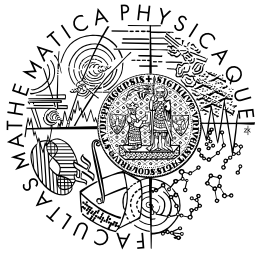

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Mitchell Borchers

Active learning in E-Commerce Merchant Classification using Website Information

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis: Mgr. Marta Vomlelová, Ph.D.
Study programme: Artificial Intelligence
Study branch: IUIPA

Prague 2023

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

Dedication.

Title: Active learning in E-Commerce Merchant Classification using Website Information

Author: Mitchell Borchers

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: Mgr. Marta Vomlelová, Ph.D., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Data and the collection and analysis of data has become an important part of everyday life. For example, navigation, e-commerce, and social media all make use of immense amounts of data to provide users with suggestions on the best routes to take, which new items they might be most interested in, and which content might fit best with their interests. A variety of algorithms and methods exist to process the data and use it to make predictions. One such algorithm is xPAL, which is a decision-theoretic approach to measure the usefulness of a labeling candidate in terms of its expected performance gain. With xPAL and other active machine learning methods an optimal strategy can be explored to classify new data points.

Keywords: probabilistic active learning xPAL machine learning multi-class classification active learning

Contents

Introduction

One of the main challenges of creating a successful machine learning model is obtaining labeled data. With easy access to a variety of modern tools, devices, and sensors, we are able to rapidly collect unlabeled data. But, in supervised learning, prediction models are trained using labeled data. The problem is that acquiring labels for the collected data can be expensive, time-consuming, or even impossible in some cases.

However, methods have been developed to help reduce the time required to label this data. Active learning is a semi-supervised machine learning method where the model is trained with a smaller set of labeled data but also aims to exploit trends within the unlabeled data. Active learning has been heavily researched in the past but typically with binary data. Multi-class active learning research is still in its infancy.

Active learning is different from other machine learning methods because it uses the unlabeled data and some evaluation criteria to determine which candidate could be the most beneficial to the model if it was given a label. In summary, the model requests the label from some oracle that provides the label then it takes this new labeled data point and rebuilds the model. We describe it as semi supervised active learning because of the oracle (typically a human) involved in the process that provides the label for the requested candidate data.

In our case we have some data (website urls) for some company or business that are given to us from our partner. From this data our partner currently utilizes human labor to browse the website and then label the url with a category (23 labels) and a sub-category (234+ tags) that branch from the main category but still have some relation. This is a repetitive and expensive task that could be automated using active learning.

To reduce the burden of human labeling we propose creating a workflow using Scrapy, Postgres, translation services, and semi supervised Active Learning (bayesian, expected error, etc.) that only require occasional interaction where a human can label a candidate that is most beneficial to the model such as using a decision-theoretic approach to measure the usefulness of a labeling candidate in terms of expected performance gain. The workflow takes the website as input, navigates to the webpage, collects and translates the text, and adds it to a database. We then run the model using the data from the database and return the model.

1. Active Learning

1.1 Introduction

Russel and Norvig succinctly define an agent and different types of learning in their book "Artificial Intelligence: A Modern Approach" (Russell and Norvig [2009]), their definition is paraphrased here. They define an agent as something that acts and a rational agent as one that acts so as to achieve the best outcome. If there there is uncertainty, then the agent tires to achieve the best expected outcome. Any component of an agent can be improved by learning from data. The improvements and techniques used to make them depend on four major factors:

- Which component is to be improved.
- What prior knowledge the agent already has.
- What representation is used for the data and the component.
- What feedback is available to learn from.

Here we will mostly be focused on the final point, "What feedback is available to learn from" but also slightly on the second point because we will incorporate Bayesian learning. There are three main types of feedback that determine the three main types of learning which are unsupervised, reinforcement, and supervised learning.

In unsupervised learning an agent learns patterns even though no feedback is provided. In reinforcement learning, the agent learns from a series of rewards or punishments. In supervised learning, an agent learns from input-output pairs, which can be discrete or continuous, to find a function that maps the pairs.

The goal of supervised learning is given a training set of N example input-output pairs:

$$(x_1, y_1), (x_2, y_2), ...(x_N, y_N),$$

where each y_j was generated by some unknown function $y = f(x)$, find a function h that approximates the true function (Russell and Norvig [2009]).

In reality, the lines separating the types of learning aren't so clear. Semi-supervised learning is also an important and widely used method. In semi-supervised learning we are given a few labeled examples that were labeled by some oracle (labeler, data annotator, etc.) and we must then make the most of a large collection of unlabeled examples. But what can we do with the unlabeled data?

Supervised learning models almost always get more accurate with more labeled data. Active learning is the process of deciding which data to sample for annotation (Munro [2021]). In other words, the central component of an active learning algorithm is the selection strategy, or deciding which of the unlabeled data could be the most useful to the model if it was labeled. Active learning uses

a selection strategy that augments the existing classifier, it is not itself a classifier but rather a tool paired with a classifier.

Many strategies for choosing the next points to label exist. First we will discuss query functions then we will briefly define three basic sampling approaches: uncertainty, diversity, and random sampling to get an idea of sampling. We will then discuss some more advanced sampling approaches that are used in our experiments. When sampling the unlabeled data an ordered list is returned and the top candidate is the candidate that is expected to be most valuable for the model, but we are not strictly limited to taking just one candidate.

1.2 Query Function Construction

There are various techniques used to construct the querying functions we have discussed. We will focus on pool-based active learning, but a number of interesting and relevant ideas appear within other active-learning frameworks that are worth mentioning.

1.2.1 Pool-Based

The learner calculates the potential gain of all the unlabeled points in the pool, then requests the label for the point that maximizes the expected information gain for the classifier. For pool-based multiclass active learning, a labeled pool and an unlabeled pool are presented to the algorithm. In each iteration, the algorithm selects one instance from the unlabeled pool to query its label (Huang and Lin [2016]).

1.2.2 Stream-Based

The learner is provided with a stream of unlabeled points. On each trial, a new unlabeled point is drawn and introduced to the learner who must decide whether or not to request its label. Note that the stream-based model can be viewed as an online version of the pool-based model (Baram et al. [2004]).

1.2.3 Membership Queries

On each trial the learner constructs a point in input space and requests its label. This model can be viewed as a pool-based game where the pool consists of all possible points in the domain (Baram et al. [2004]).

1.3 Sampling Stratagies

1.3.1 Random Sampling

Random sampling is rather self explanatory as we randomly select an unlabeled data point from the pool and have an oracle provide a label.

1.3.2 Diversity Sampling

Diversity sampling is the set of strategies for identifying unlabeled items that are underrepresented or unknown to the machine learning model in its current state. The items may have features that are unique or obscure in the training data, or they might represent data that are currently under-represented in the model.

Either way this can result in poor or uneven performance when the model is applied or the data is changing over time. The goal of diversity sampling is to target new, unusual, or underrepresented items for annotation to give the algorithm a more complete picture of the problem space (Munro [2021]).

1.3.3 Uncertainty Sampling

Uncertainty sampling is the set of strategies for identifying unlabeled items that are near a decision boundary in your current machine learning model. If you have a binary classification task, these items will have close to a 50% probability of belonging to either label; therefore, the model is called uncertain or confused.

These items are most likely to be wrongly classified, so they are the most likely to result in a label that differs from the predicted label, moving the decision boundary after they have been added to the training data and the model has been retrained (Munro [2021]).

1.3.4 xPAL

Extended probabilistic gain for active learning (xPAL) is a decision-theoretic selection strategy that directly optimizes the gain and misclassification error, and uses a Bayesian approach by introducing a conjugate prior distribution to determine the class posterior to deal with uncertainties. Although the data distribution can be estimated, there is still uncertainty about the true class posterior probabilities.

These class posterior probabilities can be modeled as a random variable based on the current observations in the dataset. For this model, a Bayesian approach is used by incorporating a conjugate prior to the observations. This produces more robust usefulness estimates for the candidates Kottke et al. [2021].

1.3.5 PAL

Probabilistic Active Learning (PAL) follows a smoothness assumption and models for a candidate instance both the true posterior in its neighborhood and its label as random variables. By computing for each candidate its expected gain in classification performance over both variables, PAL selects the candidate for labeling that is optimal in expectation. PAL shows comparable or better classification performance than error reduction and uncertainty sampling, has the same asymptotic linear time complexity as uncertainty sampling, and is faster than error reduction (Krempl et al. [2014]).

1.3.6 ALCE

Active Learning with Cost Embedding (ALCE) is a non-probabilistic uncertainty sampling algorithm for cost-sensitive multiclass active learning. They first designed a cost-sensitive multiclass classification algorithm called cost embedding (CE), which embeds the cost information in the distance measure in a special hidden space by non-metric multidimensional scaling. They then use a mirroring trick to let CE embed the possibly asymmetric cost information in the symmetric distance measure (Huang and Lin [2016]).

1.3.7 QBC

Query by committee uses an ensemble of classifiers that are trained on bootstrapped replicates of the labeled set (Seung et al. [1992]).

1.3.8 EER

Monte Carlo estimation of error reduction (EER) estimates future error rate by log-loss, using the entropy of the posterior class distribution on a sample fo the unlabeled examples, or by 0-1 loss, using the posterior probabilities of the most probable class for the sampled unlabeled examples Roy and McCallum [2001].

1.4 Miscellaneous Definitions

Beta Prior Conjugate Prior Decision-Theoretic Dirichlet Distribution Expected Performance Gain Ground Truth : The true value of a random variable. The label provided by the oracle. Posterior Probabilities Omniscient Oracles Random Variable

1.5 Summary

Now it should be more clear how the sampling strategy is the major component of active learning. The query function construction is also important but it is just a means of routing the data to be sampled. In the next chapter we will look into the specifics of xPAL.

2. Understanding xPAL

We have introduced many different active learning models in the previous section, and we will test some of these models on our data. However, we will mainly focus on using the xPAL sampling strategy and a pool based query function. The xPAL sampling strategy is a decision-theoretic approach to measure the usefulness of a labeling candidate in terms of its expected performance gain. We can estimate the data distribution but but we are uncertain about the true class posterior probabilities. The class posterior probabilities are modeled as a random variable based on the current observations. Therefore a Bayesian approach is used by incorporating a a conjugate prior to the observations. In general, the idea is to estimate the expected performance gain from a new labeled data point and select the best one (Kottke et al. [2021]). Variable definitions are listed in Table 2.1

	Definition
L	Loss
R	Risk
$R_{\mathcal{E}}$	Empirical Risk
$\mathcal{L}$	Labeled Data
$\mathcal{U}$	Unlabeled Data $\{x_1, ..., x_n\}$
$\mathcal{E}$	Labeled and Unlabeled Data
$p(x, y)$	Joint distribution of random variables x and y
$f^{\mathcal{L}}$	Classifier that maps input x to output y for class l

Table 2.1: Variable names and definitions.

2.1 Kernel

A kernel based classifier is used in xPAL which determines the similarity of two data points. The kernel function $k(x, x')$ is a function that maps two data points to a real number. The kernel frequency estimate $\boldsymbol{k}_x^{\mathcal{L}}$ of an instance $\boldsymbol{x}$ is calculated using the labeled instances $\mathcal{L}$. The y-th element of that C-dimensional vector describes the similarity-weighted number of labels of class y.

$$\boldsymbol{k}_x^{\mathcal{L}}, y = \sum_{(x',y')\in\mathcal{L}} \mathbb{1}_{y=y'} \boldsymbol{K}(x, x') \tag{2.1}$$

The Parzen Window Classifier uses the labeled data for training and predicts the most frequent class.

$$f^{\mathcal{L}}(x) = \arg\max_{y\in\mathcal{Y}} \left(\boldsymbol{k}_{x,y}^{\mathcal{L}}\right) \tag{2.2}$$

2.2 Risk

For xPAL, Kottke et al. use the classifications error as the performance measure and minimize the zero-one loss. The risk describes the expected value of the loss

relative to the joint distribution given some classifier. The zero-one loss returns 0 if the prediction from the classifier is equal to the true class else it returns 1.

$$R(f^{\mathcal{L}}) = \mathop{\mathbb{E}}_{p(x,y)}[\boldsymbol{L}(y, f^{\mathcal{L}}(x))] \tag{2.3}$$

$$= \mathop{\mathbb{E}}_{p(x)}\left[\mathop{\mathbb{E}}_{p(y|x)}[\boldsymbol{L}(y, f^{\mathcal{L}}(x))]\right] \tag{2.4}$$

$$\boldsymbol{L}(y, f^{\mathcal{L}}(x)) = \mathbb{1}_{f^{\mathcal{L}}(x)\neq y} \tag{2.5}$$

Because it is not known how the data is generated $P(x)$ Kottke et al. use a Monte-Carlo integration with all the data $\mathcal{E}$ to represent the generator. The empirical risk $R_{\mathcal{E}}$ is the average of the loss over all the data points in the dataset. Because it is not known how the data is generated $P(x)$ Kottke et al. use a Monte-Carlo integration with all the data $\mathcal{E}$ to represent the generator. The empirical risk $R_{\mathcal{E}}$ is the average of the loss over all the data points in the dataset.

$$R_{\mathcal{E}}(f^{\mathcal{L}}) = \frac{1}{|\mathcal{E}|}\sum_{x\in\mathcal{E}} \mathop{\mathbb{E}}_{p(y|x)}\left[\boldsymbol{L}(y, f^{\mathcal{L}}(x))\right] \tag{2.6}$$

$$= \frac{1}{|\mathcal{E}|}\sum_{x\in\mathcal{E}}\sum_{y\in\mathcal{Y}} p(y|x)\boldsymbol{L}(y, f^{\mathcal{L}}(x)) \tag{2.7}$$

2.3 Conjugate Prior

The conditional class probability $p(y|x)$ depends on the ground truth which is unkown. As a result the conditional class probability is exactly the y-th element of the unknown ground truth vector $\boldsymbol{p}$.

2.4 Risk difference using the conjugate prior

2.5 Expected probabilistic gain

3. Analysis

In this chapter we take a deeper look into the data and the process of augmenting the data. Our partner has provided a small sample of 1000 labeled data points. This data was manually labeled by an annotator. The data consists of a merchant name, merchant website (url), merchant category, and merchant tag as shown in Table 3.1.

merchant name	merchant url	merchant category	merchant tags
State Hospital	http://hospital.com/	Health	'{"Clinic"}'

Table 3.1: This is an example of a single data point from the original data set.

The current process consists of giving the merchant url to an annotator and the annotator then views the website and either can instantly provide a label and tags for the website or in some cases may need to browse further into the website (by viewing sibling pages such as the 'About Us' sections or product pages) to get an idea of how the website should be classified.

The merchant tags are ordered by relevance, with the first tag in the list being the most general and the final being the most specific. An example of the tag hierarchy is show in Table. 3.2 where we can see that this sample consists of data from various categories all contained within the 'Eco' side tag grouping.

Category	Level 1	Level 2	Level 3	Side Tag
Travel	Local Transport	Micro-mobility	Bike Sharing	Eco
		Public Transport		Eco
Fashion	Clothing - Other	Second Hand		Eco
Car	Charging Station			Eco
	Car Sharing			Eco

Table 3.2: This is an example of how the tags use different levels.

Similarly to the annotator our goal is to automate the navigation, collection/storing process, and classification of the website. This pipeline speeds up the browsing process and can allow the annotator to spend much less time annotating and require the annotator to only annotate data expected to drastically improve the classifier.

The initial 1000 data points we received were essentially just labels with a pointer (a url) to where the data is. The labels needed the text from the websites that the annotator viewed to begin the classifying process. To gather the text data from the websites we used the Scrapy framework to extract text data from a single top level page of a website. We chose to only scrape the top level page text because of the results published in another study where it was observed that adding more pages to the data set does not necessarily mean obtaining better results (Sahid et al. [2019]).

Out of these initial data points 179 contained links that could not be accessed or links that provided no text data that could be scraped. Out of the remaining 821 data points 274 of them were in English.

It is important for us to have the data in English as it allows us to exploit stop words when using the Scikit-Learn TF-IDF vectorizer to construct our data set. Stop words are words like “and”, “the”, “him”, which are presumed to be uninformative in representing the content of a text, and which may be removed to avoid them being construed as signal for prediction (Fabian Pedregosa et al. [2023]).

Out of the remaining 275 English data points the data was distributed into the categories as shown in Figure 3.1.

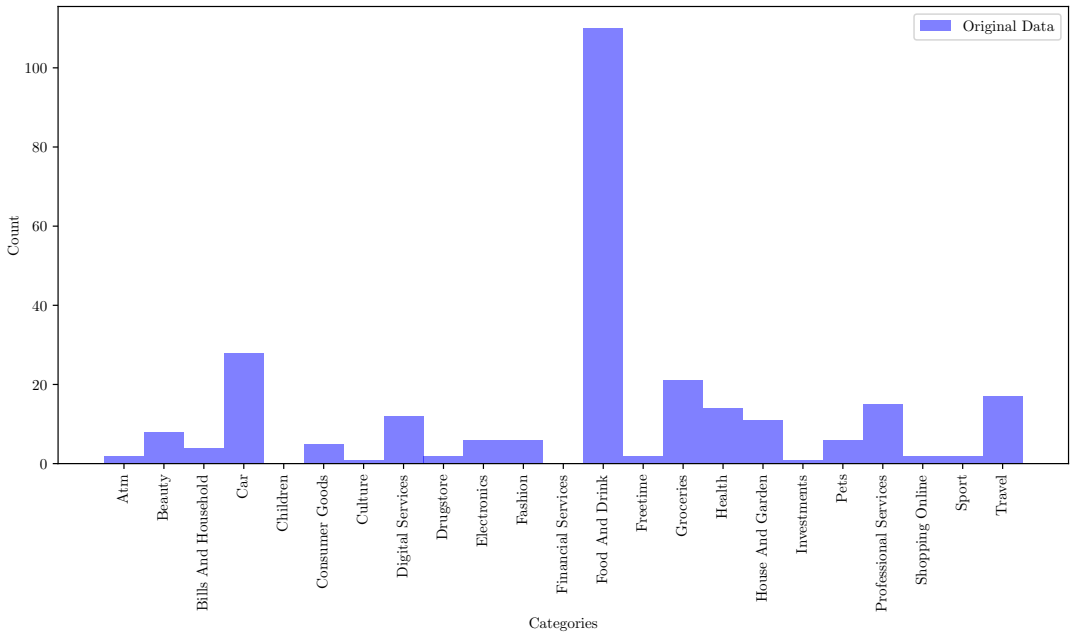

Figure 3.1: The histograms for the original usable english data.

At this stage, it was clear that our data set wasn’t representing all categories equally. The ’Food and Drink’ category has many more data points then the ’Culture’ and ’Investments’ categories which each had a single data point. The ’Children’ and ’Financial Services’ categories weren’t represented at all. Obviously this was problematic because we would like to have, minimum, three data points in each category to build train, test, and validation data sets, albeit limited as this would be.

At this point we had a significant amount of data that wasn’t being used. We decided to find a way to translate the existing data. We tried various libraries available on GitHub but we weren’t getting good results. We found that Azure had a service available and a free option of up to 2 million characters translated per month. This was a perfect option and we were able to use this api to translate the remaining data.

In addition to the original data we also manually collected and labeled 141 additional data points. All the original data and additional data are shown in Figure 3.2 and discretely in Appendix A in Tables A.2 and A.1.

In addition to translating the data we used the TF-IDF vectorizer from Scikit-Learn. We were able to find highly correlated words for each category using Chi Squared analysis with only the original data, shown below in Table 3.3. Some categories such as ’Culture’, ’Digital Services’, ’Shopping Online’ that have few

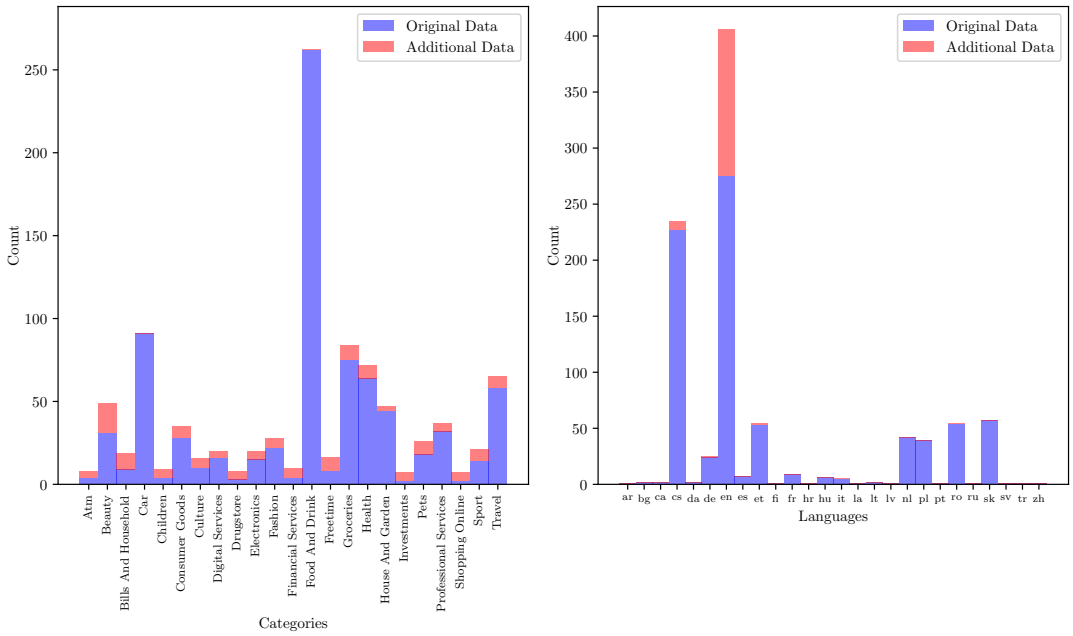

Figure 3.2: The histograms for the original and additional data for all languages.

data points have words such as 'kihnu', 'td', 'patria', respectively, which have no relative meaning to the category in English. This is likely because the collected data is sparse and the quality of the scraped text data is low.

We also calculated the variable importance using the RandomForestRegressor from Scikit-Learn and provided a list of the top 20 most important words from the TF-IDF vectorizeR. This helps us orient ourselves within the data as well as check if there may be any anomalies. The list of top 20 most important words are shown in Table 3.4.

Table 3.3: Keywords from TF-IDF with Chi Squared using the original data.

	Keyword 1	Keyword 2	Keyword 3
Atm	banking	zu	bank
Beauty	hairdressing	hairdresser	hair
Bills And Household	recycling	liberty	internet
Car	stations	auto	car
Children	toy	sold	toysrus
Consumer Goods	wallpaper	flowers	flower
Culture	museum	theater	kihnu
Digital Services	td	bitly	servers
Drugstore	alert	enable	detergent
Electronics	electronic	onoff	computers
Fashion	jewellery	men	women
Financial Services	pre	nissan	insurance
Food And Drink	bar	cafe	restaurant
Freetime	searched	casino	smartflex
Groceries	functioning	liquor	bakery
Health	optics	dental	pharmacy
House And Garden	wallpapers	paints	hardware
Investments	strive	investment	patria
Pets	breeding	pet	veterinary
Professional Services	faculty	parcel	school
Shopping Online	web	owner	joom
Sport	functional	singltrek	adidas
Travel	rooms	accommodation	hotel

Table 3.4: Variable importance, top 20 words from the vectorizer.

	importance
hotel	0.091429
car	0.053077
hair	0.029760
auto	0.028876
station	0.017934
hairdresser	0.016447
flower	0.015114
spa	0.013156
parking	0.013083
stations	0.012124
barber	0.011965
internet	0.011843
services	0.011364
rental	0.010331
salon	0.009946
accommodation	0.009681
service	0.009430
bank	0.009072
cookies	0.007105
gas	0.006985

4. Results

4.1 Original Data

Here we will explore the results of different classifiers and active learning strategies on the original data. The original data consisted of data shown previously in Figure 3.1 and discretely in Table A.1 in Appendix A.

4.1.1 Active Learning with PWC and RBF Kernel

In Figure 4.1 we have the train and test errors for four different active learning sampling strategies and we can see that xPAL and PAL both perform relatively well before starting to overfit the training data. We reviewed the resulting error data and xPAL outperformed PAL by about 1.5%. However, xPAL has much faster running time compared to PAL.

We weren't satisfied with the testing error which leveled out to about 70% for each sampling strategy. PAL and xPAL were able to rapidly reduce the testing error early on in the training process while random selection and QBC weren't able to determine the data with the highest information gain.

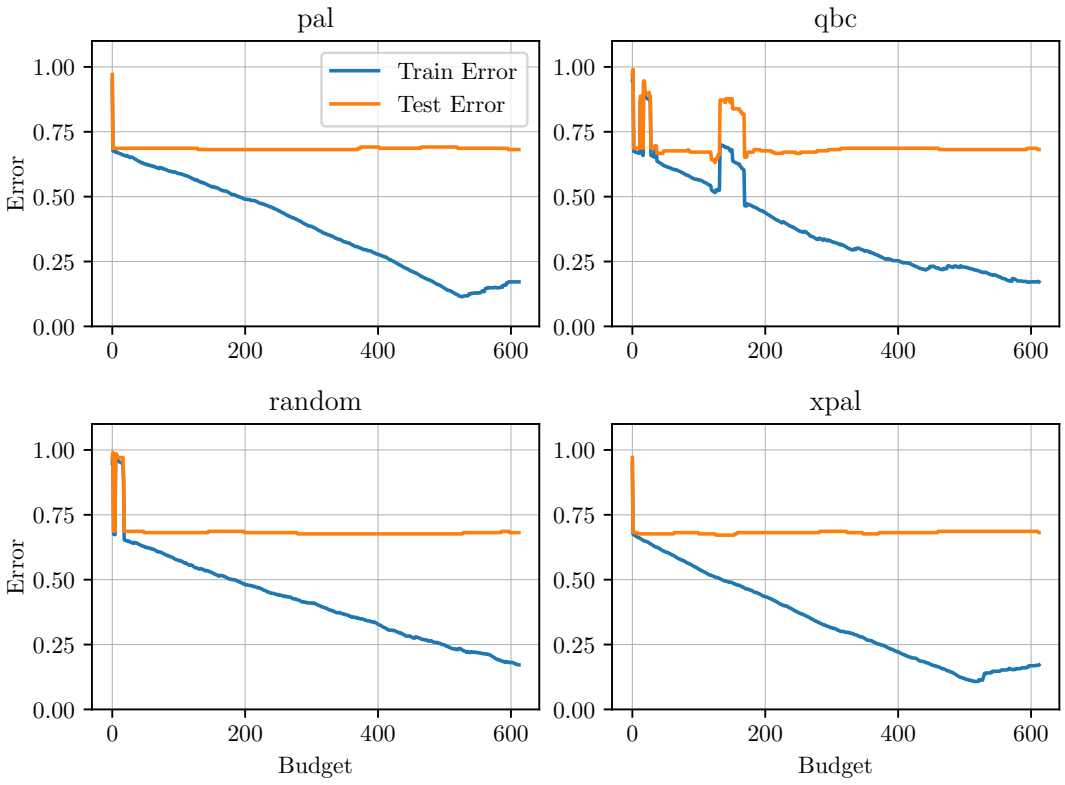

Figure 4.1: Train and test error using different query stratagies and RBF kernel for the PWC classifier.

4.1.2 Active Learning with PWC and Cosine Kernel

In Figure 4.2 we have the and test errors for the same four different active learning sampling strategies and tested on the same data except instead of using the RBF kernel the Cosine kernel was used. We found that using the Cosine kernel reduced the training error across the board by roughly 17%.

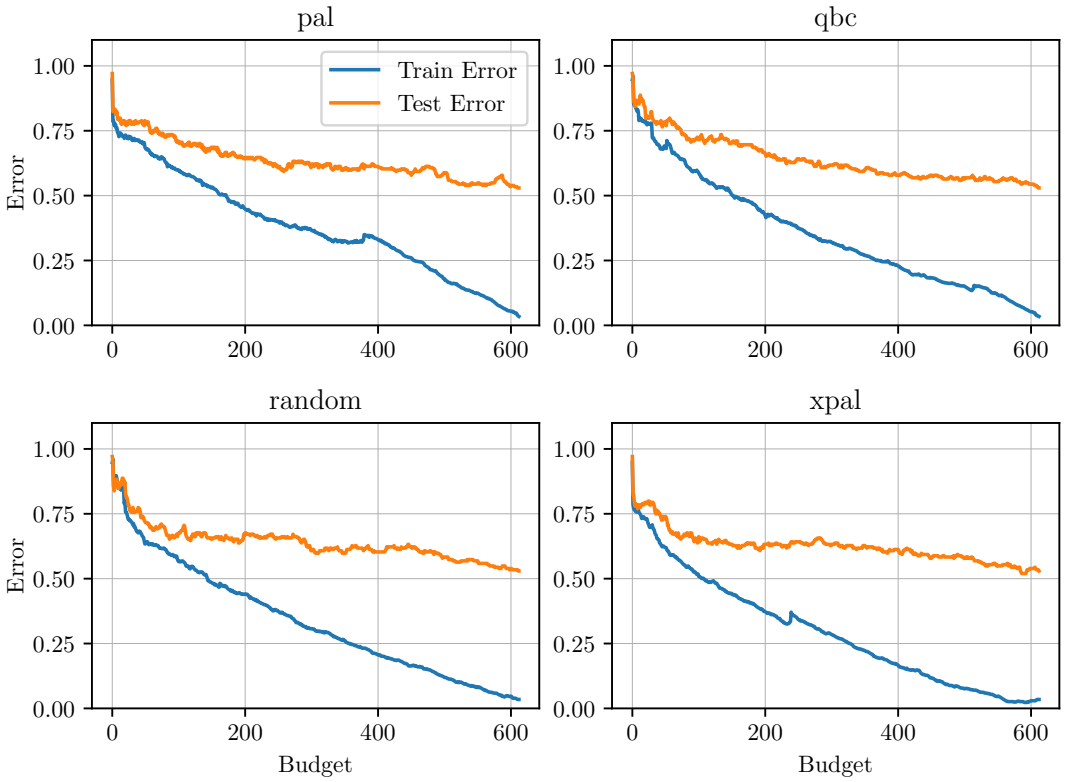

Figure 4.2: Train and test error using different query stratagies and Cosine kernel for the PWC classifier.

4.1.3 Scikit-Learn Classifier Evaluation

We also decided to try out the boilerplate classifiers from Scikit-Learn to compare performances. Again we used the original data with the same TF-IDF vectorizer as used with the previous active learning models to stay consistent. Cross validation was also used here but was not used in the previous sections. We decided to use the Cosine kernel when using LinearSVC as we learned that the performance increased in comparison to the RBF kernel when using PWC. The results for a variety of different classifiers are shown in Figure 4.3. In the box-plot, the whiskers extend from the box to the furthest data points that are within 1.5 times the interquartile range (IQR) of the box. Any data points that are beyond the whiskers are considered outliers and are plotted as individual points or symbols (diamonds) as seen in Figure 4.3.

The LinearSVC classifier performed rather well compared to most other classifiers and it is a fast running algorithm even with data that has a large feature set. We decided to look further into LinearSVC and create a few models to evaluate

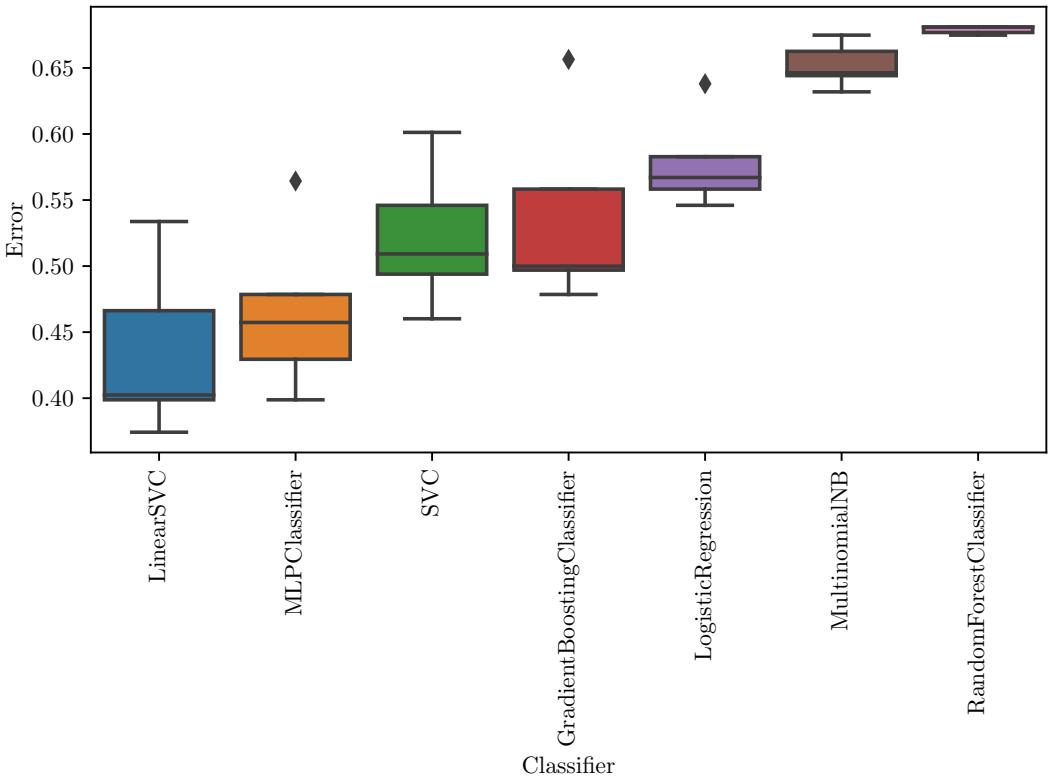

Figure 4.3: Performance comparison of standard Scikit-Learn classifiers.

its performance. We created three models, the first was a boilerplate LinearSVC with no argument modifications, the second model used the class weights parameter set to 'balanced'. The 'balanced' mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as $n\_samples/(n\_classes * np.bincount(y))$. For the third test we created a dictionary of weights for each class using the Cosine Decay function. The weights for each category ranged from 0.1 to 1.0 where the most frequent classes had smaller weights. The results are shown in Table 4.1.

	Error
Balanced Weights	0.441
Boilerplate	0.406
Cosine Decay Weights	0.392

Table 4.1: Error for three different LinearSVC models.

4.2 Original and Additional Data

Conclusion

Bibliography

Yoram Baram, Ran El Yaniv, and Kobi Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5(Mar):255–291, 2004.

Alexandre Gramfort Fabian Pedregosa, Gael Varoquaux, Vincent Michel, et al. 6.2. Feature extraction — scikit-learn.org. `https://scikit-learn.org/stable/modules/feature_extraction.html`, 2023. [Accessed 03-Mar-2023].

Kuan-Hao Huang and Hsuan-Tien Lin. A novel uncertainty sampling algorithm for cost-sensitive multiclass active learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 925–930. IEEE, 2016.

Daniel Kottke, Marek Herde, Christoph Sandrock, Denis Huseljic, Georg Krempl, and Bernhard Sick. Toward optimal probabilistic active learning using a bayesian approach. *Machine Learning*, 110(6):1199–1231, 2021.

Georg Krempl, Daniel Kottke, and Myra Spiliopoulou. Probabilistic active learning: A short proposition. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1049–1050, Prague, Czech Republic, August 2014. IOS Press. Short Paper.

Robert Munro. *Human-in-the-loop machine learning*. Manning Publications, New York, NY, July 2021.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 2:441–448, 2001.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2009. ISBN 9780136042594.

Galuh Tunggadewi Sahid, Rahmad Mahendra, and Indra Budi. E-commerce merchant classification using website information. In *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*, pages 1–10, 2019.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992.

List of Figures

List of Tables

List of Abbreviations

A. Attachments

Table A.1: Category counts of original English data.

category	category count
Atm	4
Beauty	31
Bills And Household	9
Car	91
Children	4
Consumer Goods	28
Culture	10
Digital Services	16
Drugstore	3
Electronics	15
Fashion	22
Financial Services	4
Food And Drink	262
Freetime	8
Groceries	75
Health	64
House And Garden	44
Investments	2
Pets	18
Professional Services	32
Shopping Online	2
Sport	14
Travel	58

Table A.2: Category counts of additional data.

category	category count
Atm	4
Beauty	18
Bills And Household	10
Children	5
Consumer Goods	7
Culture	6
Digital Services	4
Drugstore	5
Electronics	5
Fashion	6
Financial Services	6
Freetime	8
Groceries	9
Health	8
House And Garden	3
Investments	5
Pets	8
Professional Services	5
Shopping Online	5
Sport	7
Travel	7