Master Thesis in Statistics and Data Mining

Supervised Classification Leveraging Refined Unlabeled Data

Andreea Bocancea

Division of Statistics
Department of Computer and Information Science
Linköping University

Supervisor
Prof. Mattias Villani

Examiner
Prof. Oleg Sysoev

Contents

Abstract

This thesis focuses on how unlabeled data can improve supervised learning classifiers in all contexts, for both scarce to abundant label situations. This is meant to address the limitations within supervised learning with regards to label availability. Extending the training set with unlabeled data can overcome issues such as selection bias, noise and insufficient data. Based on the overall data distribution and the initial set of labels, semi-supervised methods provide labels for additional data points. The semi-supervised approaches considered in this thesis belong to one of the following categories: transductive SVMs, Cluster-then-Label and graph-based techniques. Further, we evaluate the behavior of: Logistic regression, Single layer perceptron, SVM and Decision trees. By learning on the extended training set, supervised classifiers are able to generalize better. Based on the results, this thesis recommends data-processing and algorithmic solutions appropriate to real-world situations.

Acknowledgments

I would like to express my sincere gratitude to any person or institution that has contributed to the success of this thesis.

I would like to give special thanks to Andreas Meisingseth and Tom Baylis for introducing such an interesting topic and advising me in various matters.

I would also like to thank my supervisor Professor Mattias Villani for proofreading the manuscript and his valuable guidance in every stage of this study. I am grateful for the thorough comments received from my opponent Roger Karlsson.

This work would have not been completed without my boyfriend's support. Thank you for sharing your knowledge with me and showing me ways to speed up the analysis in this thesis. I especially want to thank you for your encouragements and sharing your opinions about my work.

1 Introduction

1.1 Background

Binary classification is a popular problem in machine learning and it is generally solved in a supervised manner if labels exist. In order to generalize the supervised learning results, the training data must compose a representative sample of the population. Otherwise, the resulting estimates may be highly biased and the model may assign inaccurate labels to new observations. In real-world applications, many factors can contribute to scenarios containing only a few labeled data points or sample-selection bias, both of which limit the analysis regardless of the supervised method applied. The quantity and quality of labeled data are the main challenges all supervised approaches face. With less data, and consequently more influential data points, quality is an essential prerequisite for an efficient model, because the effects of noise and outliers on performance are amplified. The effects of selection-bias, noise or insufficient data can be overcome or ameliorated by modeling the underlying structure of the data. There has been an increase in research directed towards identifying efficient approaches of expanding the dataset. Noting that data is often available but is missing labels has encouraged researchers to search for methods to exploit it during the modeling process. Using unlabeled instances has been proven to aid classifiers when the assumptions regarding the data distribution are correct [62]. Semi-supervised labeling can provide the supervised classifier a more complete picture of the data space.

Semi-supervised learning utilizes both labeled and unlabeled data in the training process. The solutions proposed by semi-supervised learning consist of techniques which originate from both supervised (e.g. Support Vector Machines) and unsupervised tasks (e.g. clustering). These can be categorized as *inductive learning,* focused on using the model to learn general rules that can subsequently be used for future predictions, and *transductive approaches*, learning from the same unlabeled dataset which needs to be predicted. Transductive methods are not as popular, but in many real-world scenarios, predictions only for a specific dataset are required without the need to predict external instances [30]. The problem discussed in this paper is centered around the limitations posed by the training labels on supervised classification and in what settings adding unlabeled instances can overcome them. Therefore, the role of semi-supervised methods is to accurately extend the training set by labeling available unlabeled instances and consequently, they will be applied in a transductive manner. Supervised learning on predictions produced

by a semi-supervised technique is not common in machine learning, but there are strong reasons against using semi-supervised methods for future predictions in this analysis. Firstly, in practice, the inductive models are preferred in the final stage of the analysis because retraining is not required prior to predicting new instances. While all supervised approaches are inductive, only few semi-supervised methods are able to build a function describing the entire data space and generate inductive predictions. On the other hand, all semi-supervised techniques perform transductive labeling and can use this to extend the given training set, allowing for a more varied selection of approaches in this thesis. This setting minimizes the modeling constraints when incorporating unlabeled data in the analysis.

In real-world situations, it is rare that the labeled data points constitute a representative sample of the population. In such situations, active learning can provide more information about unknown labels and complement semi-supervised approaches. Semi-supervised and active learning techniques start with labeled data and continue by enhancing the accuracy of the model by incorporating unlabeled instances. Semi-supervised methods utilize the labeled data to extrapolate from and learn about the unlabeled dataset. These patterns are further used to infer more knowledge about the unlabeled instances. While this approach focuses on exploiting the known instances, active learning investigates how to best incorporate information from the unknown. The latter attempts to identify key observations whose labels would improve the classifier. These approaches address the limitation mentioned from different perspectives which may result in a more accurate model when combined.

Unlabeled data have mainly been explored in fields were it is widely available. However, recent research explores the advantages of additional data when the distribution of labeled and unlabeled instances is not necessarily disproportional. Ahumada [1] builds a hybrid algorithm which includes unsupervised, semi-supervised and supervised stages in the modeling process in order to improve the efficiency of semi-supervised methods when utilizing only a few unlabeled observations. Christoudias [11] focuses on audio-visual speech recognition and proposes an adaptive algorithm improving the classification of standard models trained on small proportions of unlabeled instances. Even the potential of using only a few unlabeled data is supported in various fields. Similarly, Teng [51] introduce a progressive Support Vector Machines (SVM) model which achieves high performance when applied on text classification. Magnetic Resonance Imaging (MRI) data classification is shown to be more effective when applying semi-supervised learning methods such as low density separation and semi-supervised discriminant analysis [36].

Although, few studies have analyzed the advantages of small, representative amounts of unlabeled instances with the goal of enhancing predominant patterns in the data distribution, evidence of which is presented in this paper. Varying the size of the unlabeled dataset during the analysis, produces a more comprehensive evaluation of a semi-supervised model's performance. Research indicates that increasing the proportion of labeled instances in the training dataset produces a monotonic increase

in the performance of semi-supervised link-based classification [23].

1.2 Objective

Some of the semi-supervised research problems resort to unlabeled data because the amount of labels is insufficient. The scope of this thesis is more general as it focuses on how unlabeled data can improve supervised learning techniques in all contexts, from scarce to abundant labels. Based on the results, this thesis aims to recommend solutions appropriate to real-world situations.

To enhance the robustness of the results, the properties of the methods will be evaluated with different proportions of labeled and unlabeled data. Particular focus is given to the effects of outlier removal and different active learning strategies. Therefore, in addition to the principal goal, this analysis will evaluate widely used approaches on different training scenarios, with the aim of robustly identifying good practices in training dataset pre-processing.

With regards to the scope of the analysis conducted in this thesis, unlabeled data points should be considered an intermediary tool in the learning process which aim to assist the supervised methods in improving the labeling process for future observations. The labels of interest are associated with observations from the test set, consequently the supervised learning approaches are evaluated on this set.

2 Data

2.1 Data sources

The dataset is obtained from the UCI Machine Learning Repository [40] which collects and maintains many of the datasets used by the machine learning community. The dataset selected in this thesis is the UCI Bank Marketing dataset [38]

The data was initially collected by a Portuguese bank during a telemarketing campaign dedicated to selling a service. The campaigns were carried by human agents and involved calling clients to present them with an attractive offer. A predefined script assisted them in successfully selling long-term deposits. A term deposit is a safe investment, especially appealing to risk averse investors. After multiple campaigns, an internal project was initiated designed to decrease the number of phone calls by identifying and contacting only the merchants most likely to subscribe to the term deposit.

2.2 Data description

The reports supplied to the agents during the campaign provide the data for the predictors. The reports contain necessary information for the agents when talking to the client, such as contact details, basic personal information and specific bank client details [37]. Among all the initial features, the analysis here uses explanatory variables which are not unique to the client, such as phone number.

Contacting a client can have 11 different outcomes: successfully subscribed to a deposit, rejected the offer, not the phone number owner, cancelled phone number, did not answer, fax number provided instead of phone, abandoned call, aborted by agent, postponed call by the client, postponed call by other than the client, and postponed due to voice mail. Basically, all outcomes excluding successful are unsuccessful because the client did not subscribe to the deposit [37]. These values were processed to generate a binary response variable.

Predictors:

- Personal client information
  - Age at the moment of the last contact (continuous)

  - Marital status: married, single or divorced (categorical)
  - Education level: illiterate, elementary school, secondary school, 9 years mandatory school, high school, professional course or university degree (categorical)

- Bank client information
  - The client has delayed loans (binary)
  - Average annual balances of all accounts belonging to the client (continuous)
  - Client owns a debt card (binary)
  - Client owns a credit card (binary)
  - Client owns a mortgage account (binary)
  - Clients owns an individual credit (binary)

- Contact information
  - Number of calls made in the last campaign
  - Mean duration of phone calls

- Previous campaign information
  - Number of days passed since the previous campaign
  - Total number of previous calls
  - Result of the last campaign

2.3 Data cleaning and transformation

This dataset has missing entries for some personal client information such as marital status; the data seem to be missing at random. Some of the modeling techniques applied in this analysis are unable to handle missing data or categorical variables. Since the thesis compares the performance of several algorithms trained on this dataset, observations with at least one missing entry are removed to make it possible to compare different algorithms. Consequently, the dataset's size is reduced from 45,211 to 30,488 observations.

The categorical features are transformed into multiple dummy variables to be included in the analysis, bringing the dataset to a total of 24 final predictors. These features have distinct units and scales which can strongly influence distance based algorithms such as clustering. Normalization mitigates this by scaling all feature values between 0 and 1. Normalizing or standardizing the data has become a standard step of pre-processing in data mining.

The number of clients who subscribed to a long-term deposit due to the campaign is significantly smaller than the total number of clients who rejected the offer. Only 12,6% of the persons contacted accepted the campaign offer. Since this is the category of interest in the analysis, the modeling encounters the well known class imbalance problem.

2.4 Data processing

The imbalanced nature of data is a well known challenge in the data mining community. Most techniques focus on the overall accuracy, and tend to ignore the instances belonging to the minority class. However, generally, the smaller class contains the information of interest. Approaches exist which focus on extracting accurate, relevant information from under-represented classes of data. These methods can be categorized based on the analysis stage in which they are involved.

2.4.1 Imbalanced classes and resampling

At data level, this problem is addressed in the literature by resampling from the initial dataset in order to obtain more balanced classes. Most research is focused on undersampling the majority class or oversampling the infrequent class [19]. When undersampling, the classifier disregards some available information and its performance can decrease when discarding useful training observations. On the other hand, oversampling produces synthetic data points by randomly sampling or, most commonly, duplicating instances belonging to the minority class. In addition to increased computational and memory costs [19], this method leads to overfitting when observations are duplicated [61]. These approaches seem unreliable and thus, this project will focus principally on algorithmic steps to handle the imbalance and less on alteration in the data processing level. This issue is addressed by suitably tuning the parameters, assigning misclassification weights per class or by considering methods that contain constraints specific to imbalanced classification.

2.4.2 Outlier removal

Outlier removal is a pre-processing step intended to clean the data space from rare or atypical observations, thereby providing models with a clearer data distribution. Often, outliers are characterized by erroneous observations which may confuse the classifiers and reduce their ability to generalize. Statistics has conducted a considerable amount research on outlier detection [4, 18] and proposed two categories of methods: parametric and non-parametric. The first group uses parametric models to fit the data distribution and identify observations not likely to occur, given the

data formation. The non-parametric techniques are based on clustering and represent atypical observations as small clusters; these are going to be applied in the current analysis. The simplest applicable class of clustering methods are partitioning algorithms such as K-means, but these are sensitive to outliers and therefore, unreliable. On the other hand, density based clustering identifies dense regions of data points and marks distant observations found in low density regions as outliers. This category of methods contains popular methods such as: DBSCAN [20], WaveCluster [49] and DENCLUE [27]. In a setting characterized by lack of domain knowledge and a large set of observations, DBSCAN is proven to provide the best results [65]. Furthermore, it is able to identify arbitrarily shaped clusters.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN has the advantage of not requiring the user to provide the number of clusters, which is not the case with standard partitioning and agglomerative methods. A major benefit of using DBSCAN is robustness. Neither the noise nor the ordering of the observations affect the output. In terms of computational burden, DBSCAN has a relatively low complexity compared to other clustering implementations, $O(n \log n)$.

DBSCAN categorizes the data points as follows: core observations, density-reachable instances and outliers. The core points are located in dense regions defined by two parameters: $minPts$ and ε. In the circular vicinity with radius ε, must exist at least $minPts$ in order to consider the central instance a core point. Core points have the ability to categorize other points as density-reachable if they are in this vicinity. In the end, only the clusters' edges are reachable and the internal observations are core points. The instances which can't be reached by any core point are considered outliers.

Even though clusters found by DBSCAN have natural shapes, they must have similar densities in order for DBSCAN to preserve them all. The tightness is controlled by the combination of global parameters $minPts$ and ε provided by the user which are not adaptable to individual clusters. A combination of these two parameters provides the algorithm with the required density in order for the group of observations to be considered a cluster. However, the data distribution may be described by clusters of varying densities. Since "one size does not fit all", this analysis considers parameter combinations creating a clear separation between clusters and high homogenity within the clusters. The separation ability is measured using the Silhouette coefficient, described in the next subsection. The observations not belonging to any cluster are counted as outliers. The final number of outliers found within this framework is 834 (3.9% of the training set). However, since only unlabeled outliers are removed and the proportion of unlabeled data varies, the number of outliers changes as well and 834 becomes the maximum number of outliers across all settings.

Silhouette coefficient

The Silhouette score is computed for each individual data point. It measures the difference in strength between the observation's inclusion in the cluster to which it was assigned and its relationship with other close instance groups [43].

It uses two dissimilarity measures to capture these relationships: the mean intra-cluster Euclidean distance (a) and the mean dissimilarity between the observation and all the points forming the nearest cluster (b). The *silhouette coefficient* for an individual instance is:

$$s = \frac{b - a}{\max(a, b)}.$$

According to the definition, s can assume values in the interval $[-1, 1]$. In order for s to reach the value 1, we must have $a \ll b$. A smaller a indicates a tighter relation between the data point and its cluster, which corresponds to a more appropriate assignment. A large b implies the observation is far from the nearest cluster. Therefore, a silhouette score close to 1 confirms that the observation is appropriately clustered [43] and similarly, a value approaching -1 indicates a neighboring cluster could be a more suitable assignment. Consequently, the measure is 0 when the sample is very close to the boundary separating the 2 clusters.

By obtaining a Silhouette coefficient for all data points, decisions, such as number of clusters, can be derived using a silhouette plot. The average score is an indicator of how dense the clusters are. It becomes a good reference value for all silhouettes since it is adapted to the dataset and not only to a theoretical threshold. The highest mean silhouette returned by DBSCAN on the UCI Bank Marketing dataset corresponds to 314 clusters and all individual coefficients are shown in Figure 1. The horizontal lines represent clusters of sorted silhouettes.

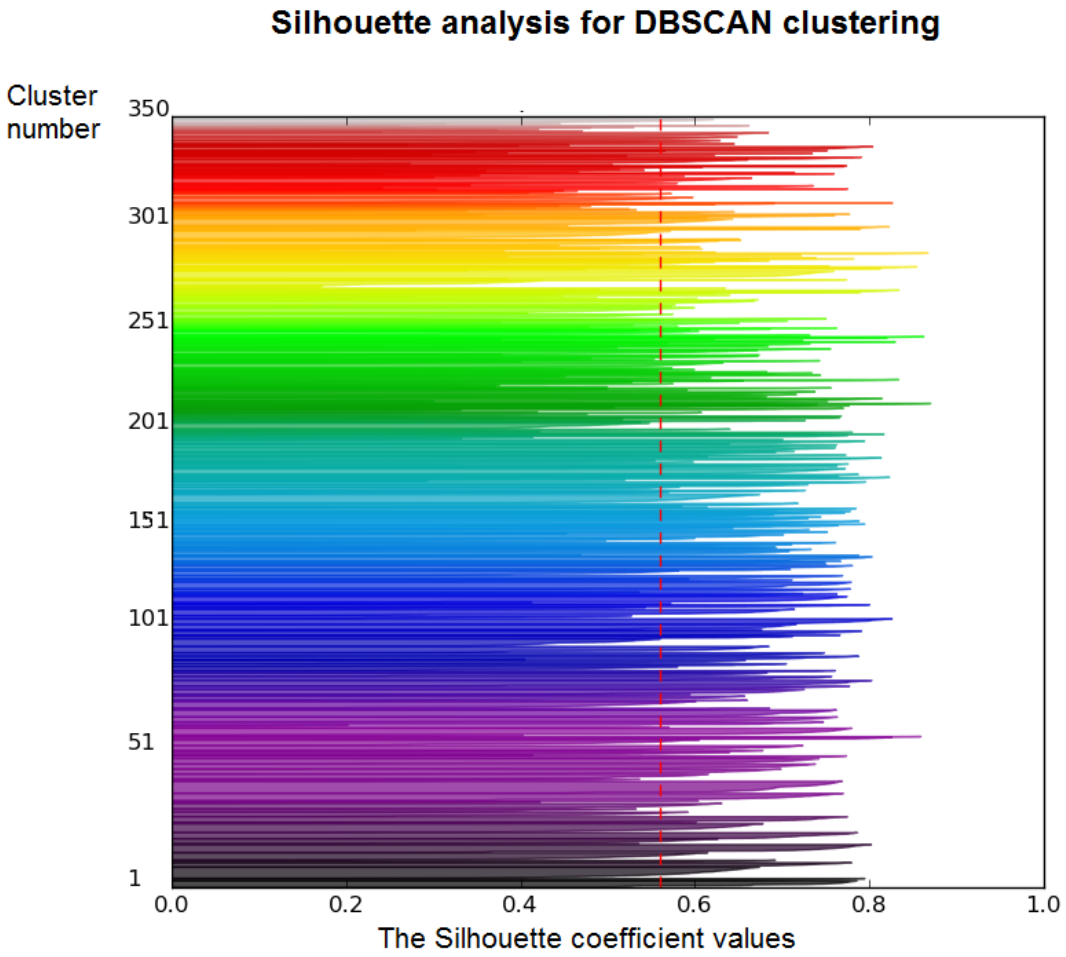

Figure 1: Silhouette analysis for DBSCAN clustering with 314 clusters clusters (ε = 0.35 and $minPts$ = 10)

The horisontal axis indicates the value of the silhouette coefficient. The interval is [0, 1] because no clusters possess values below 0. The vertical axis represents the clusters, and the thickness of the silhouettes is proportional to the corresponding cluster size.

The fluctuations of the silhouette scores are moderate and only few clusters have sub-average silhouette coefficients. These aspects, together with a relatively high average score, are positive indicators of appropriate clustering.

Most metrics of tightness, including Silhouette, are based on spherical clusters and the highest value may not fully respect some of the possibly non-spherical clusters identified by DBSCAN. This may not have a significant effect on this dataset, since the final clustering is also characterized by a relatively high homogeneity.

3 Methodology

3.1 Supervised learning

Machine learning classification is primarily associated with supervised learning which, based on a training set, creates a function which maps the feature space to the possible classes. The distinct characteristic of a supervised function is that it describes the entire data space, and it is especially useful in predicting unclassified instances. In real-world analyses, supervised learning is the most used approach when predicting new instances and evaluating the performance of a solution. The abundance of literature regarding the behavior of supervised approaches and their popularity is the main reason for using supervised learning as a means of evaluation in this thesis. The goal is to evaluate widely used approaches on different training scenarios with respect to the amount of labels and data processing techniques, with the aim of robustly identifying good practices in training dataset pre-processing. This section is not an exhaustive list of classification approaches, but a short review of important supervised learning categories and the most popular associated methods. For a more detailed review, see [31].

Instance based learning is a type of supervised learning and differs from the other supervised categories mainly during the learning process. Instead of designing a function which describes the entire space, learning is delayed until it is presented with new instances and the class is then decided locally. Methods from other categories are compatible with this approach. The difference is that the model is scoped to the single test observation and only considers instances present in the vicinity. This methodology is designed for targeted prediction, but it is not suitable for evaluating the generalization ability based on training set predictions since classifications are made only locally. In addition, a part of the data space might not be explored and the method will generalize poorly in that area. Since robustness is a primary reason for using supervised learning during evaluation, this category of techniques does not align with the goal of the thesis and no such models will be applied.

Many other categories of supervised approaches contain popular algorithms appropriate for the analysis of this thesis.In this thesis, several popular classification methods are analyzed: logistic regression, single layer perceptron, SVM and decision trees. These methods will be presented in more detail in the next sections.

No single classification technique has the ability to outperform the other methods in all contexts. Efficiency, simplicity, interpretability and applicability are the main

criteria for evaluating supervised techniques in real-world contexts. Neural networks and SVMs need larger datasets in order to create an accurate function. The UCI Bank Marketing dataset possesses a sufficient amount of data points, and thus can be considered a good candidate for these methods. On the other hand, decision trees are observed to reach higher performance when the data contains discrete variables. As a result of data pre-processing, the selected dataset contains many binary features a rule-based model would benefit from. The pruning stage incorporated in decision tree modeling reduces the effects of the noise on predictions, a benefit observed when applying SVMs as well. Decision trees and logistic regressions produce transparent results which may be used to generate domain-bounded understanding.

3.1.1 Logistic regression

The logistic regression model formulation differs depending on whether the number of classes. The method addressing binary classification is called binomial logistic regression. Logistic regression quantifies the relation between the outcome and the given predictors. This method has the advantage of providing domain knowledge through the estimated coefficients, which describe the influence of the features on the classes.

Logistic regression predictions are in the form of probabilities. For a given instance, the expected value indicates the probability that an observation belongs to the positive class. This is computed using the logistic function [32]

$E\{Y_i\} = p(Y_i = 1 \mid X_i) = \frac{exp(X_i^T\theta)}{1+exp(X_i^T\theta)}$,

where X is a matrix containing the features and a constant for the intercept, θ is the parameter vector describing the predictors influence on the class probabilities (the regression coefficients) and Y_i is the class of the ith observation. The Y_i are considered to be independently and identically distributed Bernoulli random variables. The classes are assigned based on the probability that Y_i equals 1 when predictors have the X_i values and an error term (ε_i).

Logistic regression is, in fact, a linear classifier and its decision boundary is set where $X^T\theta = 0$. The separation function is monotonic and has a sigmoidal shape. However, the logistic regression model is susceptible to overfitting [26] and a regularization procedure is the generally adopted solution, using either L1 or L2 regularization. Both shrink the regression coefficients, but L1 tends to reduce many of them to 0. This approach is known to be more helpful for sparse datasets, which is not our case here. During the parameter tuning stage, it has been observed that L2 gave better results. In this stage, different parameters and regularization techniques have been combined the most appropriate model for this dataset. The final settings were decided based on the results produced by leave-one-out cross validation applied on the training labels. By including the L2 regularization term, the optimization function becomes:

$\min_\theta \ \frac{1}{2}\theta^T\theta + C\sum_{i=1}^{n}(Y_i X_i^T\theta - \log(1 + exp(X_i^T\theta)))$,

where n is the number of features present in the dataset and C is a parameter controlling the weakness of the regularization: a smaller C produces a stronger smoothing.

We apply the logistic regression implemented in the scikit learn package from Python.

3.1.2 Single layer perceptron

The single layer perceptron is a linear classifier which decides the labels based on a linear combination of the predictors

$X_i^T W = w_0 + \sum_{i=1}^{n} w_i x_i$.

During the learning process, each feature is assigned a weight w_i, and the class is decided based on the weighted sum. If $X_i^T W$ is above a threshold, the predicted label f_i is set to 1, otherwise the observation is assigned the negative class. Part of the learning process is estimating the threshold, which connects a larger weighted sum with one class and a lower value with the opposing output. During each iteration, the single layer perceptron updates the variables' weights based on the previous weight, a learning rate α and the difference between the true label Y_i and the prediction f_i:

$w_i(t+1) = w_i(t) + \alpha(Y_i - f_i)X_i$.

Ideally, the algorithm runs repeatedly until the set of weights produces correct predictions for the entire training set. If the classes are not entirely linearly separable, 100% accuracy is not reachable and the learning does not reach convergence. In real-world contexts, it is very rare to encounter such well separated classes. In practice, it is more common to consider that the algorithm has converged when the error $\frac{1}{n}\sum_{i=1}^{n}|Y_i - f_i|$ becomes lower than a threshold specified by the user. Another stopping criterion is the maximum number of iterations. The implementation used in this thesis is taken from scikit learn module in Python and considers the number of iterations as stopping condition.

3.1.3 Support Vector Machines

Support Vector Machines (SVM) [7] is considered to be a state-of-the-art-method in machine learning classification. This approach introduces the concept of margin as a measure of distance between the separation boundary and the closest observations. The generalization ability of the model increases with the margin and therefore, SVM attempts to find the separation hyperplane which maximizes this margin.

The observations on the margin are known as support vectors and the decision function includes only these instances. The complexity of a SVM model is not influenced by the size of the training dataset, since the number of data points selected

as support vectors is generally small. This optimization step makes SVMs to be suitable for a large number of training observations. However, this advantage is not preserved when the initial space is mapped to a higher dimensional data space due to linearly inseparable classes. This transformation is computationally expensive and needs considerable tuning since choosing an adequate kernel is unintuitive. In this analysis, the SVM will be applied in the initial dimensions, and not mapped to higher dimensional spaces.

In practice, it has been observed that classes are not linearly separable and, in this case, SVM is not able to find a hyperplane because misclassified instances are not allowed to exist in the standard implementation. To address this, a popular solution proposes a soft margin which accepts but penalizes misclassifications [55]. The problem to be optimized becomes:

$$\min_{W,\xi} \tfrac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} \xi_i$$

subject to:

$$\begin{aligned} y_i f(x_i) &\geq 1 - \xi_i \\ \xi_i &\geq 0. \end{aligned}$$

The equivalence of this problem with the original optimization aim of SVM is not obvious, but fairly easy to prove. This system of equations describes the soft margin solution which allows for mislabeled instances. It introduces the slack variables ξ_i as a measure of the misclassification of the training data point x_i and maximizes the margin by minimizing $\|w\|^2$. Therefore, besides maximizing the margin, it also penalizes the slack variables through C. Since an overly large C causes SVM to overfit the data, it is good practice to estimate it by using cross-validation. In order to predict new examples, the same mapping procedure is used and their position relative to the SVM boundary set indicates their labels.

The SVM can be sensitive to imbalanced classes, therefore the classes receive weights inversely proportional to the class proportions. We use the linear SVM implemented in the Python mod scikit learn.

3.1.4 Decision trees

Decision trees have the advantage of being highly interpretable which is a valuable attribute in a business analysis where the model can be described using domain knowledge [31]. They also perform well on large datasets. The major assumption made by the proposed decision tree solution is that observations from opposite classes are different with regards to at least one features value [31].

Decision trees are constructed by splitting the values of a feature into two exhaustive intervals to create child nodes. At each step, the split is made for the variable and value which produces subsets with minimum class impurity. The Gini index I_G

measures node impurity based on the present class ratios (p_k) [8]. The following measures are computed for each node individually:

$p_k = \frac{1}{N} \sum_i I(y_i = k)$,

where N is the number of observations in that node and k iterates over all classes present in the node

$I_G = 1 - \sum_k p_k$.

The impurity measure takes values between 0 and 1 and ideally, the Gini index equals 0, which indicates the presence of only one class in that leaf node. The split is made at the node (Q) which is able to produce two groups with lowest combined impurity G, based on the parameter θ. θ is a tuple containing the variable and the associated threshold determining the split of the new nodes

$G(Q, \theta) = \frac{N_{Qleft}}{N_Q} I_G(Q_{left}(\theta)) + \frac{N_{Qright}}{N_Q} I_G(Q_{right}(\theta))$.

Each leaf node is assigned a label according to the majority class in that subset. Labeling is done starting from the root and follows the splitting rules specific to each node. The new observations are given the label of the leaf node described by their features. Similarly to the SVM modeling, the analysis assigns the classes weights inversely proportional to the class proportions.

3.1.5 Performance evaluation

The most common approach when comparing the performance of various classifiers is to split the entire dataset into two groups. The first is used in training the model and is twice as large as the test set on which the classifiers will be evaluated. The same method is used in this analysis, therefore the training is done on 70% of the observations and the rest form the test set. When the class distribution is imbalanced, it is good practice to preserve the same class ratio inside both groups.

With regard to metrics, accuracy is the standard method of evaluating supervised classifiers. However, it is not suitable when dealing with a class imbalance problem because accuracy may be high even when the minority class is completely neglected. For example, when classes are not intrinsically separated the SVM may find the maximal margin outside the conglomeration of data if the parameter C is insufficiently high. Since the primary reason for performing the analysis is often to detect the minority class, high accuracy is unsatisfactory outcome.

Given the prediction of the test dataset, the predicted labels can fall into four possible groups: true negatives, true positives, false negatives and false positives. By combining these values in relevant ways, eight measures can be created and the choice of metric depends on the aim of the analysis. Incidentally, some application domains are associated with specific metrics due to their consistent relevance within the field. In information retrieval contexts such as spotting superimposition fraud or detection of oil spills in satellite radar images [35], the dataset is seen as an

information resource and its relevance in fulfilling an informational need is computed through recall and precision [35]:

$Recall = \frac{TruePositives}{TruePositives+FalseNegatives}$,

$Precision = \frac{TruePositives}{TruePositives+FalsePositives}$.

Recall is the proportion of accurately predicted positives, while precision is the ratio of true positives in relation to the total number of positively predicted instances. In the context of the UCI Bank Marketing dataset, the recall indicates what fraction of clients was properly predicted by the model, to make a long-term deposit. The precision emphasizes how many of the clients, believed to make a term deposit, would actually do it.

For a more clear comparison between different models, the results are reported through a single metric containing both precision and recall. F1 score is commonly used for measuring the success of information retrieval and is defined as the harmonic mean of precision and recall:

$F_1 = 2 * \frac{precision*recall}{precision+recall}$.

3.2 Semi-supervised learning

The supervised approaches used in this thesis are based on inductive inference, while most semi-supervised techniques are performed in a transductive setting. The main distinction lies in the motivation of the analysis: while induction learns from the training set and a decision function is generalized for the entire data space, transduction is only focused on determining the labels of the training set. In many real-world scenarios, predictions for a specific dataset are required without the need to predict external instances [30]. In this situation, transduction may need fewer instances to produce better predictions because it can incorporate the unlabeled data in the learning process and utilize the unlabeled distribution to find the intrinsic separation areas applicable to the entire dataset. If the intention is to augment the training set with accurately labeled data to be further used in learning to predict external instances, transductive learning is the most advantageous approach for labeling the additional training data. Possessing a more thorough representation of the distribution increases the ability of the supervised learner to generalize.

The solutions proposed by semi-supervised learning consist of techniques originating from both supervised and unsupervised tasks. Semi-supervised methods have been proposed mainly in areas where unlabeled data is widely available or the labeling process is expensive since it possibly requires numerous working hours, expert opinions or special devices. In consequence, most of the semi-supervised techniques focus on building a model able to learn from a set of instances where only a small portion is labeled and the majority has no class assigned. In this case, the role of unlabeled data becomes generally to reveal the classes' distributions, which can

better indicate the location of the hyperplane separating the classes. The outcome may not be as expected if the unlabeled data increases uncertainty regarding the class membership in a region of the space which is not described by the existent labels; semi-supervised learning on distorted information about the space may also return a biased distribution. The most commonly used semi-supervised methods can be categorized into one of the following.

Generative models use explicit probabilistic models for the statistical distributions from which the data is generated. A large part of the literature is focused on mixture models which have proven to be highly accurate when the assumptions and estimates are relatively correct [52].

Heuristic approaches are not inherently built to learn from both labeled and unlabeled instances, but use supervised techniques to learn from the labeled dataset and extend the knowledge to unlabeled instances. These techniques are trained only on labeled data and use the confidence in the predictions made on the unlabeled instances to assign labels and iteratively incorporate them in the training set. This class of methods uses the principles first published under Self-Training. It's popularity has grown especially in the NLP community where research is focused on Multi-view learning and Co-training [14, 12, 39, 41, 42, 44]

Low-density separation approaches position the class boundary in regions with low density. Transductive Support Vector Machines [54] is a popular method that differs from the supervised version by considering an additional term, penalizing the unlabeled support vectors. In case the data's structure has intrinsic groups of instances belonging to the same class, Cluster-then-Label (see sec. 3.2.1) can prove to be very effective even for multi-class classification. The NLP community has published several studies about labeling done based on fuzzy clustering [57].

Graph based methods considers all examples to be nodes in a graph and defines the edges based on the similarity between the instances. There are many algorithms performing the labeling based on this type of graphical representation, including graph mincuts [6], label propagation, graph random walk [3, 29], harmonic function [64], local and global consistency [60] and others.

Semi-supervised learning is generally used when having only few labeled data. In this case there is not enough information about how the classes are distributed and therefore, the methods developed make strong assumptions about the class distribution. Since there is no previous information in the literature about the UCI Bank Marketing dataset, the methods chosen in this thesis will explore various possible distributions such as the clustering and the smoothness assumptions which are explained in the next paragraphs.

When the cluster assumption holds, the data space can be partitioned in dense regions inside which data points are more likely to share the same label. Both Cluster-then-Label and Self-Training are based on this idea. There is a high flexibility in the design of Cluster-then-Label because any labeling rule can be applied

inside the clusters, from voting to supervised or semi-supervised classification. Self-Training is less attractive because it is highly sensitive to labeling mistakes made in the first iterations and these are likely to occur [62].

Under the assumption that the class-conditional distributions are smooth, neighboring data points tend to belong to the same class. This assumption entails that observations separated by a low-density region do not have to belong to the same class. Here, proximity can be measured in the feature space or in other high dimensional space generated based on it. There are two main approaches making this assumption with slight variations. The transductive support vector machines consider the classes are being well-separated by a low density region[24] and this is where it positions the boundary. Graph-based methods instead assume that labels vary smoothly along the graph and that edge strength indicates label similarity. Graph-based algorithms are suitable for the investigation performed in this thesis since they are intrinsically transductive.

Furthermore, variations of some of the most popular algorithms will be evaluated in order to provide a robust and reasonable analysis. However, the variations are not intended to be exhaustive. In addition to the principal goal of this thesis, sensitivtity analyses are carried out with different proportions and amounts of labeled and unlabeled data.

3.2.1 Graph-based approaches

Various well known semi-supervised approaches are based on graph representations. The common characteristic is that all unlabeled and labeled data points constitute nodes in a graph and the edges are defined based on a measure of similarity. The most commonly used weight in the literature is based on Euclidian distance, such that the proximity of the nodes correlates to the weight of the edge:

$$w_{ij} = exp\left(-\frac{\sum_{d=1}^{D}\left(x_i^d - x_j^d\right)^2}{\sigma^2}\right),$$

where D is the total number of features, x_i^d is the value of component d for the observation number i and σ is a tuning parameter that penalizes larger distances.

The edges transmit the information from the known instances to the connected unlabeled data points and stronger connections facilitate the propagation. Once the graph is constructed, the most common is to minimize the following objective function:

$$\min_f \sum_{i\in\mathcal{L}} \left(y_i - f_i\right)^2 + \lambda \sum_{i,j\in\mathcal{LU}} W_{ij}\left(f_i - f_j\right)^2,$$

where f_i is the predicted class for observation i. The training set is partitioned into labeled data points (belonging to $\mathcal{L}$) and the unlabeled data $\mathcal{U}$. The first term computes the loss over the labeled instances, while the second one penalizes

similar instances having different labels. It is clear that graph-based methods are appropriate for datasets where the label smoothness property holds.

This thesis investigates the performance of two popular algorithms that appear to be highly efficient with datasets composed of more balanced classes: label propagation [63] and a graph-based approach using Gaussian fields and harmonic functions [64]. The most significant difference between these two approaches resides in the view over the sample space. The latter study introduces a continuous Gaussian field in the entire space rather than handling discrete label sets.

Label Propagation

All instances are considered to have their own distribution over possible labels and this is what the algorithm computes. The initial distributions of the unlabeled data are chosen randomly, while the known labels provide information for a stricter definition. However, instead of fixed labels with probability 1 for the known class, softer labels could be defined by lowering the preset probability. All distributions are updated during the label propagation process by using a transition matrix T:

$T_{ij} = P(j \to i) = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}}$,

where T_{ij} defines the probability of traveling from node j to the adjacent node i and w_{ij} is the weight of the edge connecting the nodes [63]. It can also be interpreted as a random walk on the graph where the label of j will randomly jump to the vertex i with probability $\frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}}$. After a finite number of iterations, the algorithm reaches convergence and labels are assigned to the unknown instances. This method differrs from the standard approach in which a function containing the weights is optimized.

An important aspect which has not been discussed until now, is the graph construction. The Label Propagation algorithm is designed around a k-Nearest-Neighbor (k-NN) undirected graph. For datasets of size as high as Bank Marketing data, the computational costs do not permit the building of a fully connected graph, therefore k must be smaller than $n - 1$.

SemiL

Graph-based method using Gaussian fields and harmonic functions [64] optimizes with respect to predictions and weights, but it defines its own function f that produces the labels to be assigned. In order to assure the smooth variation of labels along the graph, it optimizes the quadratic energy function [64]:

$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}(f(i) - f(j))^2$.

The function f that is able to minimize it is proven to have the harmonic property which means that in any unlabeled point, f equals the weighted average of f in the neighboring observations.

$f(j) = \frac{1}{d_j} \sum_{i,j} w_{ij} f(i)$ for $j = l + 1, ..., u$.

This method is designed on an 1-NN graph. There exists a scalable implementation of this approach within the software SemiL, which is also the term used in this thesis to refer to the graph based method using Gaussian fields and harmonic functions.

3.2.2 Cluster-then-Label

Most semi-supervised learning approaches are based on the extension of an existing supervised or unsupervised method with techniques inspired from the other class. Semi-supervised clustering starts from a unsupervised technique and it can vary depending on which stage the supervised knowledge about the labels is applied.

A common semi-supervised approach, Cluster-then-Label, will be evaluated in this study. It performs fully unsupervised clustering and then applies a labeling rule or a classifier within each group. In the second stage, the unlabeled instances are labelled in different ways. One alternative combines larger clusters with supervised or semi-supervised techniques on each group and predicts the unlabeled data belonging to that cluster. Another utilizes highly granular grouping and voting from the labeled data inside each cluster to determine its overall class. Although, the latter case must consider the labels' distribution among the clusters when deciding the number of clusters [2]. These approaches perform well when the partitioning matches the true data distribution [15]. Cluster-then-label is attractive when the labeled dataset potentially contains labeling inaccuracies, since labels would not influence the clustering quality [15]. This thesis evaluates both strategies in the Cluster-then-Label category.

All of the clustering approaches belong to one of the following categories: partitioning algorithms, hierarchical methods, grid-based, model-based, frequent pattern-based or constraint-based approaches [25]. Partitioning and hierarchical methods are most common, but each category has individual drawbacks and have high computational complexity. Partitioning clustering is generally sensitive to noise and outliers, while the hierarchical approach can not undo what was done in previous interative steps. Hierarchical clustering can be performed in two ways: agglomerative and divisive. Divisive hierarchical clustering begins by containing all data points and splits iteratively until a hierarchy emerges with each data point represented as cluster. The decision of merging or splitting groups of objects greatly affects the final performance because recently generated clusters are the base of the following iteration. The inability to step backwards in this iterative process and swap data points between the clusters may lead to low-quality clustering [25].

An aim of this thesis is to evaluate semi-supervised methods in order to recommend the most applicable solutions in a real-world scenario. In order to achieve this, scalability becomes an essential requirement. Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) is a good candidate because it integrates the

hierarchical approach with another clustering algorithm, resulting in a reduced computational complexity of $O(n)$ compared $O(n^2)$. BIRCH's speed and scalability is due to the construction of a tree representation of the inherent clustering structure during a single scan of the data. This concept was introduced in [59] under the name of clustering feature tree and it summarizes cluster statistics at every level. Another arbitrary clustering algorithm is applied on the "microclusters" formed at leaf nodes and creates the final "macroclusters". This addition of this last step mitigates the inability of clustering algorithms to undo previous actions.

1.2.2.1 Labeling methods

Cluster-then-Label by applying voting

One of the methods applied in this paper is Cluster-then-Label using two voting techniques inside clusters to decide the overall cluster label. BIRCH is applied during the clustering stage and the number of clusters is decided based on homogeneity and Silhouette measures. The result contains a high number of clusters because the classes are not clearly separated and voting can be highly affected by this.

In each of the 1,500 clusters, the labeled data points vote the class of the cluster. It is possible that groups of only unlabeled data points can form, and these are discarded assuming that there is insufficient information in that region of the data space to make accurate predictions. In addition, borrowing labels from neighboring clusters would reinforce the labels set during voting without additional information with which to support and cross validate. For clusters containing labeled data points, the voting can be done in a standard way in which the class with the highest number of labels is assigned to the entire cluster, or considering the difference in class proportion and therefore, one positive vote would be comparable with ~13 negative labels. This variation creates two mutally exclusive methods. The latter approach is supported by the high chance of encountering negative observations similar to the positive ones since they are much more frequent. Inside groups containing labels, but in which the voting is even, the cluster is assigned a random label with weights 1:1 or 13:1, in a way consistent with the voting technique.

Cluster-then-Label by applying semi-supervised learning

This method does not assume that data residing in the same cluster are of the same class, but allows the labels to dictate the class distributions. Inside of each labeled cluster, Label propagation is applied, while clusters completely lacking labels are removed from the training set. The number of clusters is 300 and this method differs from Label propagation applied on the entire dataset by removing the influence between clusters and treating them as independent of each other in terms of labels.

Cluster-then-Label by applying supervised learning

This approach is very similar with Cluster-then-Label performing semi-supervised learning inside the clusters as well as preserving the number of clusters. The difference lies in the classifier applied inside the cluster, which in this case belongs to the supervised class. The supervised method learns the cluster's class distribution based on the labels present inside and use it to predict the unlabeled observations grouped in the same cluster. Four different methods are based on this framework, each applying another learner to predict the clusters' labels. The supervised methods building these models are the ones described in section 3.1: logistic regression, perceptron, SVM and decision trees.

3.2.3 Transductive Support Vector Machines

SVM views observations as being points in the feature space. If the classes are not clearly separable in the initial space, a kernel function can be used to map it to a high-dimensional space in order to create a low density region between the two classes which is as wide as possible. The SVM defines the optimal separation hyperplane so that it maximizes the distance to the observations positioned closest to the boundary. This distance is referred to as margin and a higher value should lead to a better generalization ability. More details about the optimization function of a standard SVM with soft margin can be found in Section 3.1.3.

There have been considerable efforts [30, 5, 22] in the literature to extend the SVMs to be able to incorporate unlabeled data as well. The intuition behind this is that by adding unlabeled data, a more clear view of the data distribution is gained and this guides the supervised SVM boundary towards the actual low density region. The variation investigated in this thesis is Transductive SVM [30], a choice based on its broad popularity and the ability to handle imbalanced data. In comparison with the standard objective function, TSVM introduces an additional term to regularize the unlabeled data [62]:

$$\min_{W,\xi} \tfrac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} \xi_i + C' \sum_{j=l+1}^{n} \xi_j'$$

subject to:

$$\begin{array}{lll} y_i f(x_i) \geq 1 - \xi_i & \forall i = 1, l & \xi_i \geq 0 \\ y_j' f(x_j') \geq 1 - \xi_j' & \forall j = l+1, n & \xi_j' \geq 0 \\ \frac{1}{n-l} \sum_{j=l+1}^{n} \max[0, sign(f(x_i))] = r. & & \end{array}$$

The starred notations represent the unlabeled data points x_j', their predictions y_j', the slack value ξ_j' and the parameter that controls how much the unlabeled data influences the optimization C' [50]. TSVM is adapted to a setting characterized by imbalanced classes. The objective function is minimized subject to the constraint

that a proportion r of the available unlabeled data would be assigned to the positive class. $sign(f(x_i))$ represents the label that would be assigned to observation x_i. A good estimate for r can be calculated from the fraction of the known positives in the labeled training dataset.

Several variations of TSVM have been proposed, aiming to improve performance and scalability. The standard and most popular implementation of TSVM is SVM-light. On the other hand, experimenting with SVM-light on the UCI Bank Marketing dataset revealed the algorithm is not scalable. It encountered serious limitations when executed on a dataset containing 21,000 instances out of which 40% are unlabeled.

Among the methods built based on TSVM, the L2-TSVM [50] appears to preserve the performance of TSVM while making it applicable to more sparse large datasets. In the context of reaching good performance with a linear SVM, L2-TSVM enhances the TSVM's speed considerably by using the L2 loss function shown in Figure 2 and switching the labels of more than one pair in each iteration. The shape of L2 makes the gradient step to be more easily applied. The initial TSVM implementation switches at each time the labels between two unlabeled instances belonging to different classes in order to lower the objective function. L2-TSVM can switch up to $u/2$ pairs and this is one of the modifications that causes the algorithm to reach the convergence faster.

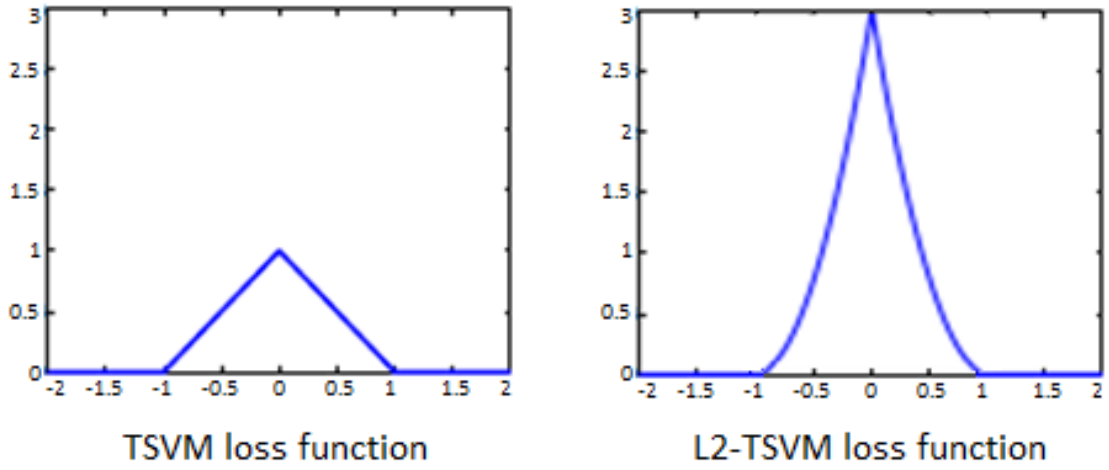

Figure 2: Loss function shapes of TSVM and L2-TSVM

3.2.4 Strengths and weaknesses of semi-supervised learning

Semi-supervised learning is especially profitable in situations where unlabeled data is available and the cost of manual labeling data is significantly higher. Obtaining labeled observations can be expensive or time consuming as it requires well-trained human annotators. At the same time, unlabeled data may be available, and the costs are generally reduced to simply collecting the data.

When assumptions about the data hold, semi-supervised learning tends to perform better than supervised approaches if combined with unlabeled data from the same distribution. Alternatively, it should be able to match or exceed performance with fewer labeled instances. Intuitively, the ability of the model to identify persistent patterns improves as the quantity of available data improves. However, this is not always the case since assumptions about the data distribution lay the foundation for most semi-supervised methods. If these modeling assumptions are misaligned with the data in question, degraded performance can be expected when compared to the corresponding supervised model trained only on the labeled data [62]. The estimation bias increases with the amount of unlabeled data added to the model [58], which is unfavorable for many real-world applications.

While shortcomings of semi-supervised methods due to model misspecification are well understood by the community, other causes of unexpected model behavior remain poorly understood. Among the possible explanations for decreased effectiveness, the most plausible appears to be the presence of outliers and other rogue instances which confuse the model rather than providing informative value. Since this class of distracting observation can potentially belong to any dataset, this aspect should be treated carefully during any semi-supervised data processing and modeling steps. To the best of our knowledge, there have been no studies in the literature investigating potential performance improvements of semi-supervised methods from careful selection of unlabeled cases. A first step in this direction is the removal of unlabeled outliers.

Semi-supervised methods are most attractive in domains involving a large pool of potentially useful, yet unlabeled data. Semi-supervised learning can have the largest impact in this context; however, current popular methods are unable to incorporate such large quantities of data efficiently. This is where scalability limitations interferes with the scope of semi-supervised learning. For example, the complexity of many graph-based algorithms is roughly $O(n^3)$ [62]. Various improvements with regard to speed have been proposed in the literature, but their performance has not yet been clearly proven. This paper attempts to evaluate some of these accelerated extended methods: L2-TSVM and SemiL.

In real-world situations, it is rare that the labeled data points constitute a representative sample of the population. This paper aims to design and evaluate various frameworks aimed to analyze data with the potential to improve the semi-supervised model's performance. In this case, a potential improvement is considered from the perspective of both labeled and unlabeled data. The model should be supplied with labels able to improve performance. Also, unlabeled data which does not degrade the modeling process should be provided. Unlabeled data removal serves the purpose of increasing the model performance with regards to outliers and rogue observations. With respect to incorporating beneficial labeled data points, these cannot be chosen since they belong to the analysis context. In a business scenario, the amount of labels is a critical constraint. In order to attain highly useful unlabeled data, new data points must be annotated. The pivotal question becomes identifying unlabeled

instances to be annotated which would best improve the model. The answer lies in active learning, which focuses on extracting most informative examples from a given dataset. The aim of active learning is to attain labels from an expert with the minimal number of queries while maximizing performance boost.

3.3 Active learning

Active learning is a subfield of machine learning constructed around policies which attempt to identify the data points most useful for the model in the learning process. Active learning focuses on the instances which potentially provide the most insight and it queries an expert regarding the labels of those selected data points. It also attempts to minimize the number of data points to be queried since labeling by an expert is usually costly. This technique assumes the existence of an expert who can provide the true label for any given data point which would be used to increase the number of labels in the training set. Active learning can be applied within both semi-supervised and supervised frameworks. In a semi-supervised context, unlabeled data is also used in the learning process.

Active learning can augment the labeled dataset with the most informative observations from the unlabeled pool. This differs from the supervised approach with regards to the source of the data points to be labeled by the expert. In this case, active learning selects the unlabeled observations which would best improve the overall performance if annotated by an expert. Since active learning is a framework easily adaptable to varied querying strategies and possibly very complex models, research topics within the domain tend to be specific [10, 13, 21, 34, 53, 64] and many areas are yet to be explored.

3.3.1 Data access

Active learning adapts the querying process to several scenarios where pool-based sampling appears to be the most common approach [47]. The first step in pool-based sampling is to evaluate and rank instances based on the given strategy. The top ranked instance or group of instances is labeled by an expert and is added to the training set on which the classifier will be retrained. If only one instance is labeled at a time, a new ranking is computed based on the retrained classifier's results and the entire process is repeated. In real-world applications, querying one instance at a time has been observed to be slow and expensive [46]. It is inefficient for a human annotator to wait for the model to repeatedly retrain on a large dataset after each new label is incorporated into the training set before knowing which instance to subsequently label.

Batch querying is much more reasonable in a business context, where models need to update their knowledge to incorporate real-time patterns. This mode implies

the selection of a batch of observations for which to obtain labels. More specific to the analysis conducted in this thesis, computing the results for all the approaches is too time consuming as each model must be retrained after every newly acquired label. Computational constraints make batch querying the only viable solution for the problem discussed here.

3.3.2 Querying strategies

The literature proposes a variety of algorithms for determining instances which would best utilize manual labeling resources. The instance selection is generally done based on one of three criteria: *informativeness*, *representativeness* and *modeling performance* [28]. An instance is informative from a modeling perspective if the label would increase the learners understanding. This is measured by the model's uncertainty with respect to the label's observation. Informativeness based selection strategies commonly exploit the data structure only partially and a sample bias may lead to serious degradation in active learning performance. On the other hand, for an unlabeled example to be representative, it must coincide with the overall patterns of the unlabeled data [28]. Finding representative instances requires exploring the dataset more extensively before approaching the instances close to the separation line.

In this thesis, informative sampling will be compared to representative sampling, as well as combined together with the intention of identifying compatibilities between classifiers and active learning techniques. For instance, label propagation-based methods may benefit more from a representative sample since some of the most uncertain labels may actually be on the contour of the dataset. However, SVM may benefit more from informative sampling which may be more relevant for support vectors' labels.

Informative active learning

The most popular approach for measuring informativeness is by studying the model's uncertainty with respect to the unknown labels. This querying strategy is named uncertainty-based sampling. A similar approach is querying by committee, which involves training several models on the labeled data and voting among the labels for the unlabeled instances. In this case, the degree of disagreement is a measure of the informativeness of knowing the true label for the instance. This approach is very similar to uncertainty sampling, but appears to be more robust[48]. Due to the scarcity of semi-supervised implementations, this thesis will apply uncertainty based sampling to identify labels which could supply the most information about the class distribution. Other approaches search for data points which would have the largest impact on the model's output or best reduce its generalization error [45]. The latter is not suitable with an imbalanced dataset because the generalization error would

favor assigning the negative class. In this analysis, the uncertainty is measured based on the class probabilities produced by the models for each individual observation. Since classes are imbalanced, the probability threshold for the minority class becomes lower than 0.5, which means that an instance may be predicted positive even if this class' probability is lower than 0.5. The thresholds may differ from one model to another.

Representative active learning

Many representative sampling techniques are based on selecting the centroids of clusters formed in various manners [9, 56]. A method of performing representative selection is to cluster the dataset and build the query at the cluster level. Since the annotation resources may be limited, the space is partitioned into a number of clusters equal to the amount of observations which can be labeled by an expert. The observations closest to the centroids of the unlabeled data points belonging to the same cluster are selected for labeling.

Informative and representative active learning

Most approaches containing both of these queries are designed for sampling one instance at a time, not batch sampling [46]. In the case of batch querying, these hybrid approaches will eventually select nearby data points. Informative querying is more popular but it may produce a sample bias due to the imbalanced nature of the classes. Representative querying may have an impact especially when there are significant regions of only unlabeled data. This thesis considers a strategy in which the labeling resources are split in half in order to obtain labels for both queries individually.

3.3.3 Annotation resources

Regardless of the sampling strategy, sufficient resources must exist to annotate the data. A stopping criterion is required for active learning querying. In theory, the model performance is the best indicator of when to stop learning, but this criterion is more suitable for strategies that label one observation at a time and would make the comparison of different methods more difficult. However in the business sector, resources are often allocated before knowing the specific needs of projects which might require active learning. This constraint is considered in the analysis as varying levels of annotation resources. The resources are defined as fractions of the unlabeled data amount. Reasonable values in a business setting would probably not exceed 20%. The predefined limits are 2%, 5%, 10% and 20%.

4 Results

The addition of unlabeled data points and application of different semi-supervised approaches to predict their labels can reveal distinct characteristics of the data structure. This gives rise to an interesting scenario in which supervised approaches are performed on training sets containing labels assigned according to different class distributions assumptions. Efficiency in this process is, firstly, determined by the selection of the data to be used during training. Highly qualitative training labels also contribute to efficiency due to their potential to represent the overall intrinsic class distribution rather than correctly describing the outcome of each individual data point. Another aspect which seems relevant, but has not been the focus of literature, is the compatibility between semi-supervised and supervised algorithms. A relationship arises between these two classes of methods when the outcome of the semi-supervised technique is utilized to train the supervised classifier. For instance, when assumptions over the class distribution differ, the predictions for the unlabeled data can confuse the supervised learner and even degrade performance compared to when only observed labels are used for training.

This chapter shows how the performance of the models vary when adding unlabeled data, applying data-processing techniques, or using active learning to query an expert with respect to the true labels of specific data points. These results are intended to assist when designing best practice procedures to overcome the scarce label limitations of supervised learning.

Following common practice in the semi-supervised research community, unlabeled data is obtained by removing labels from a portion of the training set. The most robust and reliable results are achieved when the proportion of missing labels varies. The methods are therefore evaluated on datasets with the following proportions of unlabeled training data: 5%, 15%, 40%, 60% and 85%. Since the classes are imbalanced, it is important to mention that all groups of labeled and unlabeled instances contain roughly the same ratio (~12.6%) of positives. All tables in this section present evaluation metrics computed on the same test set, providing a relevant basis for comparison. It is important to re-iterate that only the training set is altered during the analysis.

The UCI Bank Marketing dataset poses the problem of separating the clients into two groups: those subscribing to long-term deposits and those who do not. The evaluation metric considered is the F1 score [33].

Supervised learning performance without additional data

In a real-world context, the labeled portion of the dataset is data for which the company possesses full information, while unlabeled instances may be collected or used, but their outcome is unknown. It is important, in such cases, to assess if collecting additional data would improve the ability to predict external observations of interest.

Table 1 shows the performance of the selected supervised approaches in terms of precision and recall when applied on different levels of labeled data. For the sake of consistency, the proportions of unlabeled data are presented in the header for each column. For instance, in Table 1, when the proportion is 15%, the supervised model uses only 85% of the training set, which is labeled. As explained in the supervised learning section, learning from less data degrades the classifier's ability to generalize. Interestingly, removing labeled instances has significantly more impact when the model must learn based on less than 50% of the data.

Table 1: Supervised learning results trained on a proportion of the training dataset

Supervised Algorithm	Proportion of data removed from the training set due to missing labels														
	5%			15%			40%			60%			85%		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Logistic Regression	0.38	0.82	**0.52**	0.36	0.72	**0.48**	0.32	0.82	**0.46**	0.37	0.60	**0.46**	0.34	0.58	**0.43**
Perceptron	0.65	0.23	**0.34**	0.30	0.76	**0.43**	0.46	0.42	**0.44**	0.64	0.31	**0.42**	0.67	0.20	**0.31**
SVM	0.43	0.85	**0.57**	0.35	0.76	**0.48**	0.36	0.72	**0.48**	0.36	0.53	**0.43**	0.31	0.56	**0.40**
Decision Tree	0.40	0.83	**0.54**	0.38	0.73	**0.50**	0.32	0.76	**0.45**	0.31	0.75	**0.44**	0.34	0.55	**0.42**

This is the only stage at which the recall and precision are shown apart from the F1 score. All the following tables will present only F1 score in order for the comparisons to be clearer. However, Table 1 is meant to show how the F1 score varies in relation with recall and precision, and to show that the perceptron is very unstable. In the entire analysis, the perceptron is very volatile. It is important to understand that the perceptron displays this behavior when the two classes are not completely separable by a linear hyperplane. The sensitivity caused by this model's simplicity makes it inappropriate for this dataset.

Supervised learning performance with additional unlabeled data

The value of expanding the training set with unlabeled data can be estimated by comparing the predictive power of the supervised approaches without additional data (Table 1) and the performance of the supervised classifiers trained on labels assigned through semi-supervised learning (Table 2). To be more specific, semi-supervised learning is applied transductively on the entire training set, consisting of both labeled and unlabeled observations. As a result, a fully labeled training set is returned. The supervised approaches treat the resulting training as though it contains true labels. Subsequently, the function they learn describes the entire data space and provides predictions for the test set. The highest performance reached by each supervised method appears in bold.

Table 2: Supervised learning results after learning on the labels produced by semi-supervised approaches for the entire training dataset (the table reports the F1 score)

Supervised Algorithm	Semi-supervised Algorithm	Proportion of unlabeled data in the training set				
		5%	15%	40%	60%	85%
Logistic Regression	Cluster then label (voting weight 1:1)	0.46	0.46	0.46	0.46	0.47
	Cluster then label (voting weight 13:1)	0.38	0.38	0.37	0.37	0.32
	Cluster then label (Label Propagation)	0.47	0.48	0.51	0.52	0.51
	Cluster then label (Logistic Regression)	0.50	0.45	0.41	0.37	0.26
	Cluster then label (Perceptron)	0.53	0.52	0.51	0.48	0.43
	Cluster then label (SVM)	0.54	0.55	**0.57**	0.55	0.48
	Cluster then label (Decision Tree)	0.50	0.44	0.31	0.23	0.24
	Label Propagation	0.53	0.53	0.54	0.55	0.49
	SemiL	0.53	0.46	0.22	0.22	0.22
	L2-TSVM	**0.57**	**0.57**	**0.57**	**0.57**	**0.58**
	Only observed labels	*0.52*	*0.48*	*0.46*	*0.46*	*0.43*
Perceptron	Cluster then label (voting weight 1:1)	0.21	0.32	0.44	0.43	0.43
	Cluster then label (voting weight 13:1)	0.42	0.40	0.41	0.44	0.27
	Cluster then label (Label Propagation)	0.33	0.46	0.48	0.44	0.35
	Cluster then label (Logistic Regression)	0.07	0.32	0.40	0.47	0.37
	Cluster then label (Perceptron)	0.10	0.08	0.39	0.41	0.41
	Cluster then label (SVM)	0.05	0.06	0.47	0.36	0.25
	Cluster then label (Decision Tree)	0.07	0.31	0.46	0.44	0.33
	Label Propagation	0.50	0.48	0.24	0.12	0.30
	SemiL	0.49	0.23	0.23	0.22	0.22
	L2-TSVM	**0.52**	**0.51**	**0.53**	**0.55**	**0.51**
	Only observed labels	*0.34*	*0.43*	*0.44*	*0.42*	*0.31*
SVM	Cluster then label (voting weight 1:1)	0.47	0.46	0.46	0.47	0.44
	Cluster then label (voting weight 13:1)	0.39	0.39	0.39	0.39	0.40
	Cluster then label (Label Propagation)	0.54	0.54	0.55	0.55	0.53
	Cluster then label (Logistic Regression)	0.56	0.51	0.44	0.41	0.39
	Cluster then label (Perceptron)	0.58	0.57	0.54	0.50	0.43
	Cluster then label (SVM)	0.58	0.57	0.58	0.57	0.53
	Cluster then label (Decision Tree)	0.57	0.54	0.48	0.46	0.35
	Label Propagation	0.58	0.57	0.57	0.57	0.58
	SemiL	0.58	0.56	0.53	0.49	0.44
	L2-TSVM	**0.61**	**0.61**	**0.61**	**0.60**	**0.61**
	Only observed labels	*0.57*	*0.48*	*0.48*	*0.43*	*0.40*
Decision Tree	Cluster then label (voting weight 1:1)	0.47	0.46	0.46	0.44	0.41
	Cluster then label (voting weight 13:1)	0.40	0.39	0.40	0.39	0.37
	Cluster then label (Label Propagation)	0.50	0.49	0.50	0.49	0.48
	Cluster then label (Logistic Regression)	0.51	0.47	0.40	0.39	0.35
	Cluster then label (Perceptron)	0.55	0.53	0.50	0.46	0.46
	Cluster then label (SVM)	0.54	0.53	0.55	0.52	0.48
	Cluster then label (Decision Tree)	0.49	0.43	0.38	0.36	0.32
	Label Propagation	0.55	0.55	0.52	0.52	0.49
	SemiL	0.50	0.40	0.28	0.24	0.23
	L2-TSVM	**0.60**	**0.61**	**0.61**	**0.61**	**0.61**
	Only observed labels	*0.54*	*0.50*	*0.45*	*0.44*	*0.42*

From Table 2 we can see that using the inferred labels on only 5% unlabeled points can improved the final results. Some semi-supervised methods behave similar to supervised approaches, showing a reduction in performance as the amount of knowl-

edge about the labels decreases increases. Other techniques such as SemiL or L2-TSVM are not as susceptible and produce more stable results.

Supervised learning performance on a fully informed training set

Overall, the performance is improved by supplementing the training set with unlabeled data. However, since the upper performance limit is unknown, it is difficult to judge the effectiveness of the methods by quantifying improvement. A theoretical bound for partially labeled data could be the performance of the same supervised approaches with complete knowledge of the dataset's true labels.

Table 3 reports the theoretical upper bound for the selected approaches. The same parameters are utilized when computing the results, which guarantees a fair comparison regardless of data related changes.

Table 3: Supervised learning results on the true labels of the entire training set (without outlier removal)

Supervised algorithm	No outlier removal
	F1score
Logistic Regression	0.53
Perceptron	0.35
SVM	0.58
Decision Tree	0.56

By comparing tables 2 and 3, we can see that training on labels predicted by an appropriate classifier, such as L2-TSVM, can produce better results than learning from true labels. This phenomenon will be discussed more in the Discussion section, but it is a good indication that complete knowledge about labels does not necessarily reach the upper performance limit.

Modeling performance with outlier removal

The aim of the analysis is to provide the model a better coverage of the data space in order to obtain more accurately estimated models. However, learning on data for which very little is known is likely to produce incorrect semi-supervised labeling. Consequently, this propagates into the supervised modeling stage by influencing the learner's understanding about regions in which these hard to label points appear. In order to prevent this and accurately learn the class distributions, this category of unlabeled points should be removed from the training set.

Unlabeled outliers are strong candidates for such removal. During the data processing step, DBSCAN is used to identify the dataset's outliers and the unlabeled ones are removed. Table 4 presents the performance of semi-supervised algorithms applied on the training set, with and without outlier removal. In bold, are the metrics

where outlier removal performed as good or better than using the entire training set.

Table 4: Outlier removal effect on semi-supervised learning (the table reports the F1 score)

Supervised Algorithm	Outlier removal (DBSCAN)	Proportion of unlabeled data in the training set				
		5%	15%	40%	60%	85%
Cluster then label (voting weight 1:1)	No outlier removal	0.47	0.46	0.44	0.41	0.35
	Remove unlabeled outliers	0.46	**0.47**	0.43	0.38	0.27
Cluster then label (voting weight 13:1)	No outlier removal	0.46	0.46	0.44	0.41	0.37
	Remove unlabeled outliers	**0.46**	**0.46**	**0.45**	0.39	0.33
Cluster then label (Label Propagation)	No outlier removal	0.65	0.64	0.60	0.51	0.38
	Remove unlabeled outliers	**0.66**	**0.64**	**0.61**	**0.52**	0.34
Cluster then label (Logistic Regression)	No outlier removal	0.95	0.86	0.66	0.52	0.37
	Remove unlabeled outliers	**0.96**	**0.88**	**0.68**	**0.52**	0.35
Cluster then label (Perceptron)	No outlier removal	0.97	0.91	0.77	0.58	0.42
	Remove unlabeled outliers	**0.97**	**0.93**	0.76	**0.58**	0.38
Cluster then label (SVM)	No outlier removal	0.97	0.92	0.78	0.63	0.40
	Remove unlabeled outliers	**0.97**	**0.92**	**0.81**	**0.67**	0.37
Cluster then label (Decision Tree)	No outlier removal	0.94	0.84	0.63	0.49	0.33
	Remove unlabeled outliers	**0.95**	**0.85**	**0.64**	**0.49**	0.32
Label Propagation	No outlier removal	0.97	0.91	0.77	0.62	0.34
	Remove unlabeled outliers	**0.98**	**0.93**	**0.82**	**0.68**	**0.39**
SemiL	No outlier removal	0.95	0.82	0.51	0.35	0.26
	Remove unlabeled outliers	**0.96**	**0.85**	0.54	0.34	0.23
L2-TSVM	No outlier removal	0.98	0.94	0.83	0.75	0.66
	Remove unlabeled outliers	**0.98**	**0.95**	**0.86**	**0.77**	0.65

The results in Table 4 show that removal of unlabeled outliers produces definite improvements in the performance of semi-supervised learning when the proportion of unlabeled data does not exceed 50%. Some of the F1 scores reach values very close to 1.

Table 5 describes the results of supervised learning on semi-supervised outputs after applying outlier removal. When unlabeled outliers are removed, supervised approaches only learn from the labels assigned for the remainder of the dataset. The numbers in bold indicate the metrics, where outlier removal preserves or increases performance over using the entire training set. (Table 2).

Table 5: Supervised learning results after learning on the labels produced by semi-supervised approaches for the entire training dataset. During the data processing step, the training set was reduced by removing the unlabeled outliers found through DBSCAN. (the table reports the F1 score)

Supervised Algorithm	Semi-supervised Algorithm	Proportion of unlabeled data in the training set				
		5%	15%	40%	60%	85%
Logistic Regression	Cluster then label (voting weight 1:1)	0.46	**0.47**	0.45	**0.46**	0.44
	Cluster then label (voting weight 13:1)	0.37	**0.38**	**0.38**	0.36	0.30
	Cluster then label (Label Propagation)	**0.47**	**0.48**	**0.52**	**0.54**	**0.51**
	Cluster then label (Logistic Regression)	**0.50**	**0.46**	**0.41**	0.36	0.25
	Cluster then label (Perceptron)	**0.53**	**0.53**	**0.51**	0.43	0.42
	Cluster then label (SVM)	0.53	0.54	**0.57**	**0.57**	**0.48**
	Cluster then label (Decision Tree)	**0.51**	**0.44**	**0.31**	**0.23**	**0.24**
	Label Propagation	**0.53**	**0.53**	**0.55**	**0.56**	**0.53**
	SemiL	**0.53**	0.47	**0.22**	**0.22**	**0.22**
	L2-TSVM	**0.57**	**0.57**	**0.57**	**0.57**	**0.58**
Perceptron	Cluster then label (voting weight 1:1)	**0.33**	0.29	0.30	0.23	0.40
	Cluster then label (voting weight 13:1)	0.41	**0.43**	0.30	0.41	**0.41**
	Cluster then label (Label Propagation)	0.09	0.44	0.38	0.37	0.01
	Cluster then label (Logistic Regression)	**0.53**	**0.48**	0.24	0.28	**0.42**
	Cluster then label (Perceptron)	**0.51**	**0.35**	0.36	0.29	0.30
	Cluster then label (SVM)	**0.51**	**0.44**	0.07	0.18	0.04
	Cluster then label (Decision Tree)	**0.51**	**0.44**	0.35	0.26	0.32
	Label Propagation	0.48	**0.48**	**0.34**	**0.20**	0.07
	SemiL	0.46	**0.24**	**0.44**	**0.36**	0.00
	L2-TSVM	0.48	0.33	0.50	0.41	**0.58**
SVM	Cluster then label (voting weight 1:1)	0.46	**0.47**	0.45	**0.47**	**0.48**
	Cluster then label (voting weight 13:1)	**0.40**	**0.39**	**0.39**	**0.39**	0.39
	Cluster then label (Label Propagation)	**0.54**	**0.54**	**0.55**	**0.55**	**0.53**
	Cluster then label (Logistic Regression)	**0.56**	**0.51**	**0.45**	**0.42**	**0.39**
	Cluster then label (Perceptron)	**0.58**	**0.57**	**0.55**	0.49	0.41
	Cluster then label (SVM)	**0.58**	**0.57**	**0.58**	**0.58**	**0.55**
	Cluster then label (Decision Tree)	**0.57**	**0.54**	0.47	0.45	**0.35**
	Label Propagation	**0.58**	**0.57**	**0.57**	**0.57**	**0.57**
	SemiL	**0.58**	**0.56**	0.52	0.47	0.33
	L2-TSVM	**0.61**	**0.61**	**0.61**	**0.60**	**0.61**
Decision Tree	Cluster then label (voting weight 1:1)	0.46	**0.46**	0.44	**0.47**	0.40
	Cluster then label (voting weight 13:1)	**0.40**	**0.40**	0.39	0.38	0.35
	Cluster then label (Label Propagation)	**0.50**	0.48	**0.50**	**0.50**	0.47
	Cluster then label (Logistic Regression)	**0.52**	0.46	**0.40**	**0.39**	0.34
	Cluster then label (Perceptron)	**0.55**	**0.53**	0.48	**0.46**	0.45
	Cluster then label (SVM)	**0.54**	0.52	**0.55**	**0.53**	0.47
	Cluster then label (Decision Tree)	**0.51**	**0.43**	0.37	0.35	0.31
	Label Propagation	**0.57**	0.54	**0.53**	**0.52**	**0.50**
	SemiL	**0.50**	0.41	**0.28**	**0.25**	0.22
	L2-TSVM	**0.61**	**0.61**	0.60	0.60	0.60

The improvements observed in the correctness of the training's labels are visible in the generalization ability of the supervised classifiers as well.

Modeling performance with active learning labeling

When extending the dataset with a large amount of unlabeled data, it may be revealed that the initial labels do not provide information about a significant part of the feature space. This creates regions in which it may prove to be very difficult

for the semi-supervised techniques to make inferences. One way to mitigate this is to provide the learner with critical labels that would provide the most information about the unknown regions. When an expert's services can be involved in the analysis, more data can be labeled by querying the expert. Due to resource limitations in real-world scenarios, the data to be queried must be carefully selected. Active learning performs these selections based on three distinct queries: informative, representative and a combination of these each of which explore the space differently. The queries are done in batch-mode. The limiting resources are defined as fractions (2%, 5%, 10% or 20%) of the total number of unlabeled instances.

Table A in the Appendix presents the effects of active learning on semi-supervised modeling. The active learning strategy which provides the highest increase is shown in bold for each semi-supervised technique. The performances increase only slightly due to the new labels. One reason is that the annotation of most informative labels may not distribute the resources in the most efficient way because all observations are selected independently and they could be very similar to each other. On the other hand, the random selection of unlabeled data by random sampling is an experiment designed for this thesis and it does not describe the division of unlabeled and labeled data in the majority of real-world scenarios. Here, the initial labels are a random sample from the training data and therefore, are highly representative, which reduces the impact of the representative active learning.

The additional labels have a larger impact when the proportion of unlabeled data is higher and therefore, the number of queried observations increases. As mentioned earlier, better training accuracy does not always imply a better generalization ability. Tables 7 , 8 and 9 present the supervised learning results when 2% of the unlabeled set has been annotated beforehand. Only the results for 2% are reported because they are as good as when labeling 5% of the data, and it is advantageous to save human resources. The highest numbers are, as suspected, obtained for 20%, but this level of resources may not be profitable compared to the performance gain. The question Do the benefits of active learning justify the use of resources? is answered by comparing the performances produced with and without additional labels. The numbers in bold indicate the values which are strictly higher when choosing active learning (compared to Table 2).

Table 7: Active learning (informative querying and 2% labeling resources) effect on supervised learning results (the table reports the F1 score)

Supervised algorithm	Active Learning	Proportion of unlabeled data in the training set				
		5%	15%	40%	60%	85%
Logistic Regression	Label Propagation	0.53	0.53	0.54	0.55	**0.50**
	SemiL	0.52	0.45	0.22	0.22	0.22
	L2-TSVM	0.57	0.57	0.57	0.57	0.58
Perceptron	Label Propagation	0.22	0.17	0.19	0.19	0.07
	SemiL	0.45	**0.48**	**0.38**	0.32	0.31
	L2-TSVM	0.45	0.33	0.27	0.25	0.01
SVM	Label Propagation	0.58	0.57	0.57	0.57	0.58
	SemiL	0.58	0.56	0.52	0.50	0.44
	L2-TSVM	0.61	0.61	0.61	0.60	0.61
Decision Tree	Label Propagation	0.55	0.55	**0.54**	0.51	0.48
	SemiL	0.49	0.40	0.27	**0.25**	0.23
	L2-TSVM	0.60	0.61	0.61	0.61	0.61

Table 8: Active learning (representative querying and 2% labeling resources) effect on supervised learning results

Supervised Algorithm	Semi-supervised Algorithm	Proportion of unlabeled data in the training set				
		5%	15%	40%	60%	85%
Logistic Regression	Cluster then label (voting weight 1:1)	0.46	0.46	0.45	0.46	0.44
	Cluster then label (voting weight 13:1)	0.38	0.38	**0.38**	0.36	0.30
	Cluster then label (Label Propagation)	0.47	0.48	**0.52**	**0.54**	0.51
	Cluster then label (Logistic Regression)	0.50	0.45	0.41	0.36	0.25
	Cluster then label (Perceptron)	0.53	0.52	0.51	0.43	0.42
	Cluster then label (SVM)	0.54	0.55	0.57	**0.57**	0.48
	Cluster then label (Decision Tree)	0.50	0.44	0.31	0.23	0.24
	Label Propagation	0.53	0.53	0.54	0.55	**0.50**
	SemiL	0.53	0.46	0.22	0.22	0.22
	L2-TSVM	0.53	0.54	0.55	0.56	0.57
Perceptron	Cluster then label (voting weight 1:1)	0.21	0.32	0.30	0.23	0.40
	Cluster then label (voting weight 13:1)	0.42	0.40	0.30	0.41	0.41
	Cluster then label (Label Propagation)	0.33	0.46	0.38	0.37	0.01
	Cluster then label (Logistic Regression)	0.07	0.32	0.24	0.28	**0.42**
	Cluster then label (Perceptron)	0.10	0.08	0.36	0.29	0.30
	Cluster then label (SVM)	0.05	0.06	0.07	0.18	0.04
	Cluster then label (Decision Tree)	0.07	0.31	0.35	0.12	0.32
	Label Propagation	0.50	0.48	0.24	0.12	0.01
	SemiL	0.49	0.23	**0.30**	**0.23**	0.22
	L2-TSVM	0.40	0.34	0.51	0.53	0.52
SVM	Cluster then label (voting weight 1:1)	0.47	0.46	0.45	0.47	**0.49**
	Cluster then label (voting weight 13:1)	0.39	0.39	0.39	0.39	0.39
	Cluster then label (Label Propagation)	0.54	0.54	0.55	0.55	0.53
	Cluster then label (Logistic Regression)	0.56	0.51	**0.45**	**0.42**	0.39
	Cluster then label (Perceptron)	0.58	0.57	**0.55**	0.49	0.41
	Cluster then label (SVM)	0.58	**0.58**	0.58	**0.58**	**0.54**
	Cluster then label (Decision Tree)	0.57	0.54	0.47	0.45	0.35
	Label Propagation	0.58	0.57	**0.58**	0.57	0.58
	SemiL	0.58	0.56	0.53	0.49	**0.46**
	L2-TSVM	0.58	0.58	0.59	0.59	0.61
Decision Tree	Cluster then label (voting weight 1:1)	0.47	0.46	0.44	**0.47**	0.40
	Cluster then label (voting weight 13:1)	0.40	0.39	0.39	0.38	**0.35**
	Cluster then label (Label Propagation)	0.50	0.49	0.50	**0.50**	0.48
	Cluster then label (Logistic Regression)	0.51	0.47	0.40	0.39	0.34
	Cluster then label (Perceptron)	0.55	0.53	0.47	0.46	0.45
	Cluster then label (SVM)	0.54	0.53	0.55	**0.53**	0.47
	Cluster then label (Decision Tree)	0.49	0.43	0.37	0.35	0.31
	Label Propagation	0.55	0.55	0.52	0.52	**0.50**
	SemiL	0.50	0.40	0.28	0.24	0.23
	L2-TSVM	0.56	0.55	0.57	0.58	0.60

Table 9: Active learning (representative and informative querying and 2% labeling resources) effect on supervised learning results

Supervised algorithm	Active Learning	Proportion of unlabeled data in the training set				
		5%	15%	40%	60%	85%
Logistic Regression	Label Propagation	0.53	0.53	0.54	0.55	**0.50**
	SemiL	0.53	0.46	0.22	0.22	0.22
	L2-TSVM	0.58	0.57	0.57	0.57	0.58
Perceptron	Label Propagation	**0.55**	0.30	0.30	0.00	**0.41**
	SemiL	0.29	0.24	0.25	0.21	0.22
	L2-TSVM	0.31	0.31	0.23	0.08	0.09
SVM	Label Propagation	0.58	0.57	0.57	0.57	0.58
	SemiL	0.58	0.56	0.53	0.49	0.45
	L2-TSVM	0.58	0.57	0.58	0.56	0.56
Decision Tree	Label Propagation	0.55	0.55	**0.53**	0.52	0.48
	SemiL	0.50	0.39	0.27	0.24	0.23
	L2-TSVM	0.58	0.57	0.57	0.57	0.58

The newly labeled observations have a volatile effect on the performance: an unfortunate exploration of the data space can bias the results, while an appropriate labeling can increase the performance of the classifier.

5 Discussion

In the machine learning community, several papers [16, 17, 37] have analyzed the UCI Bank Marketing dataset for different purposes. However, most research has been focused on investigating the consequences of imbalanced classes and related applications. Among these papers, the highest performance reported in terms of F1 score was approximately 54%. Some of the models discussed in this thesis exceed this performance level.

Unlabeled data benefits

Acquiring additional data from other sources generally costs less when compared to annotating it. However, in a real-world scenario all expenses should be justified, and it is important to determine if collecting additional data would improve predictions of external observations of interest. In situations with small proportions of labeled data, the addition of many unlabeled observations may reveal that the labels were not representative of the true distribution. In such a setting, it becomes very likely that no model will accurately identify the class distribution since there may exist some portion of the data space which contains very little information.

With regards to the behavior of supervised models trained solely on the labeled portion of the dataset (Table 1), the performance depends on the size of the training data. The labeled samples are representative of the entire training set. In this thesis, the important observation is the change in performance when varying the context of the analysis, not the actual values. Therefore, comparisons between initial supervised performance and the following results take into account that Table 1 contains optimistic metrics. The sample of labeled data in a business setting would rarely be completely randomized or representative.

However, even if these estimations are optimistic, the addition of unlabeled data produces higher quality results (Table 2). Increased performance in generalization is observed in most of the semi-supervised results, primarily when the proportion of unlabeled data is smaller than that of the labeled set.

To confirm the benefits of including unlabeled data in the analysis, we can evaluate the performance between supervised models trained on a set labeled using semi-supervised algorithms (Table 2) and the same models using the training data's true labels (Table 3). The metrics based on the unlabeled data predictions can even exceed those of the true labels when the proportion of unlabeled data is very small,

such as 5%. Reaching optimal performance is possible in a balanced context where training labels are highly qualitative and represent the overall intrinsic class distribution rather than correctly describing the outcome of each individual data point. When the labeled proportion is high, there is sufficient data to clearly observe the distributions. Thus, new data points are assigned labels consistent with the intrinsic distribution and this reinforces the overall pattern while reducing the influence of less likely observations. An important conclusion derived from these results is that supervised modeling may benefit from unlabeled data even if the available labels are not considered a limitation. Intuitively, this behavior requires the unlabeled instances to be a representative sample or simply uniformly distributed over the data space.

Class distributions insights based on models' behaviors

The performances produced by the models can provide information about the class distributions. By analyzing Table 2, we can see that the perceptron is unstable which indicates that the two classes are not completely separable by a hyperplane.

All semi-supervised methods applied in this thesis produce good overall results, except Cluster-then-Label with voting which has lower values for both recall and precision compared to the other methods. When considering the conclusion drew from the results of the perceptron, it becomes understandable why this method does not achieve high performance. Two aspects contribute to the poor results: the imbalanced class distributions and the lack of a clear separation between them. These factors may lead to two types of clusters unsuitable for voting. Groups containing data points from both classes but where the majority class has more labels are likely to occur, because the overall proportion is 1 positive to 13 negatives. In such cases, voting may often generate negatively predicted clusters which mislabel positives and therefore, decrease the recall. On the other hand, clusters that are definitely positive are likely to include some negative observations because the majority class is large enough to contain some observations which are similar to positives, but do not belong to the positive class. The low precision is an indication of the presence of such clusters in the dataset. The algorithms provide insightful information about the class distributions.

As a result of unsuitable assumptions made about the data, the perceptron and cluster-then-label will not be mentioned hereafter.

Outlier removal effects

Removal of unlabeled outliers produces definite improvements in the performance of semi-supervised learning when the proportion of unlabeled data does not exceed 50%. To understand why the effects are less positive with higher proportions, we should consider that the amount of unlabeled outliers increases as the fraction of

additional unlabeled data increases. Excluding more observations from the training set increases the chance that various outliers are quality data points for the classifier, rather than erroneous or useless observations. However, the impact is generally positive, except when the fraction of unlabeled data reaches 85%.

If outliers are not removed but instead assigned labels which do not impact the supervised learning, it produces a performance increase at a semi-supervised level. However, including the predictions for such data points does not necessarily correspond to higher performance on the test set. For instance, the predictions of Cluster-then-Label (with semi-supervised or supervised classifiers), label propagation and L2-TSVM are more accurate, overall. In fact, when comparing the final supervised results with outlier removal (Table 5) and without outlier removal (Table 2), on labels set by the same semi-supervised technique, it can be seen that a higher semi-supervised performance on the training set (Table 4) may lack the same improvement on all the supervised results which are based on it.

On the other hand, although SemiL generally produces more accurate labels for the training set when outliers are removed (Table 4), the effects of the outlier removal on the supervised models are not consistent for similar proportions of unlabeled data (Table 5). Cluster-then-Label (with label propagation), label propagation, and SemiL all use graph-based algorithms to some extent, but the first two are different in the number of neighbors directly influencing the individual predictions. SemiL builds a 1-NN graph which may be more sensitive to pre-processing changes.

SVM appears to benefit the most from outlier removal among the supervised learners. The presence of outliers may cause the models to learn an overly complex separation between classes which leads to overfitting. Since SVM builds the class boundary based only on a small portion of the data, the support vectors, SVM is especially sensitive to outliers and this is visible in the performance increase when these are removed. Outliers have different definitions in the literature and the way they are defined by DBSCAN may locate them between classes and thus, helping SVM in computing the margin when removed.

Various techniques are available for identifying outliers and the defining characteristics of the output may vary slightly. The outlier removal performed by DBSCAN removes the unlabeled data points scattered between dense regions of observations and implicitly creates a more clear separation between the classes. This affects the classifiers' learning process. During clustering, the groups of instances become better delimited and the predictions within the clusters should be more confident. For graph-based solutions, outlier removal decreases the influence of separate groups of observations on each other. The unlabeled noise between dense groups of points behave like a bridge and excluding them produces lower weights for the edges connecting different groups of instances.

In conclusion, outlier removal may be an appropriate method of removing unlabeled data which may confuse the classifier when the dataset consists of more labeled data

points than unlabeled. It also increases the generalization potential of the semi-supervised models by providing a more robust learning process (not like SemiL).

Active learning

Active learning is applied in the selection of instances to be assigned labels by an expert. Active learning research has developed various querying strategies. In this thesis, the informative, representative and a combination of both sampling techniques are considered. Since this approach is not fully automated, but needs human labeling, higher costs are associated with applying it. In business applications, annotation resources are often preset before the analysis. The levels considered in this thesis are 2% , 5%, 10% and 20% of the available unlabeled data points. Consequently, two major questions arise: Is active learning improving the classifiers' performance? and Does an increase in the number of labeled instances necessarily improve the results?.

Informativeness or more specifically, uncertainty, querying can only be applied for models which produce a confidence measure for their predictions. Among the semi-supervised methods, only label propagation and L2-TSVM produce these confidence measures. SemiL does not return the model's certainty, but its learning process is very similar to label propagation and therefore, the informative queries are made based on the uncertainty returned by label propagation with 1-NN. In all cases, semi-supervised modeling performance (Table A from Appendix) increases, but only at higher proportions of unlabeled data, such as 40% or higher. This behavior is reasonable since more observations receive their true labels when the proportion of unlabeled data is large. It is important to note that the majority of the positive instances that are assigned the correct labels due to active learning were not provided through expert annotation. The observed increases in performance are not induced by humanly labeled data, but by increasing the information provided to the learners. In the case of L2-TSVM, there is a concern about where the L2-TSVM boundary is placed because almost all the informative data points are negatives. Since an increase in precision is observed, it seems that annotating previously uncertain labels moves the boundary closer to the positive class. Obtaining knowledge about observations belonging mostly to one class can have opposite effects, depending on the modeling assumptions. While such labels guide the L2-TSVM separation boundary in the right direction, they have a negative impact on SemiL where the new negative labels propagate and reinforce their class.

Annotating a representative sample of a large pool of unlabeled data improves the results of the semi-supervised models based on clustering and graphical representations of the data space (Table A, Appendix). Voting is the least affected by the additional labels, whereas graph-based approaches are the most affected. When a significant proportion of instances are missing labels, representative querying produces better results for graph based techniques than uncertainty querying. This is likely due to the presence of relatively isolated low-confidence instances in the

uncertain sample which contribute less to exploring the data space. As expected, L2-TSVM benefits much more from an informative query rather than obtaining representative labels.

For methods such as label propagation, both querying methods provide better results by informing the learner about different portions of the data space; therefore, it is reasonable to assume that the model could be further improved by a strategy combining both methods. However, Table A in the Appendix shows that this is not necessarily the case. When learning with label propagation, there seems to be a need regarding new labels depending on the amount of uncertainty present in the data space (amount of unlabeled data). On the other hand, SemiL has an interesting behavior: although the informative labels decrease its performance, combining both queries creates a good balance that provides the best results.

With regards to annotation resources, increasing the levels does not seem to have a significant influence on the effectiveness of either of the querying strategies, when a dataset contains more labeled observations than unlabeled. L2-TSVM appears to be the only method that benefits from only 10% more informative labels, even for small proportions of unlabeled data. This resource increase causes the uncertainty labeling to achieve a confidence interval containing positives as well. Such a change makes a notable impact for margin separators like L2-TSVM. Since for all other methods, overall generalization performance is roughly the same regardless of resource availability, it is most advantageous to focus on the lowest level of annotated data (2%). This provides approximately the same results on the test set and is least costly in terms of used resources. However, additional labeling at the preset levels, does not seem to produce better results on the final test data (Tables 7, 8 and 9). A more accurate labeling of the training data does not necessarily imply improved prediction over the data space. In addition, combining active learning with outlier removal does not improve the results of the studied models.

In conclusion, graph-based methods seem to learn more from both informative and representative labeling: alternating them dependent on the unlabeled proportion or combining them regardless of the proportion. Not all graph-based methods react the same, but based on the evidence presented in this paper, both types of active learning techniques should be considered in practice to obtain the best results. Little correlation is observed between the number of points labeled by an expert and improved generalization. Therefore, in a real-world setting, an increase in resources should be first supported by previous analysis. Also, it seems that the outlier removal process does not help queries to select better data from which to learn.

Applicability of semi-supervised approaches in real-world situations

The conclusion thus far is that extending the unlabeled data generally increases the ability to generalize. Distinct semi-supervised methods behave differently however and are not equally efficient. A more thorough investigation is required in order

to recommend a favorable combination of methods as a solution to the limitations posed by supervised learning.

Efficiency and applicability are the main criteria for assessing the quality of semi-supervised techniques as highly reliable approaches in real-world context. All semi-supervised approaches analyzed in this thesis are scalable, but their performance depends of the proportion of unlabeled data and the supervised approach applied on their predictions.

Among all the methods evaluated in this analysis, Cluster-then-Label is the fastest method, regardless of the decision rule used inside the clusters. The semi-supervised approaches are scarce, but this issue can be overcome by combining Cluster-then-Label with any supervised classifier that is adequate for the current data. Overall, associating clustering with supervised learns appears to provide better results than using it in combination with label propagation or voting. More specifically, Cluster-then-Label produces the best results when the inside predictions are made by SVM. SVM is known to have a good generalization ability and this can be exploited at a high level (trained on the entire training set) or more granularly by treating regions of the space individually. With respect to voting, we can conclude that weighting them dependent on the class proportions does not produce better classifications.

L2-TSVM, label propagation and SemiL can cope well with sparse data sets and utilize the sparsity in order to increase the computation speed. However SemiL performs poorly for higher proportions of unlabeled data by predicting all observations to belong to the positive class.

Label propagation and L2-TSVM appear to produce the best results (Table 4). However, when dealing with imbalanced classes in information retrieval contexts, recall is generally given more importance than precision. The focus of the problem is detecting the relevant class and negative observations having similar behavior to the minority class are expected to exist. However, it is undesirable for a high fraction of negatives to be incorrectly predicted as positive in real-world contexts. Costs may be associated to both false positive and false negatives. According to these considerations, L2-TSVM outperforms label propagation because it maintains the recall at precision at similar levels with a slightly higher recall.

It seems that L2-TSVM outperforms the other semi-supervised approaches and is apparently compatible with all of the supervised models. L2-TSVM is well adapted to handle datasets characterized by imbalanced classes. Part of the optimization procedure is the constraint that the ratio of positive to negative predictions must match the ratio of the training set. When the unlabeled data is not well known, the analysis can not guarantee that the labeled instances are representative of the overall available data, thus a constraint forcing a matching distribution might produce worse results. Another aspect justifying the high performance reached by L2-TSVM is the label switching performed in each iteration. Furthermore, the L2-TSVM learning procedure is based on a smoothness assumption which is often assumed in supervised

learning approaches. This compatibility is supported by the stability of the L2-TSVM's outcome regardless of the supervised modeling process.

However, among all semi-supervised approaches, L2-TSVM is the only algorithm problematic in the way it deals with large proportions of unlabeled data. The implementation performs a number of switches in each iteration and attempts to find confidence levels for all unlabeled data points. This process in L2-TSVM is extremely time consuming when handling unlabeled data in proportions exceeding 50%.

In a real-world setting, where the proportion of unlabeled data is low and the labels are generally representative of the dataset, L2-TSVM would apparently be an efficient and reliable method. Label propagation might become a better choice if one of these criteria is unfulfilled.

At a supervised level, it appears that SVM produces the best results, followed closely by decision trees. These two methods have used class weights when learning from the training set, which addressed the problem of imbalanced classes. If there are special interactions between different supervised and semi-supervised models, they are not obvious. The performance volatility due to combining methods with different learning strategies does not seem to be a reason for concern in practice. However, this does not imply that any combination of models would provide the same results, but that the best models must be identified and the interactions between them would most likely not influence their combined predictive power.

6 Conclusions

This thesis has investigated the effects of different unlabeled data configurations on supervised learning performance. The choice of semi-supervised method is an essential determinant of the generalization ability of supervised results. Therefore, identification of a stable and efficient solution required a careful analysis of the semi-supervised methods' behavior.

Generally applicable conclusions from the analysis in this thesis are:

- unlabeled data are able to improve the generalization ability of supervised methods, even when possessing full knowledge about the available data's labels.
- removing unlabeled outliers improves the performance of classifiers, except when the proportion of unlabeled data is very high.
- graph-based semi-supervised approaches gain more from additionally labeled representative instances compared to informative data points.
- L2-TSVM is a reliable effective approach for datasets lacking less than 30% of the labels.

There are several potential improvements which could increase the robustness of results in this analysis. First, using cross-validation during the evaluation of supervised models would provide more accurate performance estimates. Studying the performance of additional semi-supervised and supervised algorithms would improve our understanding of compatibilities between methods, especially if they attempt to vary the assumptions about the data distribution. The results of active learning may have been highly affected by selecting instances in batches. A serial-mode may present a significantly different behavior and impact on generalization ability.

7 Appendix

Table A: Active learning effect on semi-supervised learning

Semi-Supervised Algorithm	Active Learning	Proportion of unlabeled data in the training set				
		5%	15%	40%	60%	85%
Cluster then label (voting weight 1:1)	Representative -2%	0.47	0.46	0.44	0.41	0.35
	Representative -5%	0.47	0.46	0.44	0.41	0.35
	Representative -10%	0.47	0.46	0.44	0.41	0.34
	Representative -20%	0.47	0.46	0.44	**0.42**	**0.36**
Cluster then label (voting weight 13:1)	Representative -2%	0.46	0.46	0.44	0.41	0.37
	Representative -5%	0.46	0.46	0.44	0.41	0.37
	Representative -10%	0.46	0.46	0.44	**0.42**	**0.37**
	Representative -20%	0.46	0.46	0.44	0.41	0.35
Cluster then label (Label Propagation)	Representative -2%	0.65	0.64	0.60	0.51	0.38
	Representative -5%	0.65	0.64	0.60	0.51	0.38
	Representative -10%	0.65	0.64	**0.61**	0.51	0.42
	Representative -20%	0.65	0.64	0.61	**0.52**	**0.45**
Cluster then label (Logistic Regression)	Representative -2%	0.95	0.86	0.66	0.52	0.37
	Representative -5%	0.95	0.86	0.66	0.52	0.37
	Representative -10%	0.95	0.86	**0.67**	0.53	0.38
	Representative -20%	0.95	0.86	0.67	**0.54**	**0.40**
Cluster then label (Perceptron)	Representative -2%	0.97	0.91	0.77	0.58	0.42
	Representative -5%	0.97	0.91	0.77	0.58	0.42
	Representative -10%	0.97	0.91	0.77	**0.59**	0.44
	Representative -20%	0.97	0.91	0.76	0.58	**0.50**
Cluster then label (SVM)	Representative -2%	0.97	0.92	0.78	0.63	0.40
	Representative -5%	0.97	0.92	0.78	0.63	0.40
	Representative -10%	0.97	0.92	0.78	0.64	0.42
	Representative -20%	0.97	0.92	**0.79**	**0.68**	**0.53**
Cluster then label (Decision Tree)	Representative -2%	0.94	0.84	0.63	0.49	0.33
	Representative -5%	0.94	0.84	0.63	0.49	0.33
	Representative -10%	0.94	0.84	0.63	0.50	0.34
	Representative -20%	0.94	0.84	**0.64**	**0.50**	**0.34**

Label Propagation	Informative -2%	0.97	0.92	0.77	0.63	0.36
	Informative -5%	0.97	0.92	0.78	0.64	0.39
	Informative -10%	0.97	0.92	0.79	0.66	0.44
	Informative -20%	**0.97**	**0.92**	**0.80**	0.69	0.50
	Representative -2%	0.97	0.92	0.77	0.62	0.35
	Representative -5%	0.97	0.92	0.77	0.63	0.41
	Representative -10%	0.97	0.92	0.78	0.66	0.49
	Representative -20%	0.97	0.92	0.79	**0.71**	**0.65**
	Info and Repres -2%	0.97	0.92	0.77	0.62	0.35
	Info and Repres -5%	0.97	0.92	0.78	0.64	0.38
	Info and Repres -10%	0.97	0.92	0.78	0.65	0.45
	Info and Repres -20%	0.97	0.92	0.79	0.69	0.56
SemiL	Informative -2%	0.95	0.82	0.52	0.36	0.26
	Informative -5%	0.95	0.82	0.52	0.37	0.27
	Informative -10%	0.87	0.68	0.45	0.35	0.28
	Informative -20%	0.88	0.71	0.48	0.37	0.30
	Representative -2%	0.95	0.82	0.52	0.35	0.26
	Representative -5%	0.95	0.82	0.52	0.35	0.26
	Representative -10%	0.95	0.83	0.53	0.36	0.27
	Representative -20%	0.95	0.83	0.55	0.39	0.30
	Info and Repres -2%	0.95	0.82	0.53	0.35	0.26
	Info and Repres -5%	0.95	0.83	0.53	0.36	0.26
	Info and Repres -10%	0.95	0.84	0.54	0.37	0.28
	Info and Repres -20%	0.95	**0.86**	**0.58**	**0.41**	**0.30**
L2-TSVM	Informative -2%	0.98	0.94	0.84	0.77	0.69
	Informative -5%	0.98	0.95	0.86	0.79	0.73
	Informative -10%	**0.99**	0.96	0.89	0.84	0.79
	Informative -20%	**0.99**	**0.98**	**0.93**	**0.90**	**0.87**
	Representative -2%	0.95	0.86	0.63	0.45	0.26
	Representative -5%	0.95	0.86	0.63	0.46	0.28
	Representative -10%	0.95	0.86	0.64	0.48	0.33
	Representative -20%	0.95	0.86	0.65	0.53	0.43
	Info and Repres -2%	0.95	0.86	0.64	0.46	0.27
	Info and Repres -5%	0.95	0.86	0.65	0.48	0.30
	Info and Repres -10%	0.96	0.87	0.66	0.51	0.36
	Info and Repres -20%	0.96	0.88	0.69	0.57	0.46

Bibliography

[1] Ahumada, H. C. and Granitto, P. M. (2012). A simple hybrid method for semi-supervised learning. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 138–145. Springer.

[2] Albalate, A., Suchindranath, A., and Minker, W. (2010). A semi-supervised cluster-and-label algorithm for utterance classification. In *Intelligent Environments (Workshops)*, pages 61–70.

[3] Azran, A. (2007). The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In *Proceedings of the 24th international conference on Machine learning*, pages 49–56. ACM.

[4] Barnett, V. and Lewis, T. (1994). *Outliers in Statistical Data*. Wiley Series in Probability & Statistics. Wiley.

[5] Bennett, K., Demiriz, A., et al. (1999). Semi-supervised support vector machines. *Advances in Neural Information processing systems*, pages 368–374.

[6] Blum, A. and Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. pages 19–26.

[7] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

[8] Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

[9] Brinker, K. (2003). Incorporating diversity in active learning with support vector machines. In *ICML*, volume 3, pages 59–66.

[10] Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, pages 273–304.

[11] Christoudias, C. M., Saenko, K., Morency, L.-P., and Darrell, T. (2006). Co-adaptation of audio-visual speech and gesture classifiers. In *Proceedings of the 8th International Conference on Multimodal Interfaces*, ICMI '06, pages 84–91, New York, NY, USA. ACM.

[12] Clark, S., Curran, J. R., and Osborne, M. (2003). Bootstrapping pos taggers using unlabelled data. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 49–55. Association for Computational Linguistics.

[13] Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*, pages 129–145.

[14] Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, pages 100–110. Citeseer.

[15] Dara, R., Kremer, S. C., and Stacey, D. A. (2002). Clustering unlabeled data with soms improves classification of labeled real-world data. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 3, pages 2237–2242. IEEE.

[16] Elsalamony, H. A. (2014). Article: Bank direct marketing analysis of data mining techniques. *International Journal of Computer Applications*, 85(7):12–22. Full text available.

[17] Elsalamony, H. A. and Elsayad, A. M. (2012). Article: Bank direct marketing based on neural network. *International Journal of Engineering and Advanced Technology (IJEAT)*, 2(6). Full text available.

[18] Enderlein, G. (1987). Hawkins, d. m.: Identification of outliers. chapman and hall, london new york 1980, 188 s., č 14, 50. *Biometrical Journal*, 29(2):188.

[19] Ertekin, S., Huang, J., Bottou, L., and Giles, L. (2007). Learning on the border: Active learning in imbalanced data classification. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 127–136, New York, NY, USA. ACM.

[20] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and*, pages 226–231.

[21] Freund, Y., Seung, H., Shamir, E., and Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168.

[22] Fung, G. and Mangasarian, O. L. (2001). Semi-superyised support vector machines for unlabeled data classification. *Optimization methods and software*, 15(1):29–44.

[23] Getoor, L. (2005). Link-based classification. In *Advanced Methods for Knowledge Discovery from Complex Data*, Advanced Information and Knowledge Processing, pages 189–207. Springer London.

[24] Goldberg, A. B. (2010). *New directions in semi-supervised learning*. PhD thesis, University of Wisconsin–Madison.

[25] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

[26] Hautamäki, V., Lee, K.-A., Kinnunen, T., Ma, B., and Li, H. (2011). Regularized logistic regression fusion for speaker verification. In *INTERSPEECH*, pages 2745–2748.

[27] Hinneburg, A. and Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. In *KDD*, volume 98, pages 58–65.

[28] Huang, S.-J., Jin, R., and Zhou, Z.-H. (2010). Active learning by querying informative and representative examples. In *Advances in neural information processing systems*, pages 892–900.

[29] Jaakkola, M. S. T. and Szummer, M. (2002). Partially labeled classification with markov random walks. *Advances in neural information processing systems (NIPS)*, 14:945–952.

[30] Joachims, T. (1999). Transductive inference for text classification using support vector machines. pages 200–209. Morgan Kaufmann.

[31] Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, The Netherlands. IOS Press Amsterdam. [Online; accessed 14-May-2015].

[32] Kutner, M. (2005). *Applied Linear Statistical Models*. McGraw-Hill/Irwin series. McGraw-Hill Irwin.

[33] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

[34] McCallum, A. and Nigam, K. (1998). Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 350–358, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[35] Miroslav Kubat, Robert C Holte, S. M. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning - Special issue on applications of machine learning and the knowledge discovery process*, 30(2-3):195–215. [Online; accessed 14-May-2015].

[36] Moradi, E., Tohka, J., and Gaser, C. (2014). Semi-supervised learning in mci-to-ad conversion predictionwhen is unlabeled data useful? In *Pattern Recognition in Neuroimaging, 2014 International Workshop on*, pages 1–4. IEEE.

[37] Moro, S., Cortez, P., and Laureano, R. M. (2013). A data mining approach for bank telemarketing using the rminer package and r tool. Technical report, ISCTE-IUL, Business Research Unit (BRU-IUL).

[38] Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.

[39] Müller, C., Rapp, S., and Strube, M. (2002). Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 352–359, Stroudsburg, PA, USA. Association for Computational Linguistics.

[40] Murphy, P. and Aha., D. (1992). Uci repository of machine learning databases.

[41] Ng, V. and Cardie, C. (2003a). Bootstrapping coreference classifiers with multiple machine learning algorithms. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 113–120. Association for Computational Linguistics.

[42] Ng, V. and Cardie, C. (2003b). Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 94–101. Association for Computational Linguistics.

[43] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[44] Sarkar, A. (2001). Applying co-training methods to statistical parsing. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

[45] Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

[46] Settles, B. (2011). From theories to queries: Active learning in practice. *Active Learning and Experimental Design W*, pages 1–18.

[47] Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

[48] Shao, H., Tong, B., and Suzuki, E. (2012). Query by committee in a heterogeneous environment. In Zhou, S., Zhang, S., and Karypis, G., editors, *Advanced Data Mining and Applications*, volume 7713 of *Lecture Notes in Computer Science*, pages 186–198. Springer Berlin Heidelberg.

[49] Sheikholeslami, G., Chatterjee, S., and Zhang, A. (2000). Wavecluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal*, 8(3-4):289–304.

[50] Sindhwani, V. and Keerthi, S. S. (2006). Large scale semi-supervised linear svms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 477–484. ACM.

[51] Teng, G., Liu, Y., Ma, J., Wang, F., and Yao, H. (2006). Improved algorithm for text classification based on tsvm. *Innovative Computing ,Information and Control, International Conference on*, 2:55–58.

[52] Thomas, P. (2009). Review of "semi-supervised learning" by o. chapelle, b. schölkopf, and a. zien, eds. london, uk, mit press, 2006. *Trans. Neur. Netw.*, 20(3):542–542.

[53] Tong, S. and Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66.

[54] Vapnik, V. (2000). *The nature of statistical learning theory*. Springer Science & Business Media.

[55] Veropoulos, K., Campbell, C., and Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60.

[56] Xu, Z., Akella, R., and Zhang, Y. (2007). *Incorporating diversity and density in active learning for relevance feedback*. Springer.

[57] Yan, Y., Chen, L., and Tjhi, W.-C. (2013). Fuzzy semi-supervised co-clustering for text documents. *Fuzzy Sets Syst.*, 215:74–89.

[58] Zhang, R. and Rudnicky, A. I. (2006). A new data selection principle for semi-supervised incremental learning. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 780–783. IEEE.

[59] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *ACM SIGMOD Record*, volume 25, pages 103–114. ACM.

[60] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. *Advances in neural information processing systems*, 16(16):321–328.

[61] Zhu, J. and Hovy, E. H. (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *EMNLP-CoNLL*, volume 7, pages 783–790. Citeseer.

[62] Zhu, X. (2005). Semi-supervised learning literature survey.

[63] Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report.

[64] Zhu, X., Ghahramani, Z., Lafferty, J., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.

[65] Zhuang, W., Zhang, Y., and Grassle, J. F. (2007). Identifying erroneous data using outlier detection techniques. *Ocean Biodiversity Informatics*, page 187.

LIU-IDA/STAT-A-15/003-SE