For example, Grefenstette in [32] describes transduction as mapping an input string to another output string. Alex Graves [31] also employs transduction as an equivalent term for transformation and, notably, offers a valuable compilation of NLP tasks that align with this definition.

Transduction algorithms work very well with small datasets because they do not seek generalization, usually, healthcare data consists of small data samples, because it takes too much work from doctors and other healthcare employees to label data. In [41] transductive support vector machines showed promising results in small and unbalanced datasets of molecular quantitative structure activity relationship. Transduction learning was also used to detect health change in [34], their studies showed that a transductive approach in more accurately detecting unhealthy entities with less supervision compared to other strong baselines.

2.4 Feature Selection Algorithms

Feature selection is a very important step in the machine-learning process. It is during this step that we can remove features that do not contribute significantly to the performance of the model, and in this way, two fundamental problems of modern data science can be solved.

The first problem is that databases can be very large, which makes learning algorithms run slower and more computational resources are used. These resources, such

as CPU, GPU, RAM, and Hard Disk are limited and have a cost associated with their use and many users do not have the financial conditions to acquire more resources, which requires that the model, somehow, needs to run in the current conditions.

The second problem is that the more variables, the more complex the model becomes. This is bad because it makes explainability difficult and many algorithms have a performance drop when the number of features is significantly higher than the optimal [40]. This goes against *Occam*'s razor principle which says that simple solutions tend to be more correct than complex ones. In the field of data science, Occam's razor is the principle used to deal with *overfitting*. *Overfitting* describes the scenario where a model has captured too much noise from the data and is unable to find patterns that generalize beyond the observed data. Feature selection helps to reduce the chances of *overfitting* by removing variables irrelevant to the problem, helping to keep the model simpler.

There are two main types of feature selection algorithms according to the surveys of [11, 67]:

- Wrapper techniques assess multiple trained submodels that are chosen either through sequential elimination (e.g., stepwise forward/backward) or a heuristic search algorithm. Although these evaluations driven by the models are external to the data, they are inherent to the particular modeling objective.

- Filter techniques are frameworks that are not reliant on a specific model and circumvent the computational load associated with model training found in wrapper methods. These techniques rank features based on empirical estimates of inherent data properties, such as covariance or mutual information. Both of the aforementioned approaches can be employed within the framework of Shapley values.

A simple Shapley value feature selection method would compute the Shapley value for every feature and then select the k highest ranking features. Upon review of the literature we found articles that uses variations of this simple feature selection method with Shapley value, such as [33, 61],

That said, it is understood that feature selection is a very important step in the development of a machine-learning model and can actively collaborate with the generalization of the model to the phenomenon. However, the problem of finding a generalist solution remains very difficult, even with the application of feature selection, as it is very common for models to have a much lower performance on new data. If the problem requires finding a general rule for the phenomenon, other techniques such as regularization, ensembles, and early stopping can be used [38], however, if the greatest interest is classifying a specific set of data beyond the training set, the transduction technique may be preferable, as it tries to solve a minor problem, rather than creating a regal rule for a [26] phenomenon.

2.5 Similarity Matrices Applications

We use the Mantel test to measure the correlation between two distance matrices composed of Shapley values from both source A and target B datasets for a given set of features. Then, for each set of features, we can estimate how close is the performance of that group of features in a model applied for both source and target datasets. The Mantel test has some constraints, for example, the test only calculates the correlation between two matrices at a time and both matrices must be of the same dimension. Originally the test was created by Nathan Mantel, a biostatistician to estimate the correlation of biological species [46].

The test is frequently employed in the field of ecology, where the information generally comprises approximations of the "gap" between entities like organism species. For instance, a matrix could encompass estimations of the genetic disparities (i.e., the level of dissimilarity between two distinct genomes) among all feasible species pairs under investigation, acquired via molecular systematics techniques. Meanwhile, another matrix could involve estimations of the spatial separation between the habitats of each species concerning every other species. In this scenario, the tested hypothesis aims to determine whether the genetic variability among these organisms exhibits a correlation with the geographical disparity. Irrespective of the methodology employed to quantify the genetic variation between populations, a recurring objective in landscape of genetics is to assess the level of spatial patterns present in the genetic distance matrix. For instance, it is typical to utilize clustering methods, such as k-Nearest Neighbors and Principal Component Analysis (PCA), to visually represent the connections between populations based on these matrices. However, these approaches do not explicitly assess the influence of geographic space. By far, the Mantel test is the most widely utilized approach to examine the relationship between geographic distance and genetic divergence [19].

In this work, we used the Mantel test as a distance measure, there are many other ways to measure the distance between two matrices, mostly derived from Euclidean geometry such as the Cosine distance and Manhattan difference. Both methods are usually used in clustering algorithms, such as K-Nearest Neighbors, in our example we kind of cluster our feature sets, into transduce-able or not transduce-able, where transduce-able are those feature sets that have similar performance in both Source A and Target B.

There are many applications of distance measures algorithms [16, 28], which are usually trying to cluster training set data with test data to derive the test point a label, so the closer the test point is to a certain training point, the higher is the probability that both have the same label. In our work, we compared the Mantel test with other distance measures, but the Mantel correlation was better to identify potential transduce-able feature sets.

2.6 Other Machine Learning Techniques Applied in the Healthcare Field

The use of Machine Learning techniques in the health area is very extensive, but we can highlight some main applications, such as disease diagnosis based on text and image, and disease prediction focused on an early diagnosis [22], in which, generally, the diagnosis of professionals can be more inaccurate. The first topic is well illustrated by works on Alzheimer's diagnosis [49] with CNN networks and the second topic is commonly associated with the diagnosis of cancer, still in the early stage of the disease [60].

Early diagnosis is not an easy task, as people usually only know the symptoms associated with the most severe phase of an illness. Serious diseases such as cancer can manifest themselves in different ways in the body in its initial phase, such as fatigue, cough, body aches, etc. Such symptoms may go unnoticed by a layman [47], which underscores the importance of public awareness campaigns that encourage people to seek medical help for tests [42].

To discover changes of clusters in real data over time is an area of study commonly called *Cluster Migrations* and is widely used in contexts of fraud detection [17] and financial markets [53]. However, it can also be applied in the sentinel detection scenario, as these are the factors that make the cluster change over time. One of the possibilities with *Cluster Migrations* is the creation of visualizations that show the migrations of clusters, facilitating a temporal analysis of the problem, and allowing the analysis not to be assured in only one point in time.

One of the Machine Learning areas that have been really used lately in the health environment is Causality. Because what we really want to determine is the factor that caused a disease or is related to the onset of the disease. An interesting area of research along these lines is determining which drugs are responsible for causing adverse drug effects.

Adverse drug effects account for 6.5% of hospitalizations in the UK [50] and studies indicate that the mortality rate increases annually [64]. Studies in this line [54] were able to correctly rank the drug families in the order of those with the highest probability of causing Myocardial Infarction, surpassing simple Cox regressions, normally used in the health area. The idea of the study [54] is to eliminate confounding variables from features and be able to use mining of emerging patterns to infer causality and use regression to rank drugs.

In this work, we use a very common dataset in the field, which is the UCI Heart Disease Dataset. This dataset is vastly used in many types of research, In [7] it was used to train Neural Network ensembles, in [35] it was used for a Minimax approach and in

[21], for instance, the dataset was used to train a genetic programming algorithm.

2.7 Contrasting our Approach with Existing Literature

As of today, this work is very unique in the literature, with little research in the field of transduction. This work could be compared to Transfer Learning works and Self Supervised Learning, but it still differs from them, as will be demonstrated. In the paper *A Survey on Transfer Learning* [48], the authors bring a benchmark of the different tasks on which, transductive learning is applied, They mention Regression and Classification tasks, like using Transductive Support Vector Machine(TSVM) and k-Nearest Neighbors, which are instance based algorithms. But there is no citation about transfer learning in the context of feature selection.

Most works using transductive learning approaches with health care date only use TSVM, [34, 41], meanwhile our work uses a different approach, by optimizing feature selection based on their feature importance and transduction level with Mantel similarity.

In the literature, no other research has used SHAP for transduction, there is no other work that used Matrix similarity for transduction learning. Usually, transduction in the literature is only applied by converting a symbol of input to another symbol of output, as a conversion or transformation step, such as in speech recognition, machine translation, and text-to-speech to name but a few. Then, transduction is commonly associated with a not supervised learning algorithm, but in our work, we use transduction as a way to transfer learning from one source dataset with labels to another target dataset with no labels, instead of trying to infer based on the instances of the target dataset only, as is done in most regression and classification tasks [13].

Chapter 3

Background

In this chapter, we provide important definitions used in our work.

3.1 Types of Learning Techniques

There are at least two different types of learning within Statistical Inference: Transductive and Inductive. Statistical inference is the process of using data analytical tools to infer properties of a probability distribution, that is, probabilities of occurrence of possible outputs of an experiment.

Induction is the technique used to train a model on a Source dataset and apply it to a never seen Target dataset [8]. It is a way of learning that seeks to find an approximate function $f(x) = y$, which maps the data x with your predictions y. So the induction seeks to find a single function that can generalize to any example of x, for this reason, induction tries to solve a more difficult problem. In contrast, Transduction is a technique that uses two already known datasets, the model is trained on the source dataset and applied on the target dataset.

The term transduction means to convert something into another form and is a popular term in the field of electronics and signal processing, where transduction is a way of converting signals, such as sound waves being transformed into energy electricity within a system. The term transduction is also used in other areas, such as biology when a microorganism transfers genetic material to another microorganism [45]. In many of these areas, transduction always has the meaning of transferring something thing from point A to point B. In the field of machine learning, Transductive learning refers to predicting specific examples from database B, given a training set A. Transduction is often used in many segments of machine learning, especially in Natural Language Processing with Encoder-Decoder architectures [30].

3.2 Feature Selection

Feature selection is a very important step in the training process of a machine learning model, which consists of selecting a subset of features from the database for the final version of the model. This step is important for several reasons, such as:

- Reduce the computational cost of model training.
- Simplify the interpretation of the most important features since the number of features is smaller and the importance of each one has a greater weight in the final decision.
- Reduce data sparsity to avoid the curse of dimensionality, which refers to the fact that too many dimensions diminish the ability of machine learning algorithms to be efficient at finding patterns [40, 56].

From a theoretical point of view, feature selection can be considered a regularization method, as it contributes to *Occam*'s razor principle, which says that simple solutions tend to be more assertive than complex ones. Regularization seeks to decrease the chances of overfitting, which describes the scenario in which the model has captured too much noise from the data and is unable to find patterns that allow generalization. Therefore, feature selection assumes that there is an optimal amount of features, and from that point on, if more features are added, the model will decrease its generalizability. The curse of dimensionality interferes with the predictive power of algorithms, by hampering their ability to discern different classes of data when there are a large number of features. This phenomenon is also studied in other areas of knowledge, such as Neurology, in the [68] article, *Feature Selection for Inductive Generalization*, the authors explore how the human brain performs the selection of image features to classify them among different species of animals.

That said, many works in the literature explore feature selection as a way to contribute to the inductive generalization of a model. On the other hand, there are several scenarios in which one only seeks to solve a specific problem, in a single data set, without it being necessary to find general rules for the phenomenon. To meet these cases, this work seeks to assess whether feature selection also contributes to model transduction.

3.3 Feature Importance

Feature importance is a measure to evaluate the individual contribution of each single feature to the model's final output. There are many ways to calculate the importance of a feature, in ensemble algorithms, such as XGBoost [12] the feature importance is a functionality already implemented in the library. But one of the most popular algorithms is the SHAP (*SHapley Additive exPlanations*), which is a game theory approach that can give a ranking of feature importance to any model. The importance of a given feature in SHAP is given by the SHAP Value, which evaluates the feature contribution by assuming that every feature is a player in a game, where the prediction is the reward. What the SHAP value tells is the best way to distribute the reward amongst the players.

The SHAP algorithm is probably the state of the art in Machine Learning explainability, created by Lundberg et al [44], it seeks to reverse engineer any Black-Box predictive model to explain its output. SHAP values are used to explain any model output, it could be a gradient boosting, a neural network or any other model that takes a set of features as input and outputs predictions.

SHAP values are derived from Shapley values, a principle originating from the realm of game theory. However, game theory necessitates the presence of two essential components: a game and participating individuals. How does this concept relate to the comprehensibility of machine learning? Let's envision a scenario where we possess a predictive model, and in this context:

- the "game" is reproducing the outcome of the model.
- the "players" are the features included in the model.

Shapley's function involves assessing the impact that each participant has on the overall outcome of the game. Similarly, SHAP aims to quantify the influence that each individual characteristic has on the prediction generated by the model.

It is crucial to emphasize that when we refer to a "game," we are actually referring to a solitary instance or observation. Each observation constitutes a distinct game. In fact, SHAP primarily focuses on providing local interpretability for a predictive model. SHAP values are matrices in the *n-samples* x *n-features* format and each cell of the matrix contains the SHAP value of a given feature in a given sample.

Once the Shapley value can tell the importance of a feature, many feature selection algorithms use it to rank all features by their importance and select only the k most important features. In this work, the Shapley Values are also used to measure the importance of features, but instead of choosing the first ones in the ranking, the chosen features are the ones that are important in both datasets source A and target B.

In this work, SHAP values are the mechanism of action of a model. This means that the features considered important by SHAP determine the model output. Then, models with a similar mechanism of action, i.e with similar SHAP Values tend to have similar outputs

3.4 Matrix Similarity

The Matrix similarity is derived from the Mantel Test, which is a statistical test to find the correlation (similarity) between two matrices of the same dimension. The test was invented by Nathan Mantel, a biostatistician in 1967. The test is widely used in ecology to estimate the similarity between two characteristics of species of organisms. For example, a matrix can contain the genetic similarity of two species, that is, the similarity of two different genomes, while another matrix can count the geographic similarity of the zone where the animals are located. In this case, the hypothesis being tested is whether genetic variation is correlated with geographic location variation.

The Mantel test tests the correlation between two similarity matrices. These similarity matrices are preprocessed before, working like distance matrices. It is a non-parametric test and calculates the statistical significance of the correlation by permutating rows and columns. The test uses Pearson's correlation to determine the similarity coefficient r, which exists within the range of -1 and +1, where closer to -1 indicates a strong negative correlation and proximity to +1 indicates a strong positive correlation. A value of r equals to 0 indicates that there is no correlation. Therefore, when performing the test to evaluate the correlation between two matrices, the statistical significance of the test is evaluated, by verifying that the *p-value*, is generally less than 0.05, as it represents a confidence interval of 0.95.

In contrast to the conventional application of the correlation coefficient, where the assessment of any potential deviation from a zero correlation is based on its significance, a different approach is taken. This alternative method involves subjecting the rows and columns of one of the matrices to numerous random permutations, with the correlation coefficient recalculated after each permutation. The significance of the observed correlation is determined by the proportion of permutations that result in a higher correlation coefficient.

The underlying reasoning is that if the null hypothesis, suggesting the absence of any relationship between the two matrices, holds true, then randomly permuting the rows and columns should equally likely produce a larger or smaller coefficient. By utilizing the permutation test, not only are the issues stemming from the statistical dependence

of elements within each matrix addressed, but it also eliminates the need to rely on assumptions regarding the statistical distributions of the matrix elements.

In this work, the Mantel similarity was used to estimate the similarity between the two feature importance matrices, from the training and test sets, respectively. In this case, the hypothesis being tested is whether the variation of the contribution of the features in the training set is correlated with the contribution of the features in the test set.

Overall, our work uses the background previously described to propose a transductive regularizer for feature selection algorithms, to select optimal features for a target dataset B, by choosing a set of features that is equally important in both source dataset A and target dataset B, for this we calculate the mantel similarity of the Shapley Values for the candidate set of features applied on both datasets.

3.5 Performance Metrics

One of the most important ways to assess a model's usefulness is to measure its performance, which essentially tells us how decent is the model. But there are many performance metrics, and each one of them is best suited to evaluate a different aspect of the model's usefulness. In this work, we use the Area Under the Curve (AUC) to measure how well a classifier can distinguish between two classes. It does so by calculating the area under the Receive Operator Characteristic (ROC) curve, which plots the True Positive Rate against the False Positive Rate at various thresholds. The higher the AUC, the better the performance and when AUC is close to 0.5 the model is as bad as a random predictive model. A good AUC value means the model can split data into its labels correctly to at least some thresholds.

The AUC can be pictured as a metric that tells how well the model orders the predictions. When the model outputs a prediction it is just a probability for a class (has Alzheimer (1) or does not have Alzheimer (0) for example), as we use classification models we want the prediction to be able to be translated into a binary value, for that we need to define a threshold. For example: if prediction > 0.5, the class is positive for Alzheimer's (1), otherwise, it is 0. However, we can have any threshold value, this is where the AUC comes in, for each threshold it calculates the rate of True Positives (Customers classified as positives that are actually positives) and False Positives (Customers classified as positives that are actually negatives). A value of 0.5 makes the model's predictions for both classes overlap, that is, the model cannot distinguish between one class and the other, it is practically random, while the model is better able to distinguish between one

class and another, we are closer to an AUC value equal to 1. What is the difference between AUC and F1, Accuracy etc? These are point-wise metrics, that is, they look, at a threshold, if the prediction is right or not, and calculate a score, whereas the AUC looks at the ordering of the predictions regardless of a threshold. Consequently:

- A model with high AUC and low accuracy means that the model is bad, but you can find a threshold where the scores are good
- A model with low AUC and low accuracy means that the model is bad and even changing the threshold will not help
- A model with high AUC and high accuracy means that the model does a good job and there are several other thresholds that will give good results as well.
- A model with low AUC and high accuracy means that the model is good, but for several other thresholds its score would be very bad

Chapter 4

Methodology

In this chapter, we describe the datasets used in this work as well as the details of the implementation of a transductive regularizer for feature selection. We compare our proposed feature selection method against a baseline, which does not use our regularizer.

4.1 Datasets

We conducted two experiments in this work, in which each of them uses two different datasets. For both of the experiments, the machine learning algorithm used was the XGBoost [12].

For Experiment 1, the databases used come from two different hospitals. One of them from the Geriatric ward of a hospital and the other from the Neurology ward. The dataset comes from the paper *Alzheimer's Disease: Risk Factors and potentially protective measures* [58]. Both datasets have different test results used for the diagnosis of neurological diseases. The labels of the two datasets are the diseases diagnosed in the patient, namely, Mild Cognitive Impairment (MCI) and Alzheimer's. In addition to these two labels, the databases contain tests from healthy people called the control group. Overall, the Geriatrics database has 167 patients and 45 characteristics, the Neurology base has 160 patients and 122 characteristics. However, as we want to use learning transduction between the two databases, both need to have the same features, the intersection of features reduces the final set to twenty features. In order to use Mantel we need the same features on both datasets, then we take the intersection features between both, which gives us 20 features in total.

Of the 20 features presented in the dataset, six are demographic features, such as gender, schooling, age, if the person smokes or not, and if he uses alcohol. The rest of the features are lab exams, such as blood and cognition exams.

The Figure 4.1 shows the distribution of targets between Geriatric and Neurology dataset. We can see that Neurology dataset is more balanced, meanwhile the Geriatric

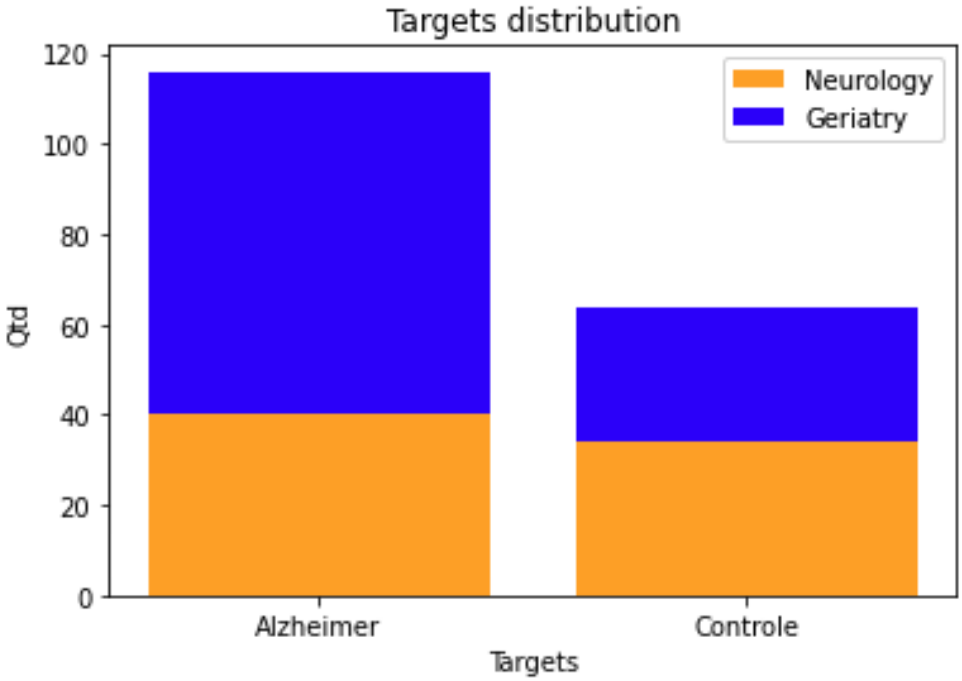

Figure 4.1: Targets distribution for Neurology and Geriatric datasets

dataset has more patients with Alzheimer's.

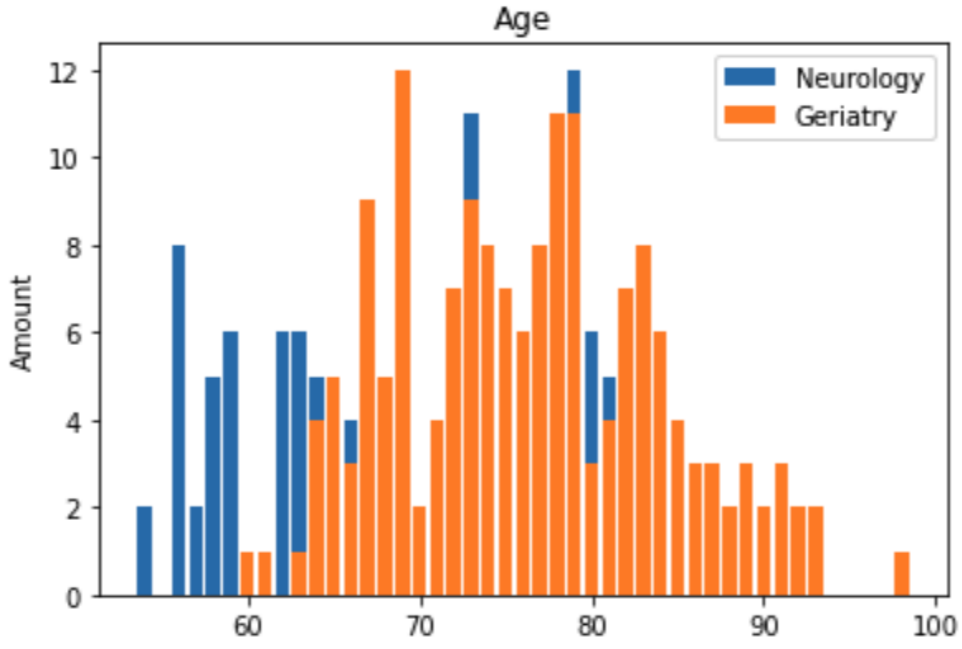

Figure 4.2: The Neurology dataset consists of younger people than the geriatric dataset

Figure 4.2 shows how the patients in the Neurology dataset are younger than the patients in the Geriatric dataset, and age is a potential feature to identify Alzheimer's disease [58]. Then, if age ends up being an important feature of the model, it could perform better for the Geriatric dataset. In fact, we also see a higher concentration of

patients with Alzheimer's in the Geriatric dataset, as shown in Figure 4.1.

Experiment 2 uses two different databases about Heart Diseases, of which one comes from the Hungarian Institute of Cardiology (Hungarian) and the other from the V.A Medical Center, Long Beach, and Cleveland Clinic Foundation (Cleveland). The datasets were extracted from the *UC Irvine Machine Learning* repository[1]. The datasets contain 76 attributes and each row of the datasets has test results from a patient and the label is whether the individual has heart disease or is healthy.

The authors of the databases are:

- 1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
- 2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
- 3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
- 4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation:Robert Detrano, M.D., Ph.D.

There are 13 intersections of features between the two datasets. Of these, only two are demographic features, gender, and age, the rest are exams, mostly related to the health of the heart.

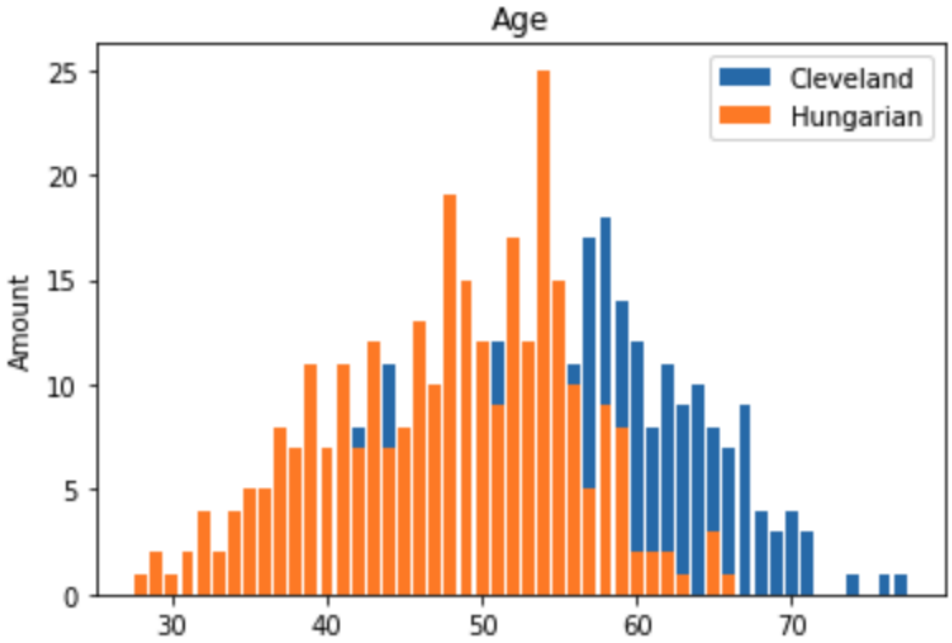

Figure 4.3: The Hungarian hospital dataset consists of younger people than the Cleveland Hospital dataset

Figure 4.3 shows that the Hungarian Hospital dataset has younger people than Cleveland, and as discussed in Experiment 1, age can be an important feature to classify a patient with Alzheimer's.

[1] https://archive.ics.uci.edu/ml/datasets/heart+disease

4.1.1 Alzheimer Dataset Features

At total, there are 20 features presented in the final dataset, from which five are demographic features and 15 are lab exams results [18].

- **sex**: Male or Female
- **schooling**: Education Level
- **age**: Patient age in years
- **smoking**: Smoke or do not smoke
- **alcoholism**: Drink Alcohol or do not drink alchool
- **lagtime-high tf+apc**: Lag Time of tissue factor activated protein c plasma factor in blood exam
- **has**: Systemic Arterial Hypertension
- **etp-low-tf-atfpi**: Endogenous thrombin potential levels of free tissue factor pathway inhibitor
- **apo-e**: Apolipoprotein E gene, protein involved in lipid metabolism
- **peak-high tf-apc**: Peak Height of Tissue factor activated protein c plasma factor in blood exam
- **pc-rq**: High sensitivity quantitative c-reactive protein blood exam
- **blood group**: Blood Type, such as O, A+, B etc
- **peak-low tf-atfpi**: Peak Low levels of free tissue factor pathway inhibitor
- **lagtime-low tf-atfpi**: Low Lag Time of tissue factor activated protein c plasma factor in blood exam
- **mini-mental**: Mini-mental status examination (MMSE)
- **lagtime-high tf-apc**: Lag Time of tissue factor activated protein c plasma factor in blood exam
- **dm**: Diabetes Mellitus
- **etp-high tf+apc**: Endogenous thrombin potential of tissue factor activated protein c plasma factor in blood exam

- **etp-high tf-apc**: Endogenous thrombin potential of tissue factor activated protein c plasma factor in blood exam
- **peak-high tf+apc**: Peak Height of Tissue factor activated protein c plasma factor in blood exam

4.1.2 Heart Diseases Dataset Features

At total, there are 13 features presented in the final dataset, from which two are demographic features and 11 are lab exams results [18].

- **sex**: Male or Female
- **age**: Patient age in years
- **cp**: Chest pain type
- **trestbps**: Resting blood pressure
- **chol**: Serum cholesterol
- **fbs**: Fasting blood sugar
- **restecg**: Resting electrocardiographic
- **thalach**: Maximum heart rate
- **exang**: Exercise induced angina
- **oldpeak**: ST depression induced by exercise relative to rest
- **slope**: Slope of peak exercise ST segment
- **ca**: Number of major vessels
- **thai**: Heart rate defect

4.2 Motivation and Use Cases

The main motivation of this work is that model generalization is a very difficult problem and with transduction, we can simplify more complex problems. In some cases,

what is sought is just the solution in a specific set of data, without the need to find general rules [62]. One key aspect of the use case is that the target dataset is known, but there are no labels available, so we cannot train a supervised model using it as input. In the case of our experiments, we have the labels available for the target datasets, but we do not use it in the transductive learning, however, we use them to validate the final performance of the model in the target datasets, which is presented in the Results section.

Therefore, in some cases, transduction is preferable to induction, once it aims to solve a smaller problem, instead of creating a general rule to a phenomenon [26], for example: Suppose two hospitals, A and B. Hospital A has developed a model capable of diagnosing Alzheimer's patients from its neurological ward. For this, data from all patients who underwent examinations at Hospital A were used. Given the success of the model, hospital A decides to market the model to Hospital B, which treats Alzheimer's patients in the geriatric ward. However, the results were not so satisfactory. An alternative would be to retrain the model with data from hospital B, but this hospital does not have the labels in its database.

In this example, the transduction technique is preferable, as it is not necessary to find a general solution for all possible hospitals, the main objective is to have a good model only for hospital B. Furthermore, the transductive feature selection technique does not need labels to generate a new model for Hospital B.

Therefore, with transduction, a hospital may be able to develop a model for the diagnosis of Alzheimer's, without worrying about the truth, only about the usefulness. As a result, more complex problems may have utility-focused models being developed, while generalization is not yet possible.

4.3 Our Proposed Transductive Feature Selection

Our hypothesis is if a feature is important in the source data and is also important in the target data, then it must be able to contribute to the model in the target data as well. SHAP is used to measure the importance of features at the source and target, whilst Mantel, which is the correlation value r is used to calculate the similarity of the importance ranking of features between the source and target. Thus, even without the target labels, the idea is to have effective features at the source that are important to the target.

As shown in Figure 4.4 the difference between our proposal and a baseline, which follows an inductive strategy, relies on how we evaluate the models. For the inductive feature selection, each feature set $\boldsymbol{f}$ is evaluated with cross-validation, which trains a

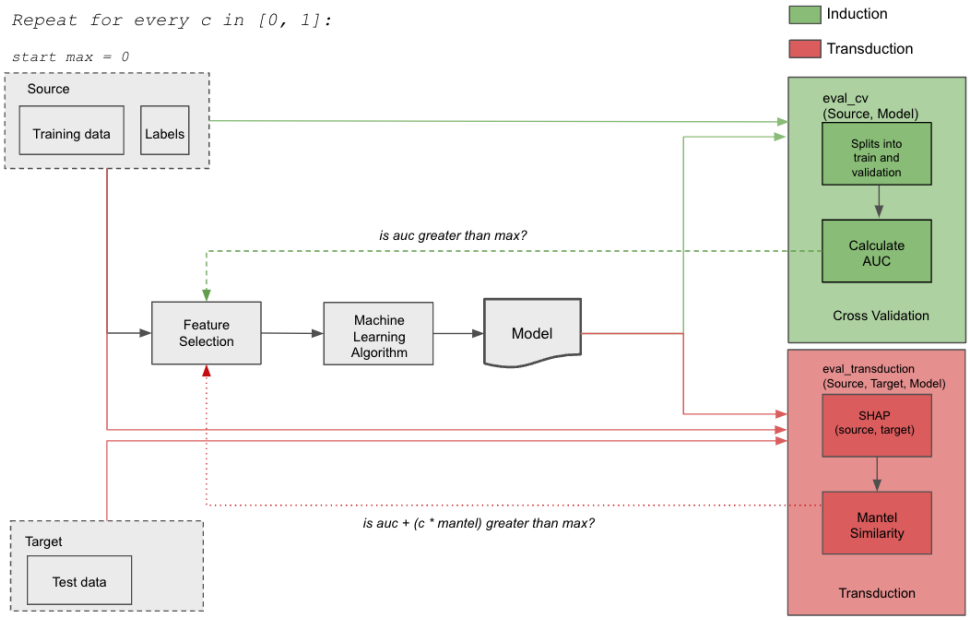

Figure 4.4: Methodology flowchart where gray arrows and rectangles are part of both methods, the green arrows and rectangles are exclusive to Induction, while the red ones are exclusive to Transduction

model using $\boldsymbol{f}$ with a sample of the source data and validates it on a different sample from the source data as well. On the other hand, for the transductive strategy, each feature set $\boldsymbol{f}$ is evaluated within a transductive evaluator, which trains a model with the whole source dataset, the given model scores both source and target datasets, and their respective feature importance is calculated using SHAP. The Mantel similarity between the source dataset's SHAP and the target dataset's SHAP measures how similar the most important features are in each dataset. Therefore we can use this measurement to infer that the features which are more cohesive in both datasets should be selected to be used for the target dataset.

An overview of our methodology is presented in Figure 4.1, No matter the feature selection strategy, all of them require a source dataset, which will be used to train the model, a machine learning algorithm, and a feature selection method that basically chooses the features that optimize a given metric. What changes from each strategy is the metric we want to optimize and how we calculate this metric. The green colored lines represent the steps to optimize the feature selection algorithm with an inductive approach, which will select the features that make the model perform with the highest AUC. On the other hand, the red colored lines represent the transductive approach, which optimizes a different metric that is not calculated using only auc, but also the matrix similarity between the feature importances matrix from the source and target dataset. The matrix similarity weight in the final metric is controlled by a regularizer denoted as $\boldsymbol{c}$. The metric optimized in the transductive approach is given by the equation:

$$auc + (c * mantel) \tag{4.1}$$

The rationale behind this metric is that we want to choose the features with high performance on the source dataset because it is the only dataset with labels where we can measure the AUC. However, we want to select features that have good ranking importance for the target dataset. In the end, we want good predictive features in the target dataset, that achieved a high performance in the source dataset.

Even though summing the performance metric (AUC) and the feature importance with matrix similarity (Mantel Test) gives us a measure, we cannot say that features with high matrix similarity values will also have a good performance, the best estimate we have is the performance in the source dataset. Then, in order to prevent low-performance features to be selected, we create a regularizer $\boldsymbol{c}$ that weights the impact of the matrix similarity, and it varies within an interval of $[0, 1]$. In our experiments, we tested different values of $\boldsymbol{c}$ and we found out that usually, the best results have a c larger than 0, but it does need to be close to 1, and in our experiments, we found a case where the best model for the target dataset had a $\boldsymbol{c}$, $= 0.2$ only (Table ??).

In general, the best models do not change much if we keep increasing $\boldsymbol{c}$, as it will be discussed in the Results section, we see that the top ranking features keep appearing even if we increase $\boldsymbol{c}$, only the lower ranked features change, therefore the final performance on the target dataset does not tend to change.

4.4 Our Implementation

Next, we briefly describe how we implemented the feature selection method with transduction.

To explain our methodology we will compare two feature selection approaches:

- **Baseline:** Common way of optimizing the selection of features seeking generalization, through the Area under the curve (AUC) metric
- **Our proposal:** Using transductive feature selection, by combining AUC and Mantel similarity together

For our transductive proposal, it is necessary to use two different data sets, a source dataset A, which will be used in training, and another target set B, which is the specific test dataset that we want to transduce learning from A. It is important to emphasize how this methodology can be applied to any existing feature selection algorithm, as an example, an algorithm will be presented that tests all possible combinations of features and chooses the set that optimizes the chosen metric.

```
1: for f ∈ features do
2:     auc ← eval_cv(A, y, f)
3:     if auc > max_auc then
4:         max_auc ← auc
5:         best_features ← f
6:     end if
7: end for
```

Algoritmo 4.1: Baseline - Optimization with only AUC metric
Variables: y is the label for A

4.4.1 Baseline Approach

In this approach, the feature selection algorithm seeks to find the feature set with the highest AUC, for this, it loops over all possible feature sets and chooses the set with the highest AUC value in cross-validation. For this method only the source A data set is used as input, the pseudo-code from Algorithm 4.1 illustrates the method.

In the algorithm below, f represents the feature set being evaluated, the *eval cv* function calculates the AUC of the feature set in the source domain, the *max auc* stores the highest AUC value found and the *best features* variable stores the respective highest AUC feature set. In the end, the baseline method returns the feature set that has the highest AUC in the source domain.

4.4.2 Proposed Approach

The proposed solution optimizes feature selection by AUC and also by Mantel similarity, in which the impact of the Mantel value on the decision is regularized by a variable c, which varies within an interval of $[0, 1]$. The pseudo-code from Algorithm 4.2 illustrates the method.

In Algorithm 4.2, c represents the regularizer value used in the interval of $[0, 1]$ with steps of 0.1 and f represents the feature set being evaluated, the *eval cv* function calculates the AUC of the feature set in the source domain, and the *eval transduction* function calculates the similarity between the SHAP importance matrices of the feature set f in both source and target domains, represented by A and B respectively, whereas yA and yB represents source and target labels respectively. The R variable calculates the equation to be optimized, which sums the AUC of the source with the mantel similarity multiplied by the c regularizer. The *max* variable stores the highest R-value found and

```
1: for c ∈ [0, 1] do
2:   for f ∈ features do
3:     auc ← eval_cv(A, y, f)
4:     mantel ← eval_transduction(A, yA, B, yB, f)
5:     R ← auc + (c ∗ mantel)
6:     if R > max then
7:       max ← R
8:       best_features ← f
9:     end if
10:  end for
11: end for
```

Algoritmo 4.2: Proposal - Optimization with AUC and Mantel
Variables: yA is the labels for A and yB the labels for B

the *best features* variable stores the respective highest R feature set. In the end, the proposed method returns the feature set that has the highest AUC + (c * mantel).

One way to improve the results was to use only a percentage of the best features of the source and target datasets through the **topFeatures** method (Algorithm 4.4), this percentage can be controlled by the *factor* parameter. Algorithm 4.3 shows Algorithm 4.2 adapted to use the **topFeatures** method.

The **topFeatures** method returns a dataset containing only a percentage defined as *factor* from the best features. By using only a percentage of the best features, we compare the similarity only between the features that actually impact the model the most. Because, without this, the solution could select a set of similar features only among those with a low position in the ranking of importance, or fail to choose a good set of features because some less important features have a low correlation between source and target distance matrices. The pseudo-code from Algorithm 4.3 illustrates the method.

For example, in the Figure 4.5, all tables are distance matrices, in the left part of the image we consider all the data from both matrices and the correlation is 0.74, in the right part of the image we consider only a *factor*, which is a fraction of the most important features, of the matrices and we achieve a higher correlation.

```
for c ∈ [0, 1] do
    for f ∈ features do
        A_top ← topFeatures(A, factor)
        B_top ← topFeatures(B, factor)
        auc ← eval_cv(A_top, y, f)
        mantel ← eval_transduction(A_top, yA,
        B_top, yB, f)
        R ← auc + (c ∗ mantel)
        if R > max then
            max ← R
            best_features ← f
        end if
    end for
end for
```

Algoritmo 4.3: Proposal Using Only the Top Features
Variables: y is label for A

```
n_features ← ceil(X.features.size ∗ factor)
top_features ← shap_importance.head(n_features)
```

Algoritmo 4.4: TopFeatures method

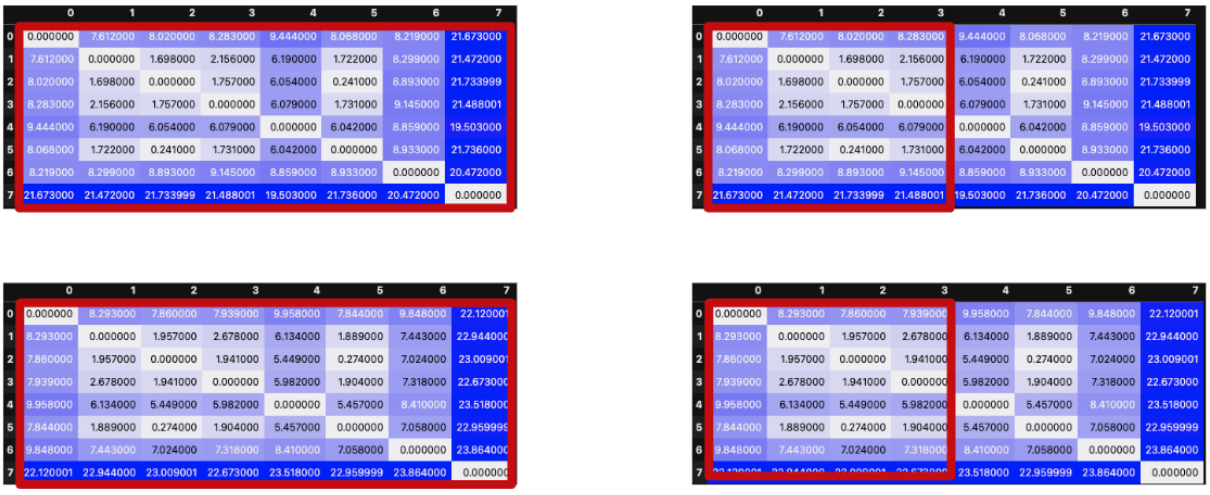

Figure 4.5: Distance matrices of feature importances. In the left we consider the whole matrices and in the right we consider only a factor of the matrices. When we consider the factor of the most important features we achieve a better correlation

Chapter 5

Results

In this chapter, we describe the results of the experiments designed to evaluate the transductive feature selection. In Section *Performance x Matrix Similarity*, we discuss the trade-offs between performance (AUC) and matrix similarity (Mantel). Then in Section *Transduction Performance*, we show the main results of this work, comparing our approach against a baseline in two scenarios and four different datasets. Finally, we explain the impacts of the regularizer in the final performance of the models in *Section Explaining regularizer impact on results.*

5.1 Performance x Matrix Similarity

In this section, we discuss the trade-off between performance in the source dataset and the similarity between the feature importance matrix from source and target datasets. The equation (4.1) shows that we choose the feature sets that optimize both performances in the source dataset and matrix similarity. Then, we could have different combinations of performance and similarity, such as features that have a good similarity between datasets but poor performance in the source dataset.

Figure 5.1 plots the AUC x Mantel distance for each model generated in Experiment 1 with data trained on Neurology and Geriatric using Alzheimer's datasets and shows how each model obtained from a different combination of features can behave. In general, we would like to have features with a good performance on the source dataset and a high similarity, as seen in Figure 5.1, we can spot some clusters of models with high performance and high similarity, in fact, if we split the plots into quadrants we would be able to tell the trade-off of each model. The models in the upper-left corner have less performance, but high matrix similarity, and the upper-right corner are the candidates for best models because they present high performance and high similarity. The lower-left corner is blank in Figure 5.1, and they would be the models with the lowest performance and lowest mantel similarity. Lastly, in the lower-right corner are the models with low

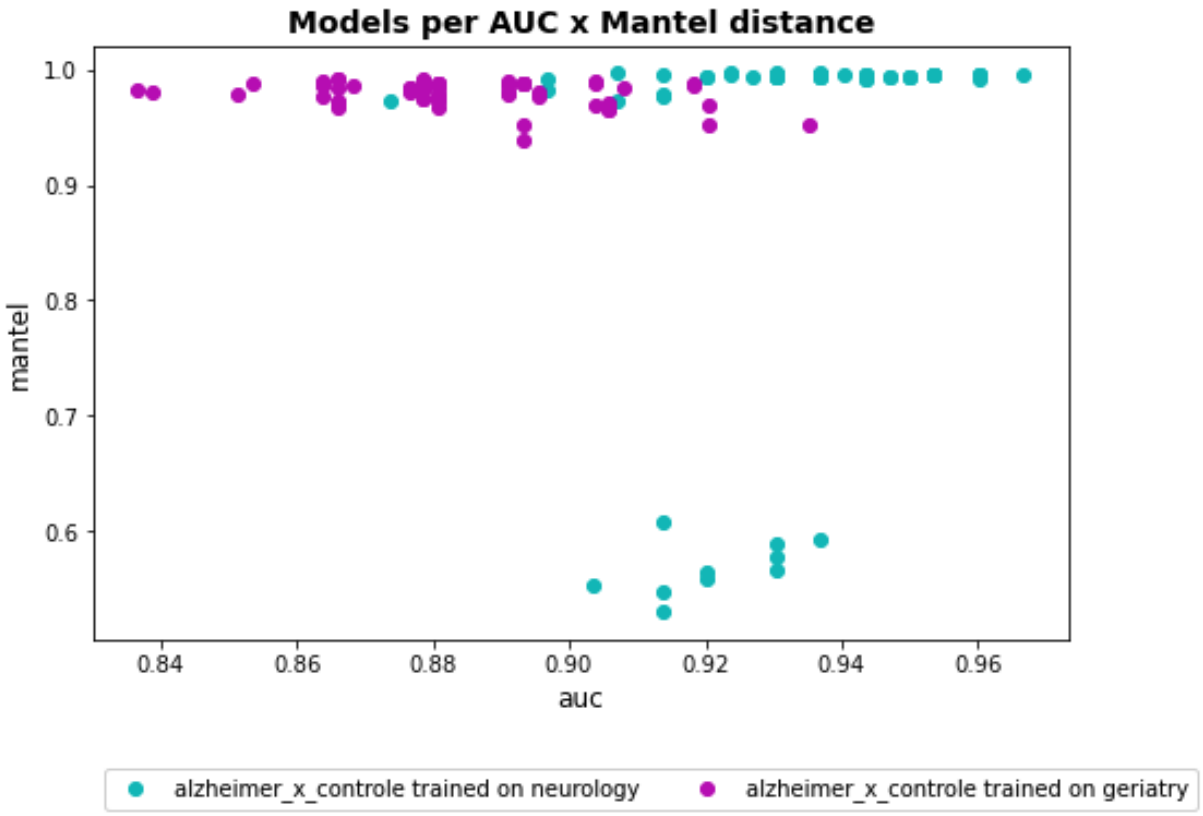

Figure 5.1: Performance x Matrix Similarity scatter plot for Experiment 1. The blue dots represent models trained with neurology data and the purple dots represent the models trained with geriatric data

similarity but high performance. In general what we see is a bigger density of models with higher similarity, which indicates that most features are able to contribute to a classifier in both source and target dataset.

5.2 Benchmark For Algorithms Used in Our Approach

In both experiments we use the XGBoost Algorithm [12], which is a gradient boosting algorithm that is considered to be one of the state-of-the-art algorithms other than Neural Networks [23], but since the datasets used in this work are very undersampled, Neural Networks have a higher chance of overfitting [20], then we compared XGBoost with Random Forest and XGBoost yield better models. On average XGBoost had an AUC of 0.91 against a 0.84 for Random Forest, despite that, XGBoost has a special function in SHAP which makes it run faster, as SHAP is a greedy algorithm it takes a long time to

compute many combinations of features, and calculates its SHAP values.

XGBoost belongs to the family of boosting methods, meanwhile, Random Forest belongs to the bagging methods family. The objective of a boosting algorithm is to minimize the loss function by using weak learners and, usually, gradient descent. The idea is that many weak learners together can form a strong learner, in the case of XGBoost the weak learner used a simple decision tree, that altogether with some optimized and efficient implementation of gradient boosting and regularization techniques, make of XGBoost a very fast and scalable choice of machine learning algorithm.

On the other hand, Random Forest is a bagging algorithm, which is an ensemble method formed by multiple decision trees that average decisions to make a final prediction. Differently from bagging methods, boosting methods are added sequentially to the ensemble, which allows the model to correct itself iteratively, changing the weights of the data points, which diminishes the chances of a random prediction, which is more likely to happen in a random forest, since we cannot guarantee that some of its trees suffer from class imbalance and overfitting for example [52].

For the similarity dimension, we use the Mantel test for both experiments, we also tested other matrix similarities algorithms, such as the Pearson and Euclidian distance. In the sample tests we did, Mantel scored the highest values of AUC when applied in conjunction with XGBoost, achieving an AUC of 0.64 on average, against 0.55 for Pearson correlation and 0.5 for Euclidian distance. The cons of the Mantel approach are that the matrices must have the same dimension, which forces us to lose some data, as we have to remove features that do not belong to both datasets, but most importantly we need to also remove some patients from the dataset, as we need to ensure that both datasets have the same number of patients too.

5.3 Transduction Performance

In this section, we use AUC to calculate the performance of the chosen model for the baseline approach and the chosen model for the transductive regularizer approach on the target dataset. The idea is to validate the final performance of the chosen model at the end of the feature selection pipeline for each of the approaches. Even though our approach does not require the target dataset to contain labels, the target datasets used in the experiment contain labels, which we do not use for the feature selection, but use for the final model validation.

For the first experiment, we have two different datasets, one from the Neurology ward of a hospital and the other from the Geriatric ward, both containing exam results and

a label saying if the person has Alzheimer's or not. Thus we conducted two experiments, one using Neurology data as the source dataset and the Geriatry data as the target dataset, the other experiment is the other way around with Geriatry data used as the source and Neurology as the target. For each of those experiments, we compared the baseline approach, which does not use transduction, against our transductive feature selection in terms of AUC in the target dataset. In Table ?? we have the results for the experiments, in which the results for the transductive method our split by different c values, from 0 to 1, varying by 0.1.

The results in the first experiment showed that the transductive method was able to outperform the baseline for at least one value of c. In the experiment training with the Neurology dataset, we have our approach with better performance for every $c >= 0.2$. On the other hand, when we trained using the Geriatric dataset and validated on the Neurology dataset, our approach was only better than the baseline for $c = 0.2$

For the second experiment, we also have two different datasets, one from the Hungarian Institute of Cardiology (Hungarian) and the other from the V.A Medical Center, Long Beach, and Cleveland Foundation (Cleveland), both containing exams results and a label saying if the person has heart disease or not. Thus we also conducted two experiments, one using the Hungarian dataset as the source and the Cleveland dataset as the target, for the other experiment we switched roles. As done in Experiment 1, we compared the baseline approach against our approach by calculating the AUC in the target dataset for each experiment. In Table ?? we have the results of the experiments.

The results in the second experiment showed that our approach also outperformed the baseline performance for every $c >= 0.2$ in both experiments.

In our experiments, we varied the value of c to measure how much the impact of transduction would affect the final model performance on the target dataset. However, the c variable is not the only variable we use in our approach. Another parameter we can vary is the *factor*, which is the proportion of the top features from SHAP that will be considered to calculate the Mantel similarity. In other words, the *factor* controls how much we want to measure similarity between the most important features only, or between all features. The idea behind it is that we do not want to penalize features that show high similarity between the most important features in the source and target dataset but on the other hand show low similarity for bottom-ranked features, which are features that probably do not contribute much individually in the final output. in Table ?? we have the results from Experiment 2 varying by a factor as well, and we see that the performance changes by *factor* and most of the best results are concentrated in the lower factors, which use fewer features to calculate similarity.

Source	Target	Baseline	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
Neurologia	Geriatria	0.847	0.862	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889
Geriatria	Neurologia	0.804	0.825	0.833	0.758	0.758	0.758	0.758	0.758	0.758	0.758	0.758

Table 5.1: Experiment 1 AUC result by c regularizer

Source	Target	Baseline	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
Cleveland	Hungarian	0.598	0.715	0.764	0.764	0.764	0.764	0.764	0.764	0.764	0.764	0.764
Hungarian	Cleveland	0.754	0.754	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756

Table 5.2: Experiment 2 AUC result by c regularizer

Source	Target	Factor	Baseline	c=0.1	c=0.2	c=0.3	c=0.4	c=0.5	c=0.6	c=0.7	c=0.8	c=0.9	c=1.0
Cleveland	Hungarian	0.1	0.598	0.590	0.637	0.637	0.712	0.712	0.712	0.712	0.674	0.674	0.674
Hungarian	Cleveland	0.1	0.754	0.754	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756
Cleveland	Hungarian	0.2	0.598	0.590	0.637	0.637	0.712	0.712	0.712	0.712	0.674	0.674	0.674
Hungarian	Cleveland	0.2	0.754	0.754	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756
Cleveland	Hungarian	0.3	0.598	0.590	0.637	0.637	0.712	0.712	0.712	0.712	0.674	0.674	0.674
Hungarian	Cleveland	0.3	0.754	0.754	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756
Cleveland	Hungarian	0.4	0.598	0.715	0.764	0.764	0.764	0.764	0.764	0.764	0.764	0.764	0.764
Hungarian	Cleveland	0.4	0.754	0.737	0.737	0.737	0.737	0.737	0.737	0.737	0.737	0.737	0.737
Cleveland	Hungarian	0.5	0.598	0.715	0.764	0.764	0.764	0.764	0.764	0.764	0.764	0.764	0.764
Hungarian	Cleveland	0.5	0.754	0.712	0.712	0.712	0.724	0.724	0.724	0.724	0.724	0.724	0.724
Cleveland	Hungarian	0.6	0.598	0.715	0.715	0.715	0.715	0.715	0.715	0.715	0.715	0.715	0.715
Hungarian	Cleveland	0.6	0.754	0.706	0.706	0.706	0.706	0.737	0.737	0.737	0.737	0.737	0.724
Cleveland	Hungarian	0.7	0.598	0.715	0.717	0.674	0.687	0.687	0.687	0.687	0.687	0.687	0.687
Hungarian	Cleveland	0.7	0.754	0.698	0.698	0.734	0.734	0.734	0.734	0.734	0.734	0.764	0.734
Cleveland	Hungarian	0.8	0.598	0.601	0.716	0.716	0.716	0.716	0.716	0.716	0.716	0.662	0.662
Hungarian	Cleveland	0.8	0.754	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756
Cleveland	Hungarian	0.9	0.598	0.590	0.637	0.637	0.712	0.712	0.712	0.712	0.674	0.674	0.674
Hungarian	Cleveland	0.9	0.754	0.754	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756
Cleveland	Hungarian	1	0.598	0.590	0.637	0.637	0.712	0.712	0.712	0.712	0.674	0.674	0.674
Hungarian	Cleveland	1	0.754	0.754	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756

Table 5.3: Experiment 2 AUC result by c regularizer per factor

5.4 Explaining Regularizer Impact on Results

The regularizer c is a parameter that weights the impact of the matrix similarity in the final decision of our optimization function (4.1). So what is the expected behavior of increasing or decreasing c? In general, if we increase c we intend to increase the transduction, meanwhile, if we decrease it we are closer to what the baseline does, which is basically picking the features with the highest AUC in the source dataset. What we see in Table ?? and ?? is that for most experiments, increasing the c parameter value, increased AUC as well, because we were selecting features with the highest matrix similarity. However, there is one experiment where only one specific value of c was better than the baseline, which means that by increasing transduction we were losing

performance. This is why we developed the regularizer, there is no guarantee that features with higher matrix similarity will perform better in the target dataset. Then, one should use c as an experimental parameter and choose the one that brings the best results.

Apo-e feature discussion As stated by [3] approximately 70 percent of the likelihood of developing Alzheimer's disease can be attributed to genetic factors. In the case of early symptoms of Alzheimer's, mutations in the APP, PSEN1, and PSEN2 genes are often responsible. On the other hand, a late case is primarily linked to a variation in the Apo-e gene (apolipoprotein E gene) [9, 10]. Apo-e is a protein that is a know risk factor associated with Alzheimer, it just appears after the regularizer c is bigger than 0 (figure 5.3). Apo-e is associated with age as well, as it is more common in elderly people, the neurology source dataset has younger people, so Apo-e tends to be less relevant, but for geriatric, which contains more elderly people, it is more relevant. Transduction is able to select this feature because it is very important for the target.

Figures 5.2 up to 5.11 shows the ranking of the selected features by each c parameter value in Experiment 1 when using Neurology as the source dataset and Geriatry as the target dataset. Features are color-coded according to their values (highest: red, lowest: blue). For instance, low values of mini-mental are associated with positive SHAP values. We see that the feature ranking and the SHAP values change in $c = 0.2$ and then stay the same. This is exactly the same behavior we see in Table **??**, where the performance stays constant for every $c >= 0.2$. Then, the reason why the performance is constant after some c is that as the similarity weight increases, features with higher matrix similarity are selected.

Therefore we see that for most of our experiments, there is a high correlation between the performance in the target dataset with the feature importance similarity that the selected feature set has between both source and target datasets. Meanwhile, the regularizer is useful to control scenarios where higher similarity can decrease auc, then in experiments, we can select the best parameters to optimize the model.

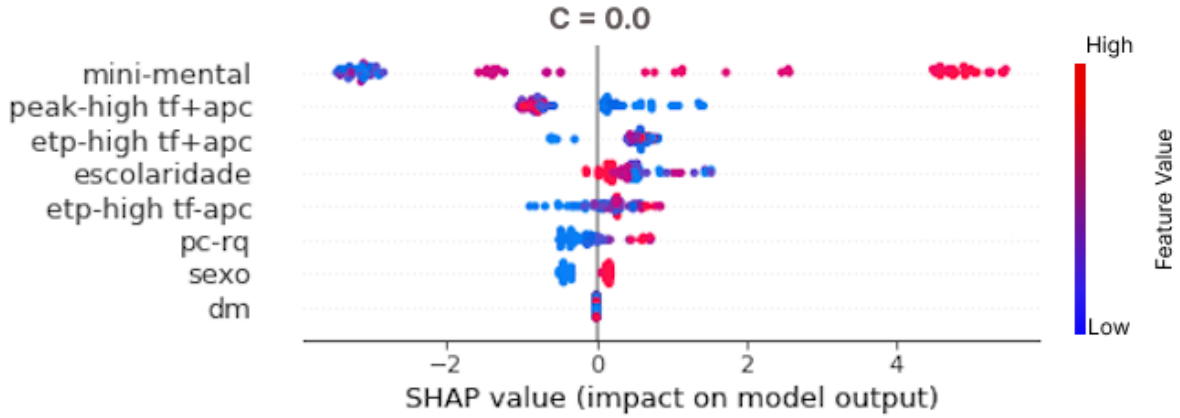

Figure 5.2: SHAP values for features selected with regularizer c = 0. Features are color-coded according to their values (highest: red, lowest: blue)

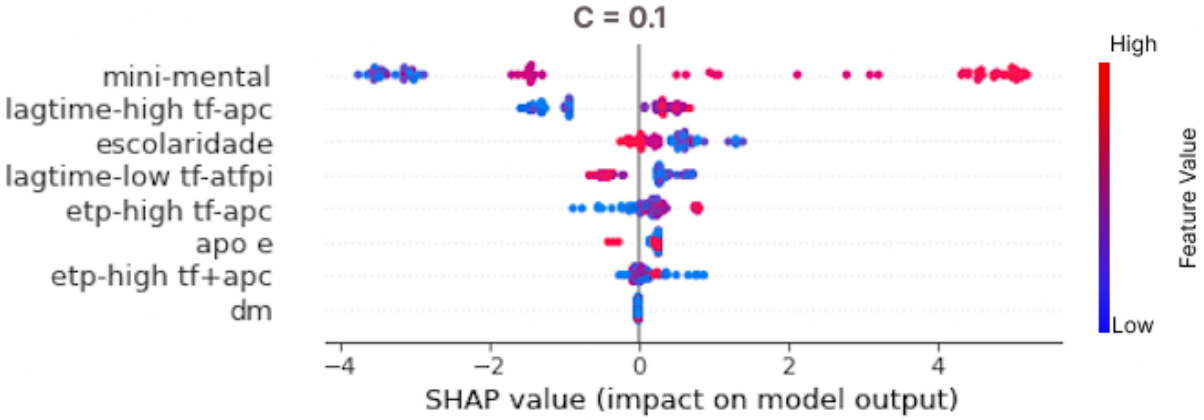

Figure 5.3: SHAP values for features selected with regularizer c = 0.1. Features are color-coded according to their values (highest: red, lowest: blue)

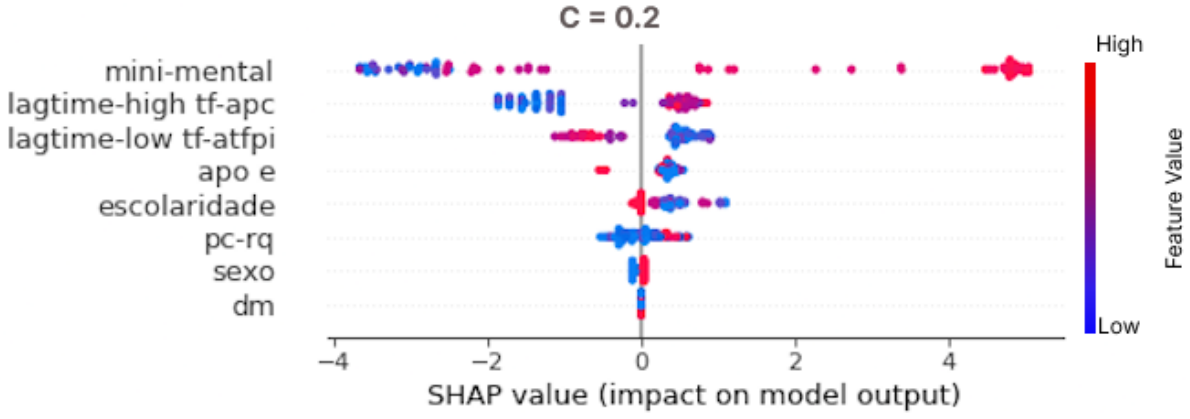

Figure 5.4: SHAP values for features selected with regularizer c = 0.2. Features are color-coded according to their values (highest: red, lowest: blue)

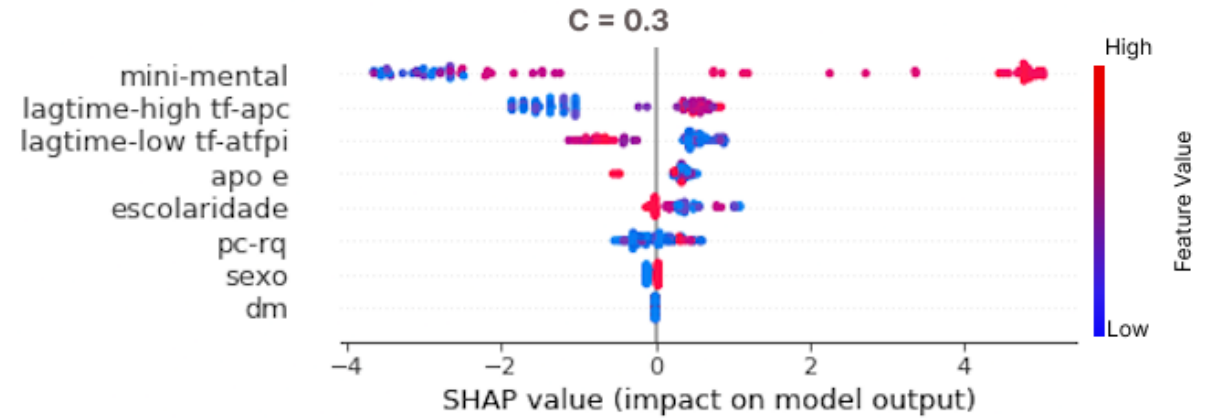

Figure 5.5: SHAP values for features selected with regularizer c = 0.3. Features are color-coded according to their values (highest: red, lowest: blue)

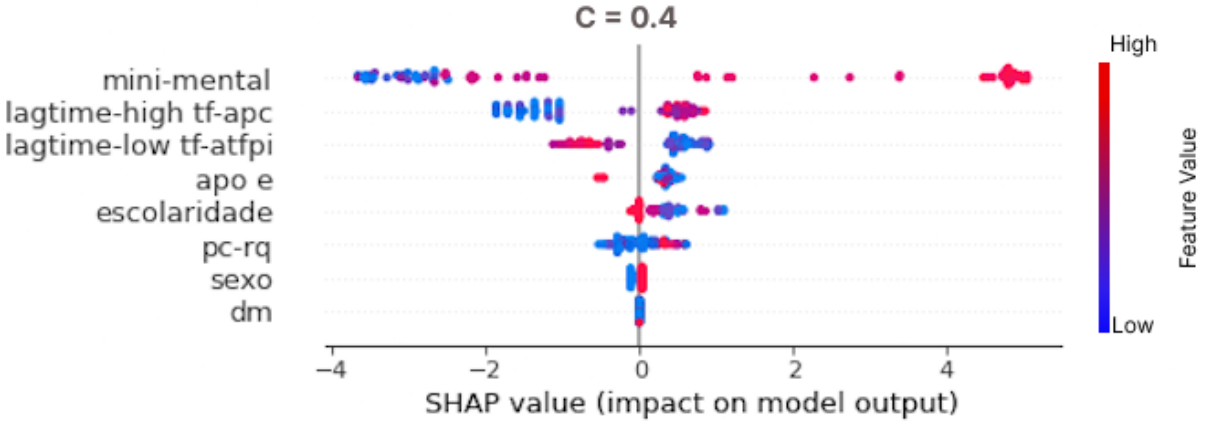

Figure 5.6: SHAP values for features selected with regularizer c = 0.4. Features are color-coded according to their values (highest: red, lowest: blue)

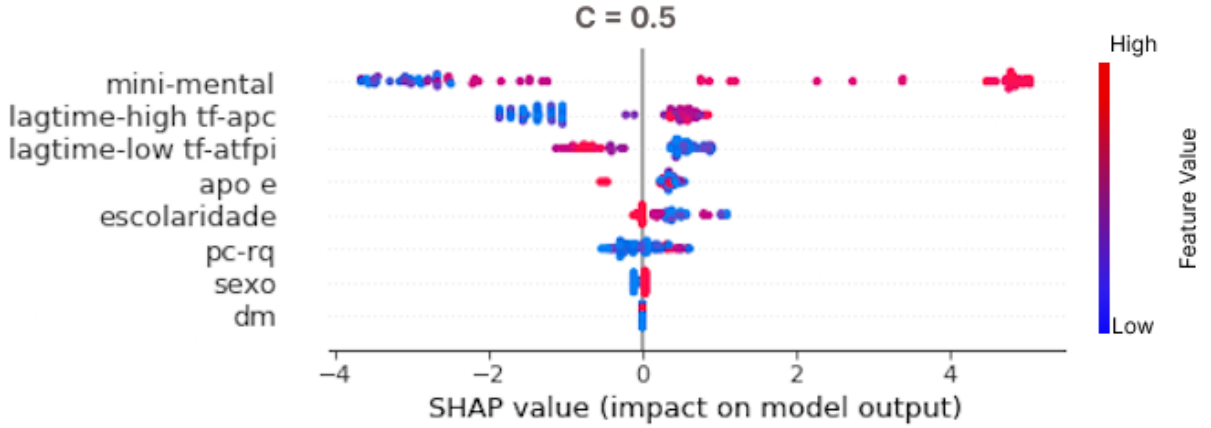

Figure 5.7: SHAP values for features selected with regularizer c = 0.5. Features are color-coded according to their values (highest: red, lowest: blue)

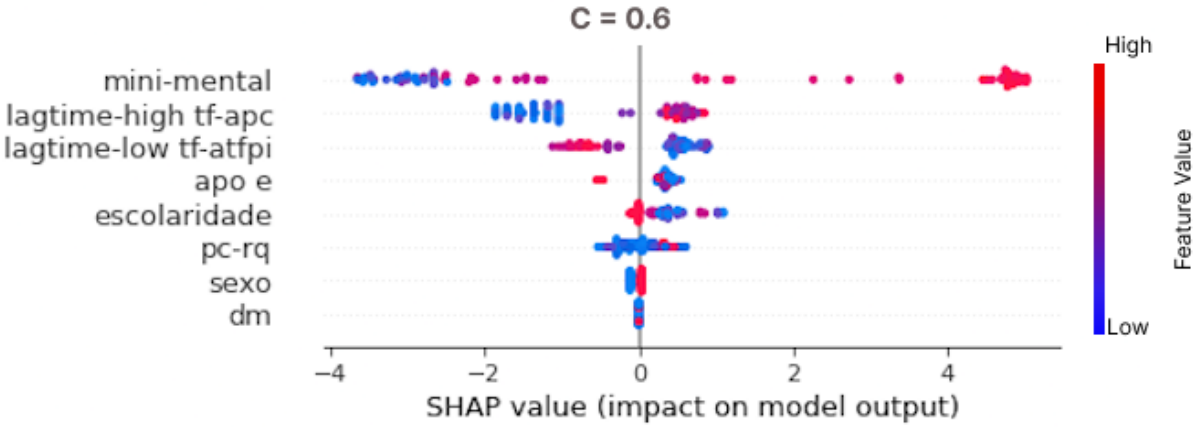

Figure 5.8: SHAP values for features selected with regularizer c = 0.6. Features are color-coded according to their values (highest: red, lowest: blue)

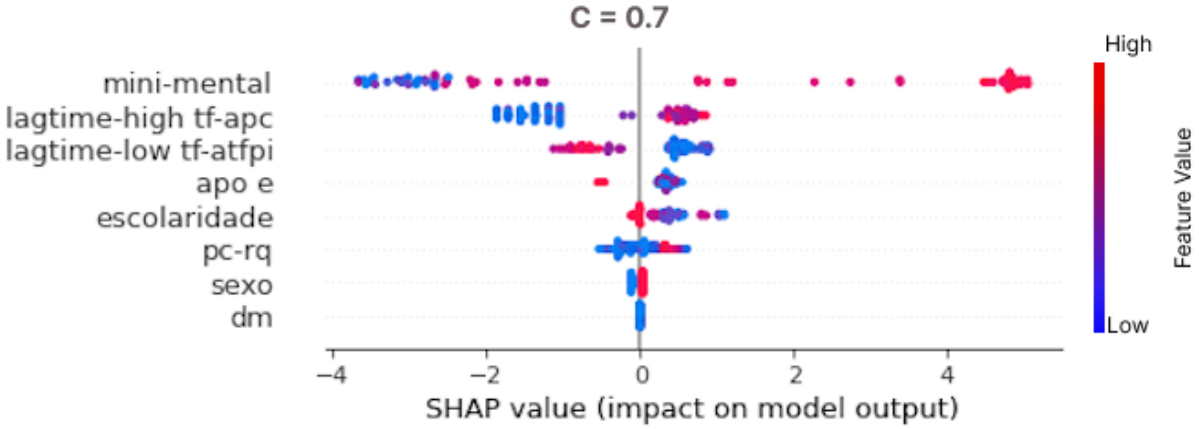

Figure 5.9: SHAP values for features selected with regularizer c = 0.7. Features are color-coded according to their values (highest: red, lowest: blue)

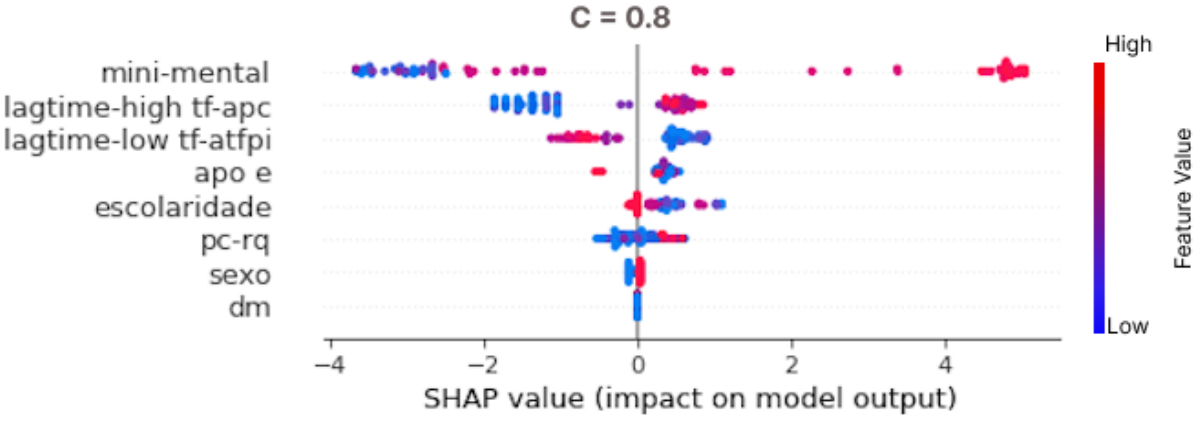

Figure 5.10: SHAP values for features selected with regularizer c = 0.8. Features are color-coded according to their values (highest: red, lowest: blue)

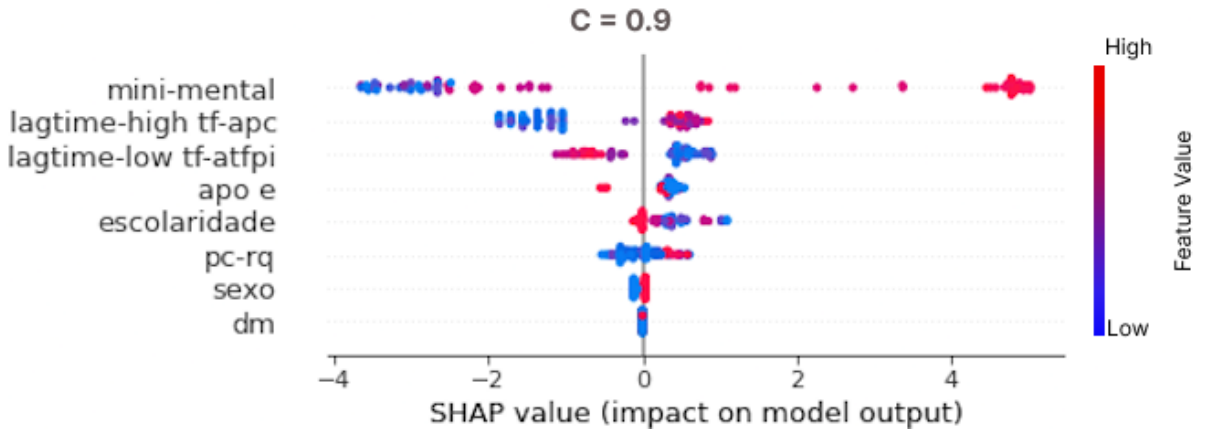

Figure 5.11: SHAP values for features selected with regularizer c = 0.9. Features are color-coded according to their values (highest: red, lowest: blue)

Chapter 6

Concluding Discussion

In this work, we discussed a new approach for feature selection in a scenario where is desired to apply a model in unlabeled target datasets, and for this all we have is source dataset with labels and the target and source dataset are from different data domains. In our approach we train N models, where N represents every possible combination of feature sets, and then we calculate the SHAP values for each feature set in the source and target dataset. Moreover we calculate the matrix similarity between the source and target datasets SHAP values and rank them with the equation 4.1 to choose the top feature set, this feature set is chosen to be used for the model to be used in the target dataset.

The first problem addressed in this research is the fact that induction learning is a hard problem and most of the times we do not seek a model to generalize, we just need a model to work specifically for one data domain or target set. Since generalization is hard, it is very hard for a model trained on a source dataset to work properly in a target dataset without re-training. In a case where there are no labels available for the target, we could use transduction to select a new set of features, and in our tests the transduction feature selection we proposed achieved better results than an induction technique ??.

Next, we detailed how SHAP and Mantel test could be used to create the equation 4.1 that uses a transduction regularizer to control how much we want our transduction to impact the final result. This approach is interesting because it allows us to control the final result and test many different values for the regularizer.

We provided many contributions that are relevant to the field. First, we presented a new feature selection approach. This method seeks the transduction of the model and not the generalization. With transduction, it was possible to find better models than with generalist strategies. Second, we proposed a regularizer to control transduction inside a system, by combining feature importance and Mantel similarity. We hope our efforts can become a baseline for other solutions using transductive strategies.

Finally, our findings suggest that transduction approaches can be successful in feature selection tasks and more research should be done. For future work, we aim to test the method in new datasets and in conjunction with other feature selection techniques and assess this method for large datasets and verify its scalability.

References

[1] Hisham Al-Mubaid and Syed A. Umair. A new text categorization technique using distributional clustering and learning logic. *IEEE Trans. Knowl. Data Eng.*, 18(9):1156–1165, 2006.

[2] Andrew O. Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. In *Workshops Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, pages 77–82. IEEE Computer Society, 2007.

[3] Clive Ballard, Serge Gauthier, Anne Corbett, Carol Brayne, Dag Aarsland, and Emma Jones. Alzheimer's disease. *The Lancet*, 377(9770):1019 – 1031, March 2011. Copyright © 2011 Elsevier Ltd. All rights reserved.

[4] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In John Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics, 2007.

[5] John Blitzer, Ryan T. McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In Dan Jurafsky and Éric Gaussier, editors, *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 120–128. ACL, 2006.

[6] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

[7] Gavin Brown. *Diversity in neural network ensembles*. PhD thesis, University of Birmingham, UK, 2004.

[8] Dariusz Brzezinski. Fibonacci and k-subsecting recursive feature elimination. *CoRR*, abs/2007.14920, 2020.

[9] Rita Cacace, Kristel Sleegers, and Christine Van Broeckhoven. Molecular genetics of early-onset alzheimer's disease revisited. *Alzheimer's Dementia*, 12(6):733–748, 2016.

[10] Miguel Calero, Alberto Gómez-Ramos, Olga Calero, Eduardo Soriano, Jesús Avila, and Miguel Medina. Additional mechanisms conferring genetic susceptibility to alzheimer's disease. *Frontiers in cellular neuroscience*, 9:138, 04 2015.

[11] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers Electrical Engineering*, 40(1):16–28, 2014. 40th-year commemorative issue.

[12] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM, 2016.

[13] Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27, 1967.

[14] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. In Pavel Berkhin, Rich Caruana, and Xindong Wu, editors, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pages 210–219. ACM, 2007.

[15] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classifiers for text classification. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 540–545. AAAI Press, 2007.

[16] Debasis Das, Somnath Bhattacharya, and Bijan Sarkar. Material selection in engineering design based on nearest neighbor search under uncertainty: a spatial approach by harmonizing the euclidean and taxicab geometry. *Artif. Intell. Eng. Des. Anal. Manuf.*, 33(3):238–246, 2019.

[17] Denny, Peter Christen, and Graham J. Williams. Analysis of cluster migrations using self-organizing maps. In Longbing Cao, Joshua Zhexue Huang, James Bailey, Yun Sing Koh, and Jun Luo, editors, *New Frontiers in Applied Data Mining*, pages 171–182, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[18] A.W.J.H. DIELIS, E. CASTOLDI, H.M.H. SPRONK, R. VAN OERLE, K. HAMULYÁK, H. TEN CATE, and J. ROSING. Coagulation factors and the protein c system as determinants of thrombin generation in a normal population. *Journal of Thrombosis and Haemostasis*, 6(1):125–131, 2008.

[19] José Alexandre F. Diniz-Filho, Thannya N. Soares, Jacqueline S. Lima, Ricardo Dobrovolski, Victor Lemes Landeiro, Mariana Pires de Campos Telles, Thiago F.

Rangel, and Luis Mauricio Bini. Mantel test in population genetics. *Genetics and Molecular Biology*, 36(4):475–485, 2013.

[20] Claudio Filipi Goncalves dos Santos and João Paulo Papa. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Comput. Surv.*, 54(10s):213:1–213:25, 2022.

[21] Jeroen Eggermont, Joost N. Kok, and Walter A. Kosters. Genetic programming for data classification: partitioning the search space. In Hisham Haddad, Andrea Omicini, Roger L. Wainwright, and Lorie M. Liebrock, editors, *Proceedings of the 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus, March 14-17, 2004*, pages 1001–1005. ACM, 2004.

[22] L Elliss-Brookes, S McPhail, A Ives, M Greenslade, J Shelton, S Hiom, and M Richards. Routes to diagnosis for cancer - determining the patient journey using multiple routine data sets. In *Br J Cancer*, 2012.

[23] Elizabeth Fernandes, Sérgio Moro, and Paulo Cortez. Data science, machine learning and big data in digital journalism: A survey of state-of-the-art, challenges and opportunities. *Expert Syst. Appl.*, 221:119795, 2023.

[24] Daniel Vidali Fryer, Inga Strümke, and Hien D. Nguyen. Shapley values for feature selection: The good, the bad, and the axioms. *CoRR*, abs/2102.10936, 2021.

[25] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Hongjun Lu, and Philip S. Yu. Text classification without negative examples revisit. *IEEE Trans. Knowl. Data Eng.*, 18(1):6–20, 2006.

[26] Alex Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. 01 1998.

[27] Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In Gregory F. Cooper and Serafín Moral, editors, *UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24-26, 1998*, pages 148–155. Morgan Kaufmann, 1998.

[28] Partha Ghosh, Takaaki Goto, and Soumya Sen. Computing skyline using taxicab geometry. In *5th International Conference on Applied Computing and Information Technology, 4th International Conference on Computational Science/Intelligence and Applied Informatics, 2nd International Conference on Big Data, Cloud Computing, Data Science & Engineering, ACIT/CSII/BCD 2017, Hamamatsu, Japan, July 9-13, 2017*, pages 7–12. IEEE, 2017.

[29] Yoav Goldberg and Graeme Hirst. *Neural Network Methods in Natural Language Processing*. Morgan amp; Claypool Publishers, 2017.

[30] Yoav Goldberg, Graeme Hirst, Yang Liu, and Meng Zhang. Neural network methods for natural language processing yoav goldberg (bar ilan university)morgan & claypool (synthesis lectures on human language technologies, edited by graeme hirst, volume 37), 2017, xxii+287 pp; paperback, ISBN 9781627052986, $74.95; ebook, ISBN 9781627052955, $59.96; doi: 10.2200/s00762ed1v01y201703hlt037. *Comput. Linguistics*, 44(1), 2018.

[31] Alex Graves. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711, 2012.

[32] Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. Learning to transduce with unbounded memory. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1828–1836, 2015.

[33] Ritam Guha, Hussain Ali Khan, Pawan Kumar Singh, Ram Sarkar, and Debotosh Bhattacharjee. CGA: a new feature selection model for visual human action recognition. *Neural Comput. Appl.*, 33(10):5267–5286, 2021.

[34] Abhay Harpale. Health change detection using temporal transductive learning. In *Machine Learning, Optimization, and Data Science: 7th International Conference, LOD 2021, Grasmere, UK, October 4–8, 2021, Revised Selected Papers, Part II*, page 178–192, Berlin, Heidelberg, 2021. Springer-Verlag.

[35] Kaizhu Huang, Haiqin Yang, Irwin King, Michael R. Lyu, and Laiwan Chan. Biased minimax probability machine for medical diagnosis. In *International Symposium on Artificial Intelligence and Mathematics, AI&M 2004, Fort Lauderdale, Florida, USA, January 4-6, 2004*, 2004.

[36] Thorsten Joachims. Transductive inference for text classification using support vector machines. In Ivan Bratko and Saso Dzeroski, editors, *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pages 200–209. Morgan Kaufmann, 1999.

[37] Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.

[38] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning, 2017.

[39] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.

[40] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997. Relevance.

[41] Evgeny Kondratovich, Igor Baskin, and Alexandre Varnek. Transductive support vector machines: Promising approach to model small and unbalanced datasets. *Molecular Informatics*, 32:261–266, 03 2013.

[42] MM Koo, R Swann, S McPhail, GA Abel, L Elliss-Brookes, GP Rubin, and G Lyratzopoulos. Presenting symptoms of cancer and stage at diagnosis: evidence from a cross-sectional, population-based study. In *Lancet Oncol*, 2020.

[43] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6402–6413, 2017.

[44] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. volume abs/1705.07874, 2017.

[45] Rafael Trindade Maia and Magnólia de Araújo Campos. Introductory chapter: Genetic variation - the source of biological diversity. In Rafael Trindade Maia and Magnólia de Araújo Campos, editors, *Genetic Variation*, chapter 1. IntechOpen, Rijeka, 2021.

[46] Nathan Mantel. The detection of disease clustering and a generalized regression approach. *Cancer research*, 27 2:209–20, 1967.

[47] Y Moriarty, J Townson, H Quinn-Scoggins, L Padgett, S Owen, S Smits, R Playle, P Dimitropoulou, B Sewell, V Kolovou, P Buckle, B Carter, A Edwards, J Hepburn, M Matthews, C Mitchell, RD Neal, M Robling, F Wood, and K Brain. P improving cancer symptom awareness and help-seeking among adults living in socioeconomically deprived communities in the uk using a facilitated health check: A protocol for the awareness and beliefs about cancer (abacus) randomised control trial. In *BMC Public Health*, 2019.

[48] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. 22(10):1345–1359, oct 2010.

[49] A. Payan and G) Montana. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. In *Arxiv*, 2015.

[50] Munir Pirmohamed, Sally James, Shaun Meakin, Chris Green, Andrew K Scott, Thomas J Walley, Keith Farrar, B Kevin Park, and Alasdair M Breckenridge. Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients. volume 329, pages 15–19. BMJ Publishing Group Ltd, 2004.

[51] Rajat Raina, Andrew Y. Ng, and Daphne Koller. Constructing informative priors using transfer learning. In William W. Cohen and Andrew W. Moore, editors, *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 713–720. ACM, 2006.

[52] Fatwa Ramdani and Muhammad Tanzil Furqon. The simplicity of xgboost algorithm versus the complexity of random forest, support vector machine, and neural networks algorithms in urban forest classification. *F1000Research*, 11:1069, 2022.

[53] Roni Ramon-Gonen and Roy Gelbard. Cluster evolution analysis: Identification and detection of similar clusters and migration patterns. volume 83, 04 2017.

[54] Jenna M. Reps, Uwe Aickelin, and Richard B. Hubbard. Refining adverse drug reaction signals by incorporating interaction variables identified using emergent pattern mining. volume 69, pages 61–70, 2016.

[55] Kanoksri Sarinnapakorn and Miroslav Kubat. Combining subclassifiers in text categorization: A dst-based solution and a case study. *IEEE Trans. Knowl. Data Eng.*, 19(12):1638–1651, 2007.

[56] Matthias Scholz, Martin Fraunholz, and Joachim Selbig. Nonlinear principal component analysis: Neural network models and applications. In Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 44–67, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[57] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[58] Alves LCV de Souza LC Borges KBG Carvalho MDG Silva MVF, Loures CMG. Alzheimer's disease: risk factors and potentially protective measures. may 2019.

[59] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. Can you trust

your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

[60] Q.0 Song, L. Zhao, X. Luo, and X Dou. Using deep learning for classification of lung nodules on computed tomography images. journal of healthcare engineering. 2017.

[61] Sandhya Tripathi, N. Hemachandra, and Prashant Trivedi. On feature interactions identified by shapley values of binary classification games. *CoRR*, abs/2001.03956, 2020.

[62] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer New York, 2006.

[63] Pengcheng Wu and Thomas G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In Carla E. Brodley, editor, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004.

[64] Tai-Yin Wu, Min-Hua Jen, Alex Bottle, Mariam Molokhia, Paul Aylin, Derek Bell, and Azeem Majeed. Ten-year trends in hospital admissions for adverse drug reactions in england 1999–2009. volume 103, pages 239–250, 2010. PMID: 20513902.

[65] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus F. M. Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael S. Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, 2008.

[66] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *Int. J. Inf. Technol. Decis. Mak.*, 5(4):597–604, 2006.

[67] Hujun Yin, David Camacho, and Peter Tiňo, editors. *Intelligent Data Engineering and Automated Learning - IDEAL 2022 - 23rd International Conference, IDEAL 2022, Manchester, UK, November 24-26, 2022, Proceedings*, volume 13756 of *Lecture Notes in Computer Science*. Springer, 2022.

[68] Na-Yung Yu, Takashi Yamauchi, Huei-Fang Yang, Yen-Lin Chen, and Ricardo Gutierrez-Osuna. Feature selection for inductive generalization. *Cognitive science*, 34:1574–93, 11 2010.