

Reducing annotation burden in travel pattern analysis

Rahul Dhopeswar Sjoerd Dirksen Svetlana Dubinkina
Relinde Jurrius Ivan Kryven Chengyandan Shen
Cristian Spitoni Chandra Tamang

July 1, 2025

Contents

1	Introduction	2
2	Methods	3
2.1	Classification methods	3
2.1.1	Random forests	3
2.1.2	Bayesian Neural Nets	3
2.2	Active learning	4
2.2.1	Sampling strategy	5
2.2.2	Timing of sampling	5
2.2.3	Model updates	5
2.2.4	Model evaluation	6
2.3	Uncertainty-based sampling	6
2.3.1	Entropy-based sampling	6
2.3.2	Bayesian Active Learning by Disagreement	7
3	Travel purpose classification	7
3.1	Data	8
3.2	Feature weighting	9
3.3	Training and optimization	9
3.4	Results	10
4	Future work	10
4.1	Create a complete active-learning pipeline	12
4.2	Discovering classification labels bottom-up	12

Abstract

Statistics Netherlands (CBS) develops in-house machine learning models to extract information from complex surveys and sensor data, which requires a significant amount of high-quality labelled data. A resource-prohibitive factor in model development is the limited burden that can be placed on human annotators responsible for attributing pre-defined labels to the data. Moreover, on the conceptual side, it is not clear which labelling system is best suited for a given dataset. In this report we exploit online active learning to reduce the amount of data that needs to be labelled by annotators. We test two simple active learning pipelines for a CBS case study, where the goal is to accurately predict the purpose of travel stops in GPS travel data. The experimental results obtained during the SWI 2025 workshop demonstrate the promise of this approach. At the end of our report, we identify avenues for future work to fully develop an online active learning pipeline aimed at reducing annotator burden, as well as data-driven strategies to improve the labelling system.

KEYWORDS: SWI, CBS, active learning, data valuation

1 Introduction

Statistics Netherlands (Centraal Bureau voor de Statistiek, CBS) has the statutory task to compile statistics on a wide range of important topics for society and to make their outcomes publicly available. To this end, CBS gathers data from individuals and enterprises and processes these as statistics. To extract insights from complex survey and sensor data, CBS is increasingly using machine learning (ML) models. Since CBS emphasizes transparency and reproducibility, it has started developing and training its own tailor-made ML models. As training ML models requires a large amount of high-quality labelled data provided by human annotators, *annotator burden* quickly becomes a prohibitive factor. It is therefore vital to reduce this burden, both by reducing the number of annotators using effective and efficient sampling methods and by reducing the burden per human annotator. For the Study Group Mathematics with Industry 2025 (SWI2025), CBS submitted the general challenge to find new methods to reduce annotator burden and provided (data for) two concrete case studies: classifying purchases from scanned grocery shopping receipts and classifying the means of transportation and the purpose of travel stops in GPS travel data collected through a mobile app.

In this report we focus on the specific challenge of minimizing the labelling burden per human annotator in developing an ML model for classification. We consider a scenario where an ML model is trained using a small set of labelled training data and focus on minimizing the number of requested labels to update and improve the model while it is in use. We first cast this challenge as an *online active learning problem*, the problem of selecting the most informative data points to be labelled in a dynamic environment where data is continuously collected, and very briefly survey relevant active learning literature. Afterwards, we focus on the GPS case study and

aim to develop a classifier to predict the purpose of travel stops, assuming that raw GPS travel data has already been correctly segmented into travel sections and travel stops. We develop simple active learning pipelines for this goal featuring two different pairs of active sampling methods and ML classification methods: the first combines entropy sampling with a random forest classifier and the second pairs Bayesian Active Learning by Disagreement with a Bayesian neural network obtained via Monte Carlo Dropout. We conducted small-case experiments with the travel data collected by CBS during the five-day workshop. For one of the tested pipelines, our preliminary results show that active sampling methods allow to reach the same accuracy with significantly fewer labelled data compared to random sampling, providing a proof-of-concept of the power of active learning to reduce the number of labels requested from human annotators. At the end of our report, we identify avenues for future work that can be pursued to arrive at a more complete and sophisticated active learning pipeline and alternative ways to reduce annotator burden.

This report is organized as follows. In Section 2 we review the preliminaries on (online) active learning and the used active sampling strategies and classification methods. In Section 3 we consider the travel stop classification problem and present our experimental results. Finally, in Section 4 we discuss possible avenues for future work.

2 Methods

2.1 Classification methods

In this report, we consider two different methods for multi-class classification: Random Forests and Bayesian Neural Nets.

2.1.1 Random forests

Decision trees (see, e.g., Breiman et al. (2017)) are popular methods for classification and regression tasks that excel in interpretability. At same time, decision trees tend to overfit their training data. *Random Forest* (RF, Breiman (2001)) is an ensemble technique that aggregates the output of multiple decision trees to improve accuracy and reduce overfitting. To ensure diversity in the decision tree models, each tree is trained on a bootstrap sample of the data and in each split in the decision tree algorithm only a random subset of features is used. In a classification setting, the outputs of the trees are aggregated use a majority vote. Its ability to deal with missing or noisy data and handle large datasets make RF a widely used machine learning technique, see, e.g., Hastie et al. (2009).

2.1.2 Bayesian Neural Nets

Bayesian Neural Nets (BNNs) provide a framework for modelling uncertainty in neural network learning by learning probability distributions over the network weights

rather than point estimates (see, e.g., Blundell et al. (2015); Gal and Ghahramani (2016) for general references). They are also effective in handling limited data, mitigating the typical reliance of neural networks on large amounts of data. To achieve computational efficiency and for ease of implementation, we adopt the *Monte Carlo Dropout* (Gal and Ghahramani (2016)) approach as our Bayesian neural network framework. Originally introduced to reduce overfitting, dropout, when activated during both training and inference, makes each forward pass in the neural network stochastic. The article Gal and Ghahramani (2016) proposed that performing multiple forward passes through a stochastic network allows for Monte Carlo estimation of both the predictive mean and variance, quantifying model uncertainty. The authors introduced the term *MC-Dropout* and demonstrated that this can be theoretically interpreted as a variational Bayesian approximation, with each forward pass interpreted as drawing a sample from an approximate posterior distribution. It has since seen numerous applications for active learning, e.g., Tsymbalov et al. (2018) and Gal et al. (2017).

2.2 Active learning

Supervised learning methods require a significant amount of labelled training data in order to generalize well, i.e., to reach good predictive performance on new data. At the same time, there are many applications where it is expensive or cumbersome to label data, as it typically requires detailed inspection of the data and manual annotation. This scenario is common in, e.g., text classification, speech annotation, and protein structure prediction. *Active learning* is the subfield of machine learning that aims to reduce the number of samples that need to be labelled by developing strategies to identify the ‘most informative’ samples in a set of data. Traditionally, active learning has mostly focused on a *pool-based* scenario, where one selects and labels a set of samples from a static pool of data. In practice, one often faces a dynamic scenario—online scenario—where data arrives as a stream and one has to decide upon arrival whether or not to label a datum. In a single pass variant one has to decide whether or not to label a datum immediately as it arrives. Alternatively, in a *window-based* or batch-based variant, which is applicable to the GPS case study posed by CBS, arriving data can be collected in a limited buffer before data needs to be labelled.

Active learning has been studied for decades. There are several surveys, e.g., Settles (2009, 2012); Fu et al. (2013); Aggarwal et al. (2014); Kumar and Gupta (2020) on the traditional pool-based scenario. In addition, there are specialized surveys on deep active learning, which focus on active learning methods in the context of training deep learning models, e.g., Ren et al. (2021); Li et al. (2024), and the use of large language models to assist in sample selection and sample annotation, e.g., Xia et al. (2025). A good starting point for the literature on the online active learning scenario is the recent survey Cacciarelli and Kulahci (2024). Here we will only summarize some key concepts and aspects of active learning that are relevant for the CBS challenge.

2.2.1 Sampling strategy

Perhaps the most important component of an active learning method is the sampling or query strategy that is used to select instances to be labelled by a human annotator. In this work, we use two *uncertainty-based* sampling strategies to select data points for the which the model prediction is least confident according to a pre-specified uncertainty measure (Nguyen et al. (2022)), see Section 2.3 for details. Examples of other common sampling strategies are *diversity- and density-based* approaches, which aim to select instances that are diverse and representative of the data distribution, and *disagreement-based* strategies, which select data points where there is (the most) disagreement among multiple predictive models. We refer to Settles (2012); Monarch (2021); Cacciarelli and Kulahci (2024) for a more extensive discussion and overview of sampling strategies.

2.2.2 Timing of sampling

A second important aspect of an active learning method is the timing of labelling requests to human annotators. The labelling burden for human annotators is related not only to the amount of samples that are requested to be labelled, but also to the frequency and predictability of labelling requests. An excessive burden leads to a higher dropout of voluntary annotators and increased labelling fatigue, which may decrease the quality of the provided labels. Single pass active learning implicitly assumes online availability of a human annotator to flexibly respond to labelling requests, whereas window-based active learning requests annotators to label a batch of data with a pre-specified maximum amount of data to be labelled. The length of the time window needs to be balanced. A long time windows allows to collect more data for selection, although the quality of the provided labels may decay (especially where human memory is a limiting factor) and infrequent labelling requests risk annotator disengagement.

2.2.3 Model updates

Once new instances have been labelled, the predictive model needs to be updated. A simple approach, which is suitable for a window-based setting and is taken in this report, is to fully re-train the model using the updated set of labelled training data. While using active learning, one collects a growing set of labelled data over time and it is infeasible in practice to store an unlimited amount of data. Moreover, in many situations the data stream is not stationary, so that data and model drift will occur. For instance, when creating a system to predict the mode of transport used on segments of a person’s travel route, a faster e-bike may be introduced, creating different travel patterns. Hence, any annotated datum will generally become less relevant as time progresses. In these cases, it is of interest to use methods for data and model drift detection as part of the active learning process and to discard data in an informed manner. Although we do not pursue this direction, let us mention that one can view the problem of selecting data to discard as a data valuation problem,

see e.g., Sim et al. (2022); Ghorbani and Zou (2019), where one wishes to identify the most valuable elements in a training dataset.

2.2.4 Model evaluation

To gauge the effectiveness of an active learning strategy and to compare different strategies, one needs to fix a suitable evaluation metric. A popular approach, which we pursue here, is to generate learning curves that plot the prediction accuracy on a test set over the number of labelled examples used for training. It is standard to use uniform random sampling as a benchmark. In addition, non-parametric statistical tests can be used to compare the performance of different strategies, see e.g., the discussion in Reyes et al. (2018).

2.3 Uncertainty-based sampling

Let us now briefly introduce the two main sampling strategies that we explore in our active learning framework. The first strategy, entropy-based sampling, is explored in combination with RFs. The second, Bayesian Active Learning by Disagreement (BALD), is used together with BNNs.

2.3.1 Entropy-based sampling

Entropy-based sampling is an active learning strategy that can be used in a classification setting. In this strategy, data points for which the trained model is most unsure of its class are added to the training dataset. This uncertainty is quantified by the Shannon entropy. If the model is confident about its class prediction, the entropy is low, while if multiple classes have similar probabilities, the corresponding entropy is higher. By adding data points with high entropy to the dataset, the algorithm adds more points closer to the decision boundary. Hence, it results in the resolution of the decision boundary with fewer data points, so that the burden on data annotators is reduced significantly. However, such a model may be less stable and more sensitive to new data points.

To formally introduce entropy-based active sampling, let us consider a classification problem with c classes and let $\rho(\cdot|\cdot) : \{1, \dots, c\} \times \mathbf{R}^n \rightarrow [0, 1]$ denote the predictive distribution, i.e., $\rho(y|x)$ is the predicted probability that x belongs to class c . The Shannon entropy of the predictive distribution at x is defined by

$$H(\rho; x) := - \sum_{y=1}^c \rho(y|x) \log \rho(y|x),$$

Let us consider an initial training set \mathcal{S}_{ini} and a predictive distribution $\hat{\rho}$ fitted to this data (in our case study, $\hat{\rho}$ will be a random forest classifier). Let $\mathcal{D}_{\text{unl}} \subset \mathbf{R}^n$ be a batch of unlabelled inputs and let $b \in \mathbb{N}$ be the labelling budget. The idea of entropy sampling is to request labels for a set of b elements from $\mathcal{D}_{\text{unl}} \subset \mathbf{R}^n$ which have the

largest entropy. That is, if x_1^*, \dots, x_m^* are the elements of \mathcal{D}_{uni} ordered decreasingly according to their entropy, then the updated training set \mathcal{S}_{up} is

$$\mathcal{S}_{\text{up}} := \mathcal{S}_{\text{ini}} \cup \{x_1^*, \dots, x_b^*\}.$$

2.3.2 Bayesian Active Learning by Disagreement

In this sampling strategy, we utilize the predictive variance obtained from the stochastic outputs of our Bayesian neural network. The idea is that when the model typically exhibits high variance in the logits, it confidently predicts conflicting classes with high probabilities. Acquiring such points ideally should add "new" information which is less represented in the training data, thereby reducing model uncertainty. To this end, we follow the Bayesian Active Learning by Disagreement (BALD) sampling strategy Houthby et al. (2011), which prioritizes labelling data points which maximize mutual information between the predictions and the model's posterior. Concretely, for given input x this mutual information is equal to

$$\begin{aligned} \text{BALD}(\rho; x) &:= H\left(\mathbb{E}_{\theta \sim \rho(\cdot|x)}[\rho(\cdot|x, \theta)]\right) - \mathbb{E}_{\theta \sim \rho(\cdot|x)}\left[H(\rho(\cdot|x, \theta))\right] \\ &= -\sum_{y=1}^c \mathbb{E}_{\theta \sim \rho(\cdot|x)}(\rho(y|x, \theta)) \log(\mathbb{E}_{\theta \sim \rho(\cdot|x)}(\rho(y|x, \theta))) \\ &\quad + \mathbb{E}_{\theta \sim \rho(\cdot|x)}\left(\sum_{y=1}^c \rho(y|x, \theta) \log(\rho(y|x, \theta))\right) \end{aligned}$$

where $\rho(\cdot|x, \theta)$ is the predictive distribution given the input x and the parameters θ , and $\rho(\cdot|x)$ denotes the posterior distribution of the model parameters θ . In our case study below, $\rho(\cdot|x, \theta)$ is the softmax output of the Bayesian net with input x and MC dropout draw θ . Letting $\theta_1, \dots, \theta_m$ denote independent draws, we approximate the BALD score at x as

$$\begin{aligned} \text{BALD}(\rho; x) &\approx -\sum_{y=1}^c \frac{1}{m} \sum_{i=1}^m \rho(y|x, \theta_i) \log\left(\frac{1}{m} \sum_{i=1}^m \rho(y|x, \theta_i)\right) \\ &\quad + \frac{1}{m} \sum_{i=1}^m \left(\sum_{y=1}^c \rho(y|x, \theta_i) \log(\rho(y|x, \theta_i))\right). \end{aligned}$$

3 Travel purpose classification

In this section we describe the implementation of a simple active learning framework, depicted in Figure 1, in the travel purpose prediction case study.

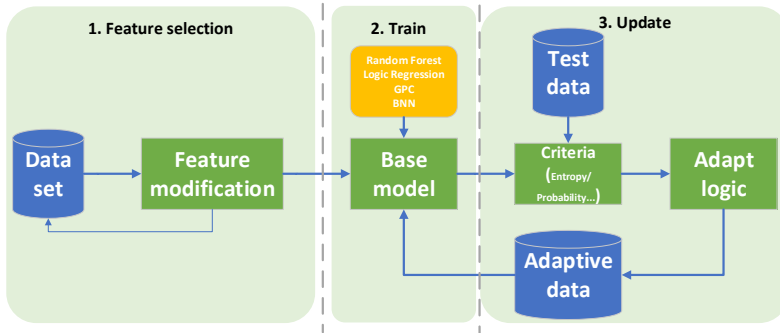


Figure 1: Active learning framework

3.1 Data

The data was collected by CBS from 59 users through a smartphone-based application (Smeets et al. (2019); McCool et al. (2021); Fourie (2025)). The app’s interface is shown in Figure 2. The collected dataset contains 4,961 observations and has 209 data features, such as stop duration, GPS location, and proximity to schools, shops, houses, and public transport stops, and 9 different labels for the travel stop purpose, such *thuis* (‘at home’) and *op visite* (‘visiting friends or relatives’). To better handle geospatial features, four distinct radii (25, 35, 50, and 200 meters) were given to evaluate discrepancies that may arise when a stop is located at a position that does not align with Open Street Map (OSM) data, for example, the feature `n_25_offices` indicates the number of offices located within 25 meters. Some (partial) instances of the collected dataset are shown in Table 1. The following problems are studied for

user_id	start_timestamp	end_timestamp	modality_purpose	duration	n_25_pois	...	n_50_offices
n	date 07:20	date 08:27	thuis	67.183333	0	...	0
n	date 09:05	date 09:30	visite	25.266667	1	...	0
n	date 10:21	date 11:54	thuis	1532.450000	0	...	0
n	date 12:09	date 15:16	visite	186.033333	1	...	0
n	date 15:39	date 06:18	thuis	879.250000	0	...	0

Table 1: An example of the clipped dataset.

this case:

1. Since the geospatial information of four distinct radii contains overlapping information, we study the reduction of overlapping features.
2. To reduce the annotation burden for users, we investigate whether active sampling can reduce the number of travel stops that need to be labelled to train an accurate model for the prediction of travel stop purpose.

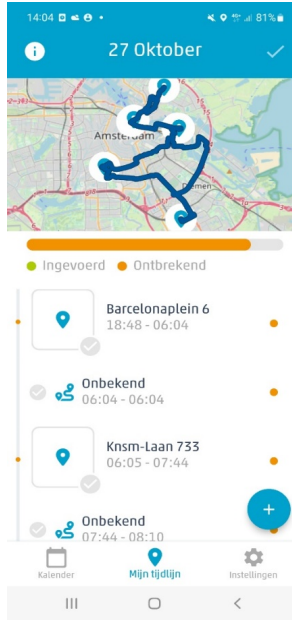


Figure 2: Interface of the mobile phone-based application.

3.2 Feature weighting

In the first step of our pipeline in Figure 1, the feature modification step, we combine GPS-features with overlapping information into a single feature. Let $GPS_{i,r}$ denote the GPS-based feature for radius $r \in \{25, 35, 50, 200\}$ and the i -th land type (such as schools, offices, supermarkets), e.g., $GPS_{school,25}$ is the number of schools within 25 meters of the stop location. For every i we combine the four features for this land type into the feature

$$GPS_i^* = \sum_r W_r 1_{GPS_{i,r} > 0}. \quad (1)$$

The indicator in Equation (1) converts $GPS_{i,r}$ into a binary value that encodes the presence (value 1) or absence (value 0) of the i -th land type within radius r of the stop location. Assuming that the proximity of a land type to the stop location increases its relevance, we set the weights equal to $W_r = 100/r$.

3.3 Training and optimization

After modifying the features as described in Section 3.2, we train two different classifiers using two different active sampling strategies: a random forest classifier (RF) using entropy sampling and a Bayesian Neural Network (BNN) using BALD. First, the total dataset is split into a test dataset (25% of the total dataset) and a training

dataset (remainder of the total dataset). The training dataset is then split into two datasets - (i) an initial training set (10% of the training data) and (ii) a set of data to evaluate the effectiveness of the active learning strategies, which we will call the *active sampling set*. Our evaluation is for a pool-based scenario, where we have the full active sampling set available when we select instances to label. In our evaluation, we actively select data in ten iterations. In each iteration we add a batch of identical size s . In the first iteration, we select the best s instances using our sampling method (entropy sampling in the case of RF, BALD in case of BNN) and add it to the training set. We then retrain using the updated training set. We then sample the best s instances from the remainder of the active sampling set, where we use the *updated* model to compute the uncertainty measure used for sampling. We continue in this fashion until the active sampling set is exhausted.

Let us now briefly specify the training of the RF and BNN classifiers. For the RF implementation, we utilized `scikit-learn`'s `RandomForestClassifier` with a standard configuration with 100 decision trees (`n_estimators = 100`) and the Gini impurity criterion for measuring the quality of splits. For the BNN, we implemented a neural network architecture in PyTorch with two hidden layers of 64 and 32 neurons respectively, using a dropout rate of 0.3 for MC-Dropout uncertainty estimation. The network was trained with the cross-entropy loss for 500 epochs using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. Since the amount of time available for implementation during the workshop was limited, we chose not to perform any hyperparameter tuning.

3.4 Results

Figure 3 shows the accuracy achieved by the random forest classifier on the test set as we add actively sampled batches of data to the training dataset. We show a comparison of the accuracy achieved by the entropy-based adaptive sampling strategy over randomly selected data. The RF trained with the entropy-based sampling strategy achieves the accuracy of the RF trained on the full dataset (i.e., the initial training set together with the full active sampling set) using only 30% of the adaptive sampling set. This shows that one can achieve the same accuracy with a significant reduction in the required number of data annotations.

Figure 4 shows the results for the BNN trained using the BALD sampling strategy. We observe that this combination does not outperform random sampling for our dataset. It may be possible to gain improved results by tuning the hyperparameters of the BNN and/or by using different active sampling methods.

4 Future work

In this report we formulated the problem of reducing human annotator burden in the CBS challenge as an active learning problem and conducted an experiment with a small, pool-based active learning pipeline as a proof of concept. Below we discuss

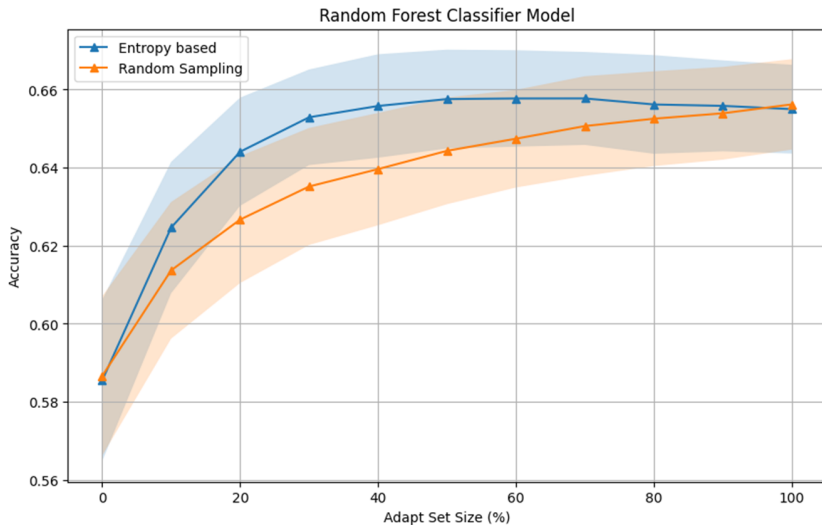


Figure 3: Accuracy of RF classifier under entropy-based sampling (blue) and under random sampling (yellow). The accuracy is plotted in terms of the percentage of the active sampling set added to the initial training data.

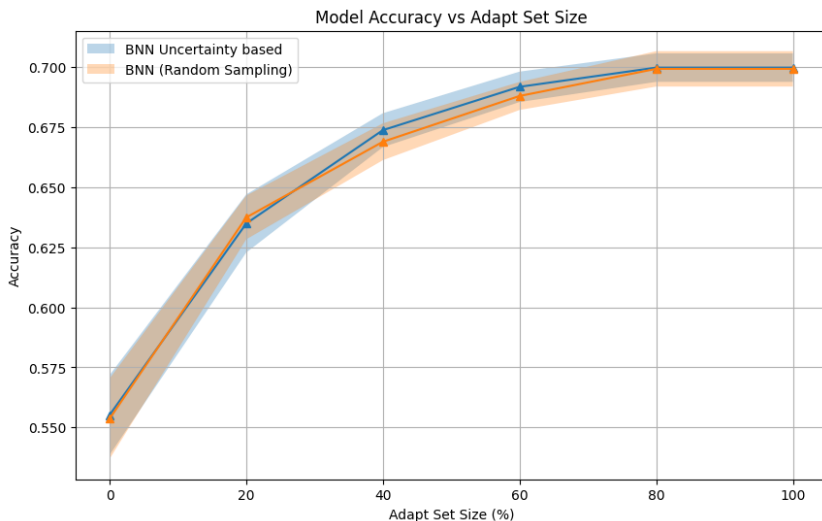


Figure 4: Accuracy of BNN classifier under BALD sampling (blue) and under random sampling (yellow). The accuracy is plotted in terms of the percentage of the active sampling set added to the initial training data.

two possible directions of research. First, we discuss the development of an online, window-based active learning pipeline with a non-stationary stream of data. Second, we consider the possibility to derive the system of labels itself from the data instead of defining it top-down.

4.1 Create a complete active-learning pipeline

A natural next step is to create a complete, online window-based active learning pipeline. Experiments with trial users can inform a suitable choice of window size and a maximum number of labelling requests per time window that ensures both continued user engagement and high-quality labelling. If the active learning pipeline is used over an extended period of time in which circumstances will change, then it is important to include both model and data drift detection methods in the pipeline. Due to data drift and the limited storage capacity for retaining labelled data, we recommend to include an informed data discarding strategy, which could combine existing methods from the literature on data valuation methods (see e.g., Sim et al. (2022); Ghorbani and Zou (2019)) with a mechanism that discounts the value of data with time, as one can expect that the accuracy of labelled data will diminish over time. Finally, one needs to decide on a strategy for model updates after collecting a batch of labelled data and after drift detection events. In this work we used complete retraining using the updated labelled training dataset. On the one hand, this may be computationally too expensive in an operational setting and one may need to resort to cheaper approximate updates (see, e.g., Cacciarelli and Kulahci (2024) for possible strategies). On the other hand, a drift detection event may warrant a more extensive model update that includes renewed hyperparameter tuning, which was excluded in our setup due to time constraints. Throughout, it needs to be ensured that the supervised learning and sampling methods in the pipeline are scalable, i.e., fit the size of the application and the available computational resources at CBS.

4.2 Discovering classification labels bottom-up

Addressing the data drift or implementing an automatic choice of the label set requires having a higher-order model. One possibility is to derive such a model from data points with ground truth labels by mapping them to a lower-dimensional space. For the purpose of demonstration, assume that:

- All data points are in \mathbb{R}^n .
- The labels of a small fraction of data points are known, we refer to them as the ground truth.
- The data points lie on a lower-dimensional manifold, e.g., they are more likely to share the same label when they are close to each other in some metric.

In principle, one can perform dimensionality reduction on the original data first and then use the low-dimensional space for label-free classification, such as k -means or

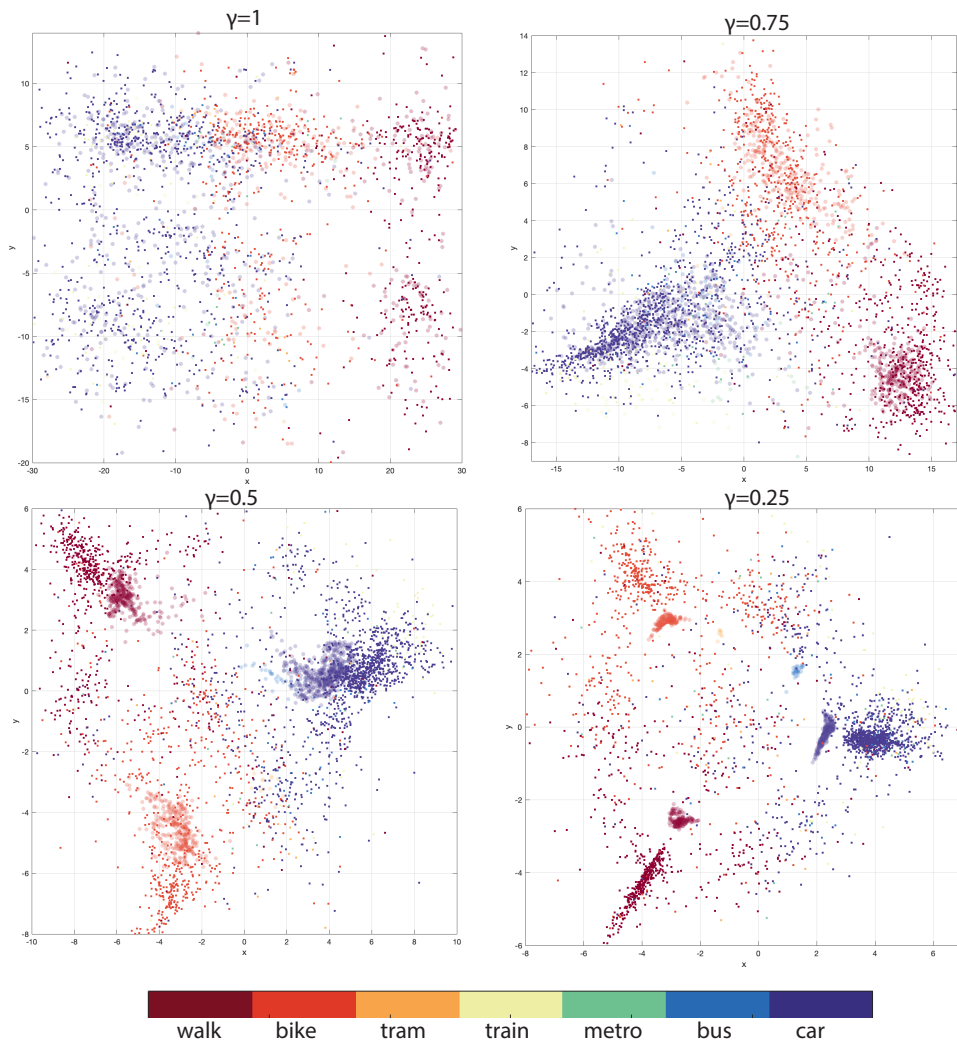


Figure 5: Embedding in 2D feature space with a metric that incorporates ground truth knowledge of 20% of labels. Values of γ are indicated above the panels. The ground truth data are large discs, non-ground truth are small boxes. One can observe that smaller values of γ improve separation of also non-ground truth data points.

spectral clustering, or for detecting data drift by extrapolating the positions of data points over time. However, in the presence of ground truth labels, we can improve this process by incorporating these labels into the distance function of the dimensionality

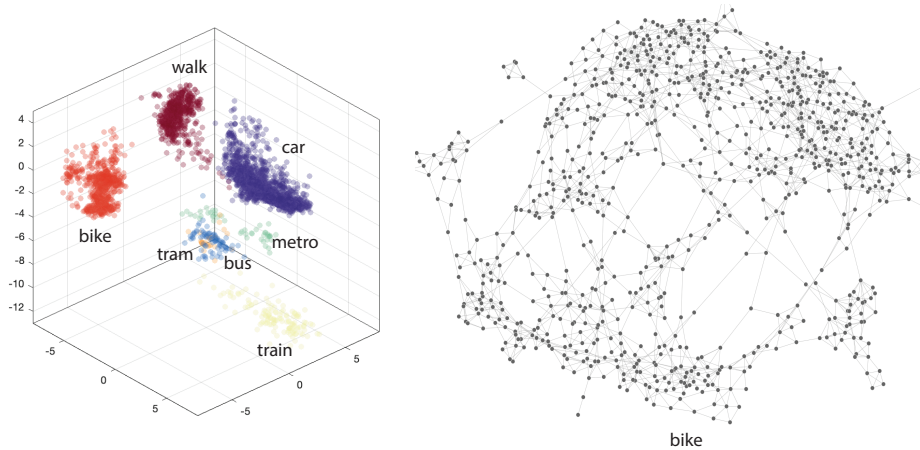


Figure 6: Embedding in 3D feature space with a metric that incorporates ground truth knowledge of all labels, $\gamma = 0.5$. *Left panel:* Clearly separated clusters feature non-trivial geometric structure as a group and on their own. *Right panel:* The structure of the network in the "Bike" cluster, with edges representing the nearest neighbours in the full space, also features heterogeneous structure, possibly leading to sub-labels.

reduction algorithm. For example, one possible choice of the distance function is as follows,

$$\text{dist}(x, y) = \begin{cases} \gamma \|x - y\| & \text{if } \text{class}(x) = \text{class}(y), \\ \gamma^{-1} \|x - y\| & \text{if } \text{class}(x) \neq \text{class}(y), \end{cases}$$

for $\gamma \in (0, 1]$.

In the example below, we used the “Means of Transport” dataset, focusing on real-valued variables such as the average speed, acceleration, *etc.* This dataset consists of 45 unique variables, which we mapped to a two-dimensional feature space using the Isomap algorithm. Around 20% of the data was randomly chosen as the ground truth, with classes proportionally represented.

Depending on what is the fraction of data points included into the ground truth set, we may model different scenarios. Incorporating a small fraction of ground truth data into the dimensionality reduction algorithm qualitatively improves the separation of classes. Figure 5 illustrates this the idea for different values of the parameter γ . Including a large proportion of data into the ground truth, allows to iteratively classify only the newly arrived data. Moreover, even if we include all of the points into the ground truth set, the embedding will not help to classify data, but one still obtains well-separated clusters of data points with a clear geometric structure, as shown in Figure 6, left panel. These clusters can be further studied for classification

into sub labels or for detecting trends. Note, that as byproduct of distance-based dimensionality reduction algorithms, one obtains a network of data points that are close in the feature space, see Figure 6, right panel. This allows to study the data with network based algorithms.

References

- Charu C Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S Yu. Active learning: A survey. In *Data classification*, pages 599–634. Chapman and Hall/CRC, 2014.
- Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.
- Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
- Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.
- Davide Cacciarelli and Murat Kulahci. Active learning for data streams: a survey. *Mach. Learn.*, 113(1):185–239, 2024. ISSN 0885-6125. doi: 10.1007/s10994-023-06454-2. URL <https://doi.org/10.1007/s10994-023-06454-2>.
- Johannes Jurgens Fourie. Rules for transport mode determination in smart travel surveys. Master’s thesis, Leiden University, 2025.
- Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and information systems*, 35:249–283, 2013.
- Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR, 2017.
- Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer, 2009.
- Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

- Punit Kumar and Atul Gupta. Active learning query strategies for classification, regression, and clustering: A survey. *Journal of Computer Science and Technology*, 35:913–945, 2020.
- Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. A survey on deep active learning: Recent advances and new frontiers. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2024. doi: 10.1109/TNNLS.2024.3396463.
- Danielle McCool, Peter Lugtig, Ole Mussmann, and Barry Schouten. An app-assisted travel survey in official statistics: Possibilities and challenges. *Journal of Official Statistics*, 37(1):149–170, 2021. doi: 10.2478/jos-2021-0007. URL <https://journals.sagepub.com/doi/abs/10.2478/jos-2021-0007>.
- Robert Munro Monarch. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Manning, 2021.
- Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1): 89–122, 2022.
- Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Comput. Surv.*, 54(9), October 2021. ISSN 0360-0300. doi: 10.1145/3472291. URL <https://doi.org/10.1145/3472291>.
- Oscar Reyes, Abdulrahman H Altalhi, and Sebastián Ventura. Statistical comparisons of active learning strategies over multiple datasets. *Knowledge-Based Systems*, 145: 274–288, 2018.
- Burr Settles. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences, Technical report TR1648*, 2009.
- Burr Settles. *Active Learning*. Springer Cham, 2012.
- Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning: "ingredients", strategies, and open challenges. In *IJCAI*, pages 5607–5614, 2022.
- Laurent Smeets, P.J. Lugtig, and Barry Schouten. *Automatic travel mode prediction in a National Travel survey: CBS Discussion paper*. Statistics Netherlands, December 2019.
- Evgenii Tsymbalov, Maxim Panov, and Alexander Shapeev. Dropout-based active learning for regression. In *Analysis of Images, Social Networks and Texts: 7th International Conference, AIST 2018, Moscow, Russia, July 5–7, 2018, Revised Selected Papers 7*, pages 247–258. Springer, 2018.

Yu Xia, Subhojyoti Mukherjee, Zhouhang Xie, Junda Wu, Xintong Li, Ryan Aponte, Hanjia Lyu, Joe Barrow, Hongjie Chen, Franck Deroncourt, et al. From selection to generation: A survey of llm-based active learning. *arXiv preprint arXiv:2502.11767*, 2025.