VECTOR INSTITUTE

Understanding Dataset Shift and Potential Remedies

A Vector Institute Industry Collaborative Project

Technical Report

Authors (in alphabetical order)

Vector Project Team

Mehdi Ataei[1], Murat Erdogdu[1], Sedef Akinli Kocak[1], Shai Ben-David[1], Shems Saleh[1]

Focus Area 1: Cross Sectional

Ali Pesaranghader[2] (Focus Area Lead), Andrew Alberts-Scherer[7], George Sanchez[3], Saeed Pouryazdian[6]

Focus Area 2: Time Series

Ahmad Ghazi[4] (Focus Area Lead), Jennifer Nguyen[5], Karim Khayrat[8]

Focus Area 3: Computer Vision

Bo Zhao[8]

[1]*Vector Institute,* [2]*CIBC ,* [3]*Thomson Reuters,* [4]*PwC,* [5]*Sun Life,* [6]*Manulife,* [7]*Intact,* [8]*BMO*

EXECUTIVE SUMMARY

Machine learning (ML) systems are trained under the premise that training data and real-world data will have similar distribution patterns. However, in dynamic industries and changing circumstances, new data distribution patterns can emerge that differ significantly from the historical patterns used for training-so much so that they have a major impact on the reliability of predictions. This difference between training and test data is known as dataset shift, and, when severe enough, necessitates adaptation. This adaptation can be accomplished either through cumbersome and expensive model retraining or leaner and more focused dataset shift adaptation techniques.

In May 2020, the Vector Institute launched the Dataset Shift Project, an industry-academia collaboration established to equip Vector's industry sponsor companies with a deeper understanding of dataset shift and its various types along with detection strategies and adaptation techniques. The project involved 15 participants: five Vector researchers and staff with expertise in ML as well as 10 technical professionals from seven Vector industry sponsor companies. It included four hands-on tutorials, developed and facilitated by Vector researchers and staff, in which participants improved their knowledge and skills through experiential learning.

The project covered three types of dataset shift:

- **Covariate Shift:** A difference in the distribution of input variables between training data and test data. Covariate shift can occur due to a lack of randomness, inadequate sampling, biased sampling, or a non-stationary environment.
- **Label Shift:** A difference in the distribution of class variables (i.e. classification results) between training data and test output. Label shift may appear when some concepts are undersampled or oversampled in the target domain compared to the source domain.
- **Concept Shift:** A difference in the relationship between the two variables used in the development of an algorithm.

In three working groups, participants investigated dataset shift in **cross-sectional, time series**, and **image data**. These dataset types aligned with participants' interests, and reflected real, current application potential in their organizations. The following summarizes the objectives and results of each working group.

Cross Sectional Data: The purpose of this study was to detect *covariate shift* in cross-sectional data, and to adapt algorithms and techniques to account for it. The group used the Iowa House Sales Prices dataset from Kaggle, and were tasked with predicting house sale prices using data from years 2006 to 2010. The main steps and objectives of this working group were to prepare cross-sectional data for experiments, apply dataset shift analysis algorithms, identify potential shifts, use shift adaptation techniques, and analyze the resulting prediction model. The group demonstrated that adaptation does not necessarily improve the performance results in all cases, implying that even the best adaptation models and transformations cannot be generally applied in different use cases.

Time Series Data: The purpose of this study was to use transfer learning and adaptive learning as a means to tackle dataset shift in retail-specifically, to estimate the sales of new goods using the data distribution patterns of current or past sales. To do this, the group used the Predict Future Sales dataset from Kaggle, which consists of historical retail-item sales data from January 2013 to October 2015. The group took two approaches:

- One approach investigated the use of transfer learning for Long Short-Term Memory networks in order to leverage the learned knowledge from one sale item and transfer it to another item with limited data. Successfully reusing previously-learned knowledge would eliminate the need to train a model from scratch, which is especially important when data is scarce and expensive to obtain.
- The other approach involved applying adaptive learning methods, which monitor the performance of the model and update its coefficients if performance deteriorates. Adaptive learning methods were used to correct potential *concept shift* in the data, as they are known to be robust against *concept shift* in dynamic environments. This approach was particularly relevant in the context of the COVID-19 pandemic, as the large and sudden shift in human behavior it caused rendered some predictive models inaccurate due to *concept shift*.

The group demonstrated that adaptive methods outperform non-adaptive methods when concept shift is present, and that results are comparable to when there is no concept shift. To better understand the effectiveness of adaptive methods, various models must be tested. The group also concluded that applying transfer learning to a new model can enhance its prediction capabilities, accelerate training, and reduce the cost of retraining a model when limited data is available.

Image Data: The purpose of this study was to use few-shot learning methods – methods using very limited training examples – to classify new data. The group's objectives were to a) reproduce the results of prototypical networks trained on the Omniglot dataset and the mini-ImageNet dataset separately, and b) reproduce the results of model-agnostic meta-learning algorithms trained on the Omniglot dataset.

The working group demonstrated that prototypical networks can tackle dataset shift using few-shot learning on the Omniglot and mini-ImageNet datasets. However, performance dropped significantly when running prototypical networks on different combinations of datasets-for instance, when training networks on the Omniglot dataset and then testing on the mini-ImageNet dataset. The group also demonstrated that model-agnostic meta-learning algorithms could tackle dataset shift when trained on the Omniglot dataset.

Overall, the Dataset Shift Project resulted in significant knowledge transfer between Vector researchers and industry participants. Industry participants developed proficiency in dataset shift detection, identification, and adaptation methodologies, established best practices in accordance with the latest academic and industry standards, and gained skills that can increase the resilience of organizations and their workforces in the face of changing environments. If put into production, these approaches have the potential to deliver enhanced efficiency, adaptability, and cost-savings. Finally, this project also demonstrated the value of collaborative efforts between industry and academia, and laid the groundwork for future projects focused on building a deeper understanding of dataset shift and methods for mitigating its effects in practical settings.

CONTENTS

1 INTRODUCTION

The Vector Institute (Vector) launched an industrial-academic collaborative project on dataset shift to tackle the challenge of dataset shift, i.e. that AI models being used in industry are trained on historical data, but applied on recent data which can have a different distribution. This project was inspired by the recent COVID-19 pandemic which has caused behavioural and economic patterns to shift so drastically that there is limited data indicative of current economic and social conditions. The phenomenon is being referred to as "dataset shift" when the data used to train models differs significantly from the data the model will see once it is deployed [1].

Developing adaptive methods to address dataset shift is an open problem in the field of ML, as it can present significant barriers to the real-world deployment of ML tools. The problem of dataset shift can stem from the way input features are utilized, the way training and test sets are selected, data sparsity, shifts in the data distribution due to non-stationary environments.

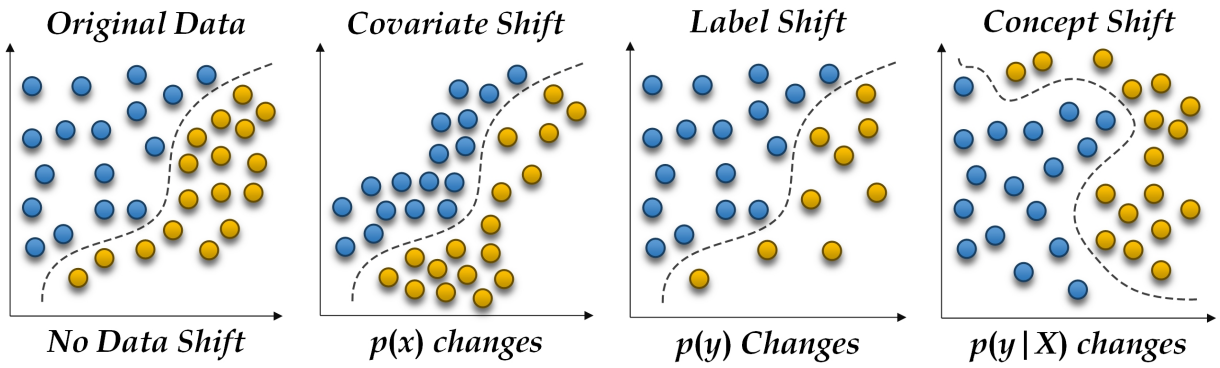
In the real world, the conditions in which we use the systems we develop will differ from the conditions in which they were developed. Typically, environments are dynamic, and sometimes the difficulties of matching the development scenario to the use are too great or too costly. It is critical to develop an understanding of the appropriateness of particular models in the circumstance of changes. Knowledge of how best to model the potential changes will enable better representation of the result of these changes. Dataset shift deals with the business of relating information in (usually) two closely related environments to help with the prediction in one given the data in the other(s).

The Vector project is intended to help its industry sponsors prepare their workforce for this new reality and ensure they have the skills and tools to be more resilient as the world continues to evolve. This report provides an overview of the collaboration between the Vector Institute and some of its industrial partners in that project.

1.1 BACKGROUND

ML systems are trained under the premise that the training and the real-world data (both inputs and outputs) have similar data distribution. This assumption can result in predictive problems in ML systems used in industries such as in retail where data is always evolving and constantly affected by consumer behaviour, resulting in change in data distribution, and hence unreliable predictions. The presence of such a discrepancy between distributions of datasets is called "dataset shift" [1, 2, 3]. Dataset shift is a common phenomenon which affects ML systems to different extents based on the amount of shift between the training and real-world distributions. This shift can vary from a small distributional shift to a bigger shift, such as that expected of a pandemic, such as COVID-19 outbreak in 2020. A pandemic shift has the potential to significantly alter the joint distribution of data and thus reduce the accuracy of models trained on data collected prior to the pandemic, with historically different distributions.

Dataset shifts are mainly categorized into three main groups: covariate shift, label shift, and concept shift [1]. These shifts could also co-occur or one lead to another. The covariate, label, and concept shifts are illustrated in Fig. 1 to let the reader conceptualize this phenomenon before explaining them.



(a) Original data (b) Covariate shift (c) Label shift (d) Concept shift

Figure 1: Dataset shift illustration (Similar to Fig. 1 in [4, 5].) Note that the dotted line is the decision boundary between the two classes; i.e., the blue and yellow data points. Here, x represents the input data and y represents the output we aim to predict.

As shown in Fig. 1, theoretically, there's no change in the decision boundary in the case case of covariate and label shifts; however, covariate shift represents a shift in the input distribution while label shift is a shift in the output distribution. On the other hand, concept shift represents a change in the relationship between the input and output therefore changing the decision boundary and learning algorithm needs to be refit from scratch.

In the following sections the **source** and the **target** domains are denoted by S and T respectively. P_S and P_T are probability distributions over data source S and T respectively. Source in this definition refers to input data/training data, while target refers to output data that the algorithm will see once deployed and which might be shifted. Domain in this case refers to a change in environment.

Covariate Shift: Covariate shift happens when the conditional distribution $P_S(y|x)$ remains the same, i.e., that conditional distribution of the source and target domains are equal, but $P_S(x)$ changes [1]. So, we have:

$$P_S(x)P_S(y|x) \neq P_T(x)P_T(y|x)$$

where

$$P_S(y|x) = P_T(y|x)$$

Covariate shift appears in data due to lack of randomness, inadequate sampling, biased sampling, and non-stationary environment.

Label Shift: Label shift is experienced when the conditional distribution $P_S(x|y)$ remains the same but $P_S(y)$ changes [1]. So, we have:

$$P_S(y)P_S(x|y) \neq P_T(y)P_T(x|y)$$

where

$$P_S(x|y) = P_T(x|y)$$

$P_S(y) \neq P_T(y)$ implies that label shift happens when some concepts are undersampled or oversampled in the target domain compared to the source domain.

Concept Shift: As to concept shift, $P_S(y)$ and $P_T(y)$ follow the same distribution but $P_S(y|x)$ differs from $P_T(y|x)$ [4]. To address concept shift, model needs to be adopted *globally* or *locally*. Global adaptation is training the model from scratch using the target data whereas local adaptation works for learning algorithms that can be refitted for some part of their decision regions; for example consider decision trees where some branches may be updated to reflect the change in the real world. Concept shift detectors compare the performance of a learner against both the source and target data; and if there is a significant difference they alarm for a drift; HDDM [6] and FHDDM [7] are examples of such detectors.

Finally, it is worth noting that *domain shift* [8]is another type that may be introduced by changes in the measurement system or how the data is described.

1.2 PROJECT OVERVIEW

This is a joint academic-industrial collaborative project launched in May 2020 to explore dataset shift in an applied setting with the supervision and expertise of Vector researchers. The purpose is to understand the problem better and explore current potential remedies. The project involved 15 participants: Five Vector researchers and staff with expertise in ML along with ten industry technical professionals from seven Vector sponsor companies. The project was conducted over seven months. During the project, weekly meetings were held as ways of communicating current updates and tasks among project members. Commonly found group activities in the weekly meetings were problem solving, decision making, prioritization, and task assignment. Weekly meetings also featured hands-on tutorials and invited guest talks on recent advances in this domain from academia. The participants established three working groups, each of which developed and performed experiments related to different shift detection and adaptation techniques. The primary objectives of the project were:

- Foster and widen productive collaboration among academic researchers and industry practitioners on projects related to dataset shift.
- To help participants explore mechanisms to understand, detect and potentially implement potential solutions for various dataset shift problems in prediction and classification using open source datasets.
- To help industry sponsors prepare their workforce for changing environments and adopt to new realities, and ensure they have the skills and tools to be more resilient as the world continues to evolve.
- To support industry sponsors with potential cost-savings if implemented in production in sponsor businesses.

Three main project focus areas arose which reflected current industry needs, participants' dataset interests, and expertise: (1) cross sectional, (2) time series, and (3) image. The remainder of this report provides a high-level overview of these focus areas and brief summaries of the working group's activities and sub-projects in each area.

2 FOCUS AREA ONE: CROSS SECTIONAL

2.1 OBJECTIVES

In this study the group focused on covariate shift detection and adaptation in typical cross-sectional data. The main steps and objectives of this working group are to prepare cross-sectional data for their experiments, apply shift analysis algorithms, identify potential dataset shifts, and use shift adaptation techniques and analyze whether they result in a better prediction model. This work consists of five steps as illustrated by Fig. 2:

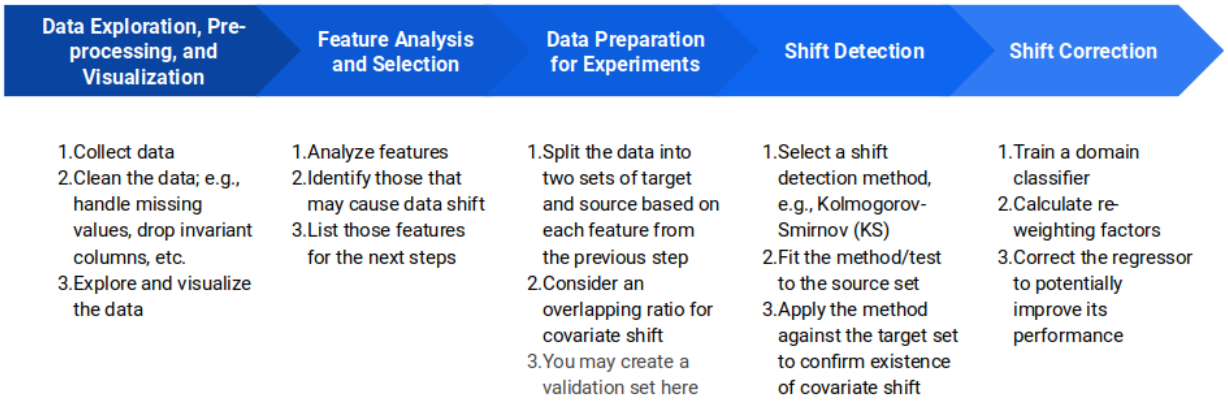


Figure 2: Shift detection and adaptation methodology

2.2 METHODS

The group used the Iowa House Sales Prices dataset from Kaggle[9]. The task was to predict house sales prices in Iowa in the given dataset the year from 2006 to 2010. The features that this dataset includes Lot Area, Neighborhood, Garage Type, Over Quality, Year Built, Kitchen Quality, Fireplace, etc.

There are 1460 training records with 80 features. Handling missing values was achieved by dropping columns with more than 60% missing values, and replacing the missing values with the mean of the respective column for numerical features, and the most frequent value in the column for categorical features. The group used *One Hot Encoding*, *Helmert Encoding*, and *Mean Encoding* to convert categorical variables into numerical values, so that they can be processed by ML algorithms alongside the numerical features [10]. Assuming that features with small variances most probably would not affect the target value prediction, the features with small variance were dropped. And finally, the features were normalized using a MinMax scaler to prevent algorithms from being ill-conditioned.

The group did a feature analysis to explore the importance of each feature in characterizing the

house prices. Then, each member of the group chose a feature to analyze whether the feature leads to data shift. SHAP values [11] after fitting the data to a Catboost regression model is shown below that can handle both categorical and numerical features. With the feature analysis results at hand, the group had a list of potential features that could be used to study for data shift.

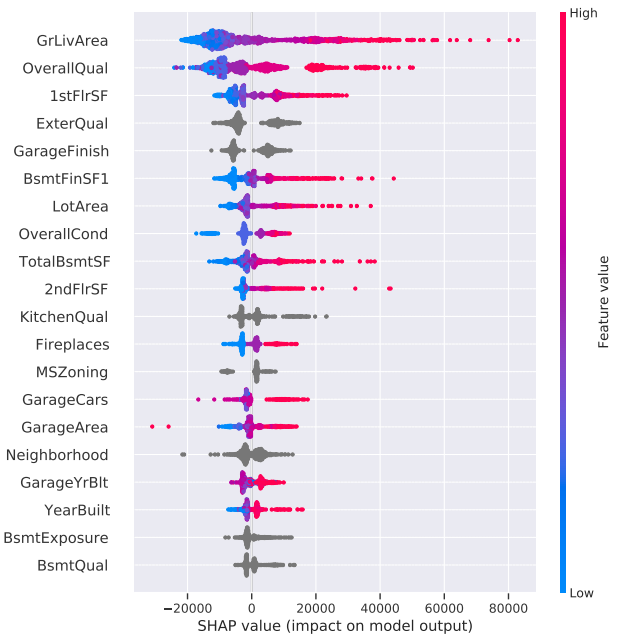


Figure 3: The top important features based on SHAP values

Based on the feature analysis the group chose to conduct the studies on three different attributes (1) yearBuilt, (2) Neighbourhood, and (3) Overall Quality.

Given the source and target samples, the task was to determine whether those samples were drawn from the same distributions. Here, the group attempted to detect covariate shift by explicitly training a *domain classifier* to discriminate between data from source and target domains [12]. To this end, the group partitioned both the source data and target data into two halves, using the first to train a domain classifier to distinguish source (class 0) from target (class 1) data. The group then applied this model to the second half and subsequently conduct a significance test to determine if the classifier's performance is statistically different from random chance. In addition, the group used Multiple Univariate Testing to detect dataset shift by adopting the Kolmogorov-Smirnov (KS) [13] with Bonferroni correction [14], which is a conservative aggregation method that rejects the null hypothesis if the minimum p-value among all tests is less than the significance level of the test.

In order to correct the covariate shift, the loss function of the regression algorithm was modified by applying sample reweighting [15]. To obtain the sample weights, the distribution ratios $\beta = P_T(X)/P_S(X)$ needed to be estimated. In order to estimate β, the group trained a domain classifier and calculated β using the predicted probability of each class. This could be achieved by assuming that each source data point has a non-zero probability of being in the test data; otherwise β would explode. It is therefore necessary to ensure that there is an overlap between source and target distributions when the data is split.

The group compared the performance of the shift adaptation across various regression models. They considered both bagging and boosting techniques in the analysis for tree-based regression models and also applied a neural network based regression model for comparison. Ensemble Trees (Random Forest, and XGBoost) and Neural Networks were used.

2.3 RESULTS AND DISCUSSION

2.3.1 PART ONE: ATTRIBUTE YEAR BUILT

The group split the Kaggle Housing dataset over the feature YearBuilt, which is an integer variable indicating the year the house was built, for the purpose of the first study. They define the source and target data as follows:

- **Source:** Houses that were built before or during the year 2000.
- **Target:** Houses that were built during or after the year 2001.

The intuition behind this split is that the group would expect a covariate shift in house features between the 20th and 21st centuries, for example, because of differences in demographics and building technologies. They confirm this covariate shift with the Kolmogorov-Smirnov test, along with simple visual inspection of the feature distributions. As an example, Figure 4 plot shows the GrLivArea variable for both source and target data:

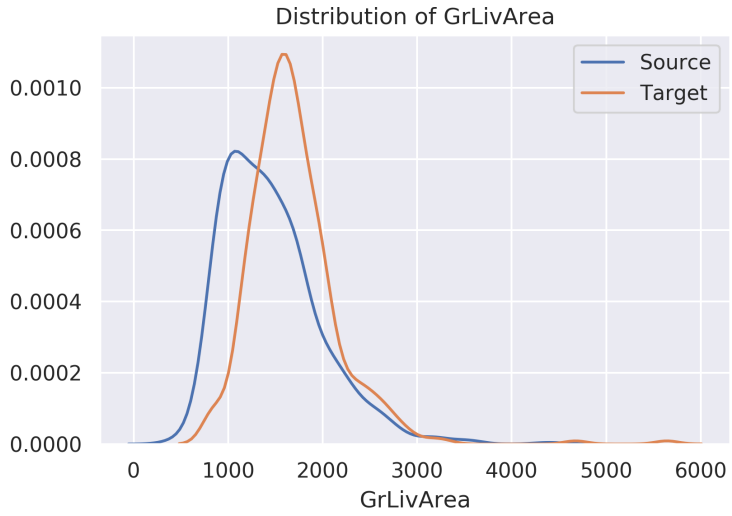


Figure 4: Distribution of source and target based on the GrLivArea attribute

Since the task was to evaluate the performance of a predictive model, the group also held-out 20% of the target data to have a set of data unseen from our training process to properly evaluate the different models they trained. Thus, the structure of the data sets looks like Figure 5:

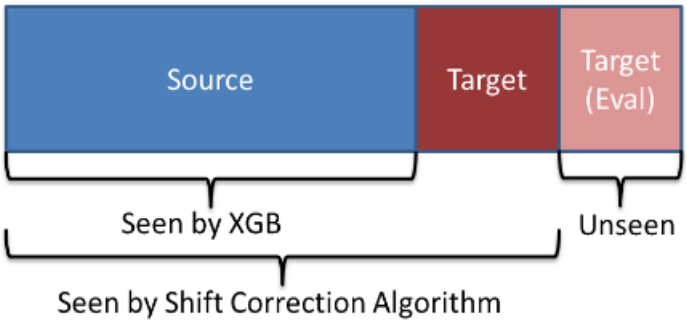


Figure 5: Source and target sets and their overlaps

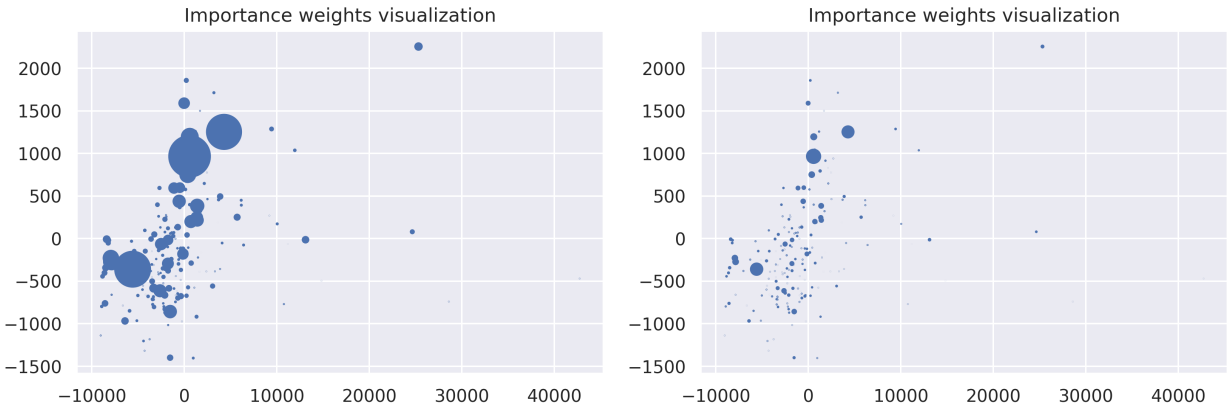
With the aforementioned datasets, the prediction task was to train an XGBoost regressor on the training set to predict the sale price of each house. Evaluation was done on the held-out data from the target data.

The experimental aspect of this is to use the source and non-held-out target data to compute covariate-shift adaptation weights, i.e., reweighting factors using different domain classifiers and transformation functions of the weights. The experiments were run 1,000 times, where the held-out set is randomly sampled each time. This allows the group to compute the mean standard deviation of the RMSE and R-squared metrics as shown in Table 1 for each attempted variation.

	Domain Classifier	Avg. ROC-AUC	Transform on β	Avg. RMSE	Avg. R-Squared
1	None	NA	NA	55,326 (12,776)	0.535 (0.222)
2	Logistic	0.962 (0.002)	None	55,694 (13,602)	0.527 (0.239)
3	Logistic	0.962 (0.002)	Min/Max Normalization	46,345 (8,873)	0.679 (0.121)
4	Random Forest	1.0 (2.83e-17)	None	54,129 (17,822)	0.521 (0.346)
5	Random Forest	1.0 (2.83e-17)	Min/Max Normalization	57,141 (17,547)	0.479 (0.345)
6	Random Forest	1.0 (2.83e-17)	Multiplied (4) by the betas from (2)	49,571 (12,928)	0.614 (0.228)
7	Random Forest (Max depth=5)	0.976 (0.001)	None	56,443 (15,790)	0.505 (0.294)
8	Random Forest (Max depth=5)	0.976 (0.001)	Min/Max Normalization	56,876 (17,135)	0.484 (0.342)
9	Random Forest (Max depth=5)	0.976 (0.001)	Multiplied (7) by the betas from (4)	52,054 (16,122)	0.558 (0.311)

Table 1: Performance of different classifiers with varying transformations on β. Variances are shown in parenthesis

Note that the first entry in the table does not use a domain adaptation method. Instead, it is used as a baseline. It can be seen in the Table 11 that a logistic regression domain classifier along with min/max normalization performs the best on the hold out set in terms and RMSE and R-squared, and also has the lowest variance in those measures. It is interesting to note that without applying min/max normalization (row 2), the predictive model performs worse than without any domain adaptation at all (row 1). To get an intuition as to why this is, the weights on the training data in a lower-dimensional space (using PCA) is visualized by Fig. 6. Without normalization, it appears that the predictive model is applying an excessive amount of weight on certain samples, leading to poor performance overall.



(a) Without normalization

(b) With normalization

Figure 6: Data points visualization after applying re-weighting factors

When using the random forest domain classifier, it is striking that despite its high performance in terms of ROC-AUC, it does not outperform the logistic regression on average and even underperforms the baseline model on occasion. During experimentation, the random forest did perform well for certain random splits of the target data, but these results were not always reproducible as can be seen by the high variance in the RMSE and R-squared values.

For experiment (9), it is worth noting that the multiplication of the weights from two different domain classifiers outperformed the use of the two domain classifiers independently. One hypothetical reason is that the individual classifiers discovered different traits of covariate shift, so their multiplication leads to a correction of each one's respective areas of underperformance.

2.3.2 PART TWO: ATTRIBUTE NEIGHBOURHOOD

For the second analysis, the group considered the neighborhood feature for creating source and target sets. The data are split as follows:

- **Source:** The College Cr. neighbourhood is used to create our source data.
- **Target:** The Old Town neighbourhood is used to create our target data.

A regressive learner was set up using neural networks to predict the house prices split up by the neighborhood feature. Two types of estimators were considered: 1. Simple Linear Regression, and 2. Multi-Layer Perceptron (MLP) with ReLU activation function. The re-weighting factors, i.e, the ratio of the distribution probabilities of the source and target data determined by a domain classifier, were applied to aforementioned learners. Table 2 shows the results of the neighbourhoods experiments.

Learning Algorithm	No adaptation Squared Error Loss	With adaptation Squared Error Loss
Linear Num. of Epochs: 10 SGD + Momentum: Learning rate: 0.0001 Momentum: 0.9	0.0535	0.0018
MLP - Relu Num. of Epochs: 50 SGD + Momentum: Learning rate: 0.0001 Momentum: 0.9	1.0169	0.0963

Table 2: Squared error loss of each learning algorithm with and without adaptation

Fig. 7 and 8 show the house price prediction results for Linear and MLP estimators. In both figures, the source data distribution is shown in red, target data distribution in green, uncorrected prediction distribution in blue, and corrected prediction distribution in yellow.

Applying sample weights in the loss function improves predicted estimates towards the target data distribution. This was supported by the lower loss values for both of the learners when applying the adaptation. Since the source and target are from different distributions, the learners were trained using the source data. Using the learner to predict the data points from the target distribution imposes an undesirable high variance because the new data points may have never sampled from the source distribution. The adaptation acts as a smoothing factor that reduces the variance and imposes bias towards impactful samples. That bias points towards the ratio of the probabilities of the betas.

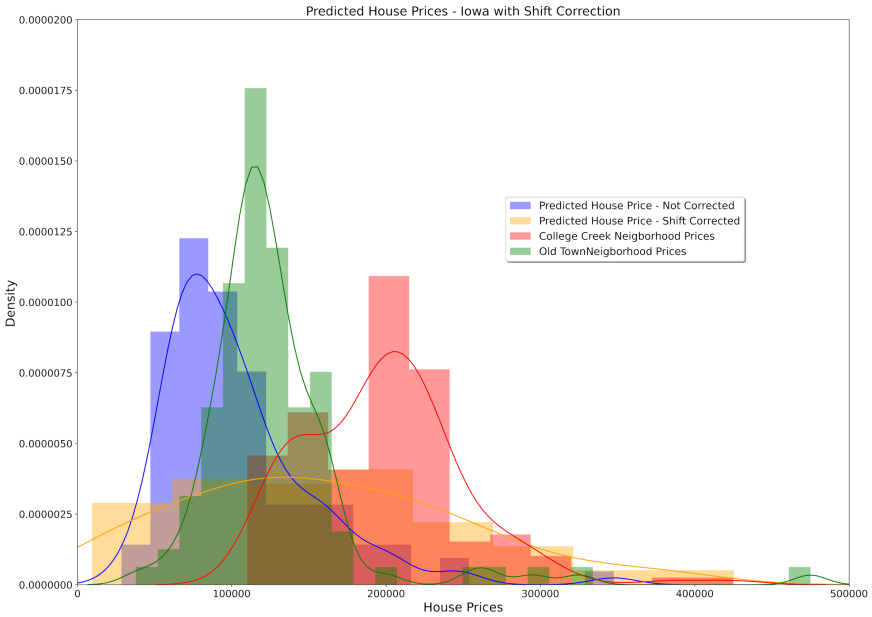


Figure 7: Data and prediction distributions for a Linear Model

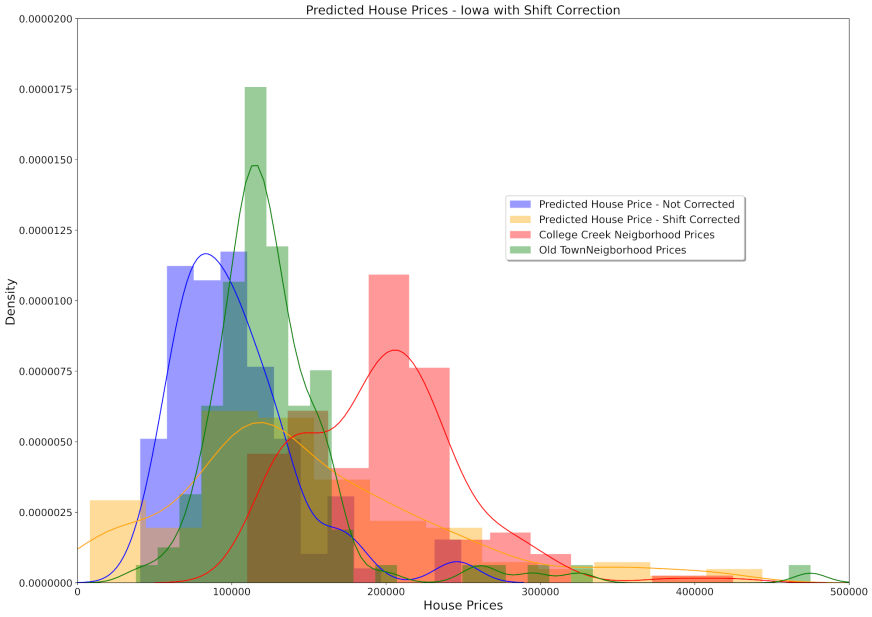


Figure 8: Data and prediction distributions for a Multi-Layer Perceptron (MLP)

2.3.3 PART THREE: OVERALL QUALITY ATTRIBUTE

For the third analysis, the group considered the overall quality feature, which is an ordinal feature that rates the overall material and finish of the house. Figure 9 shows the histogram of the OveralQual feature in the dataset with mean value of 6.

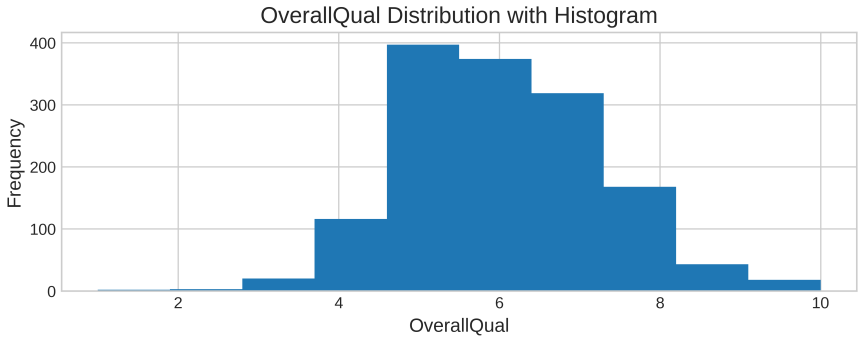


Figure 9: Histogram of OverlQual feature

In order to create the source and target sets, the data were split based on mean of the feature as follows:

- **Source:** records with OverllQual less than mean is split into source data.
- **Target:** records with OverllQual greater than mean is split into target data. Target data were split further into Validation set and Target, where the validation set was used for sample weights estimation.

The source and target data were shuffled later to create different percentages of overlaps between distributions.

Fig. 10 shows the resulting distributions of source and target data after the split. It can clearly be seen that there is a shift between source and target data. The house sales price distribution of the both source and target records is also depicted. Mean Encoding and Helmert encoding of the categorical variables were applied in the analysis, while mean encoding gave better results.

Random Forest and XGBoost were used as the domain classifiers in this part with the AUC score being 80.6345. Important predictors in the domain classifier that characterized the shift were removed before estimating the reweighting factors. Kolmogorov-Smirnov Analysis [13] also confirmed that statistically there is a shift in the data based on experimental settings in this part.

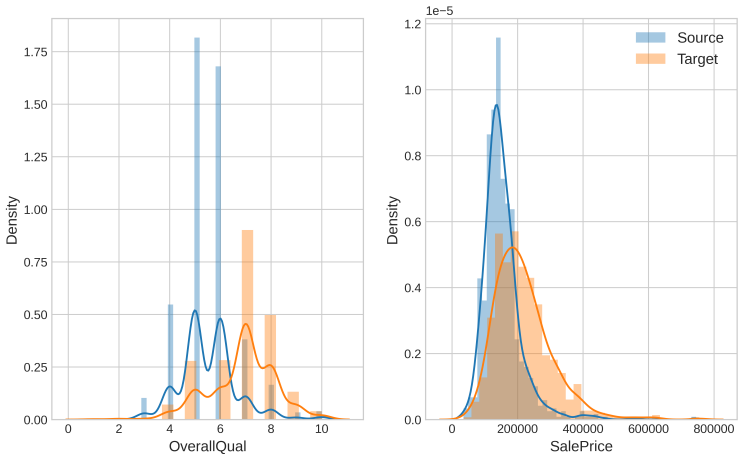


Figure 10: QveralQual distribution over source and target data

Table 3 shows the result of analysis where experimental runs were repeated 100 times. The evaluation metric are RMSE and R2, which is reported for the regression model on the target data before and after shift adaptation. Each row indicates the model used for both domain classifier and regression model. As can be seen from the results that XGboost led to better improvements considering the evaluated metrics.

Domain Classifier	**RMSE Target**	**R2 Target**	**RMSE Target w adaptation**	**R2 Target w adaptation**
Random Forest	34622.95 $\pm$ 150	0.8247 $\pm$ 0.02	34330.70 $\pm$ 50	0.8277 $\pm$ 0.03
XGBoost	35450.19 $\pm$ 120	0.8163 $\pm$ 0.02	32993.05 $\pm$ 100	0.8408 $\pm$ 0.01

Table 3: Performance of domain classifiers before and after shift adaptation

2.3.4 LIMITATIONS

The limitations that the group faced during this work include:

- Covariate shift adaptation may or may not improve the performance. For example, re-weighting factors may become incorrectly biased against certain instances and cause the classifier to overfit.
- The analysis presented in this report is specific to the dataset with the shown splits. It means that the best adaptation models or transformations presented here may not work for different use cases.
- In this study, the focus is on covariate shift while it may co-occur with label and/or concept shifts. For future work, one may also take advantage of label and concept shift detection methods.

2.3.5 BEST PRACTICES

Below are listed appropriate practices that the group discovered to be taken for addressing covariate shift:

- There must an overlap between source and target for covariate shift for using sample re-weighting.
- Consider a validation set for calculating re-weighting factors; then experiment how well the model generalizes against a test/target set.
- Verify adaptation re-weighting factor calculation carefully because a mistake in the values of the factor will result in an under fitted model that will not be able to correct shift.
- Without the distribution ratios (betas) the learners will overfit on the source data and will be unable to correct the shift. A miscalculated betas can impose the risk of underfitting the model.
- Before committing to a particular covariate shift adaptation weighting scheme, test it out on different target samples. There is evidence of there being large variance in the performance and this should be evaluated prior to model production.
- Experiment with different transformations of the weights to find one that works best. The weights may not improve the performance without some post-hoc tuning.
- Experiment with different domain classifiers. For example, the nature of a covariate shift may not necessarily be represented by a linear model (i.e., logistic regression).

2.4 CONCLUSION

Based on the three classes of experiments, the group concluded that shift detection and adaptation is advantageous in most cases of facing (covariate) shift in data. It was observed that overall loss decreased after adaptation and applying re-weighting factors in those cases. It is worth mentioning that the best domain classifier may change from one use-case to another. For example, Logistic Regression was the best choice in experiment 1, whereas XGBoost led to better results in experiment 3. Finally, the complementary experiments showed that covariate shift may not degrade the overall performance and so shift adaptation may not significantly improve the loss in such cases.

3 FOCUS AREA TWO: TIME SERIES

3.1 OBJECTIVES

In this focus area, the working group explored transfer learning and adaptive learning as means to tackle the dataset shift problem in retail industry. In retail industry, companies face a situation where they need to make decisions based on limited history. In particular, new goods are often added for sale, and the sales of these items must be estimated when only minimal historical data is available. The data distribution for the new items may be similar to that of other items (for which

adequate training data is available), but it is not identical. The group investigated the feasibility of efficiently training a ML model to deal with dataset shift in retail industry time series data. The group took two distinct approaches. In the first approach, they investigated the possibility of using transfer learning for Long Short-Term Memory (LSTM) networks [16] that leverages the similarity between the items and existing data to transfer learned knowledge from one sale item and apply it to another item with limited data without having to spend much time on retraining. In the second approach, the group explored the adaptive learning method that are robust against concept shift in dynamic environments [17, 18] to deal with possible concept shift in sales (e.g., COVID-19 pandemic would cause an unexpected shift in human behavior, resulting in predictive models becoming inaccurate due to concept shift) in the data.

3.2 METHODS

The group used Predict Future Sales dataset from Kaggle [19], which consists of historical sales data from January 2013 to October 2015. The task was to forecast the total amount of product sold. The data consists of lists of shops and product sales changes month to month. This characteristic was used to introduce a dataset shift. The group started by training forecasting models on a product that was sold in 2013 and 2014 only, and then using that trained model to predict the sales of a brand new product launched in 2015. To decide which product sale to forecast, the group iterated through the dataset looking for an item that exists in 2013 and 2014 but not in 2015 and identified a product which have the largest amount of data points (sales).

One item was selected (item 3731) from the dataset as a new item. In order to use transfer learning, the training dataset was created including only data from 2013 and 2014 without item 3731, creating a scenario where a new item is introduced for which there is no historical data for. Then lag variables of up to 7 days (1 week) were created to be fed as input to an LSTM model with with one hidden layer comprised of 29345 trainable parameters. A baseline model was built using the first subset in which include only 2013 and 2014 sales data; this was the training dataset. A rolling-forecast scenario was used where each time step of the test dataset (which is the 2015 data) was walked one at a time. The LSTM was used to make a forecast for the time step, then the actual expected value from the test set was taken and made available to the model for the forecast on the next time step. This approach was used to mimic a real-world scenario where new sales data would be available everyday and used to make the forecasting better the next day. The mean squared error (MSE) was used to compare performance of the model across the experiments.

The group explored concept shift when predicting the next-day sales of several items from the dataset. From the sales data, some features were extracted to be used for training the models: These features are days of week, lagged sales with lags of 1,2,3,4,5,6,7,14, and 28 days, and moving average of sales over a period of 7, 14, and 28 days. Initially the training data consisted of data in year 2014 and the testing data consisted of the year 2015. The group used a linear regression model as a baseline model and two adaptive models. The first adaptive model is an online version of the linear regression model which uses Stochastic Gradient Descent (SGD) updates applied to the latest points to update the coefficients of the linear model. The second model is the Hoeffding Adaptive Tree (HAT) [20]. The HAT model uses the ADWIN change detection algorithm [20] to monitor the performance of branches on the tree. If the performance of a branch (measured by MSE of its predictions) deteriorates, it is replaced with another branch which has better performance on the more recent data points. In order to have a fair comparison between adaptive and non-

adaptive methods, the group simulated retraining the non-adaptive models daily. For data with no concept shift, it was expected that non-adaptive models outperform adaptive models as the entire data is used to fit the model daily.

3.3 RESULTS AND DISCUSSION

3.3.1 PART ONE: TRANSFER LEARNING

Using the test data (year 2015), the first 19 days were used to train two identical models with one exception, one of them had transfer learning (TL) applied from our training dataset. This was done to evaluate the effect of TL by to see how the models would perform on a brand new item. Transfer learning was applied by setting the weights from the base model to the new model. The new model is the same LSTM architecture, with the exception of the weights being initialized to what was learned in the base model.

Figure 11 shows examples of the forecast without and with TL respectively (for Item 3731). The group found that on average, the model with transfer learning applied mostly performs better than the baseline model and a trained model without transfer learning applied for different items.

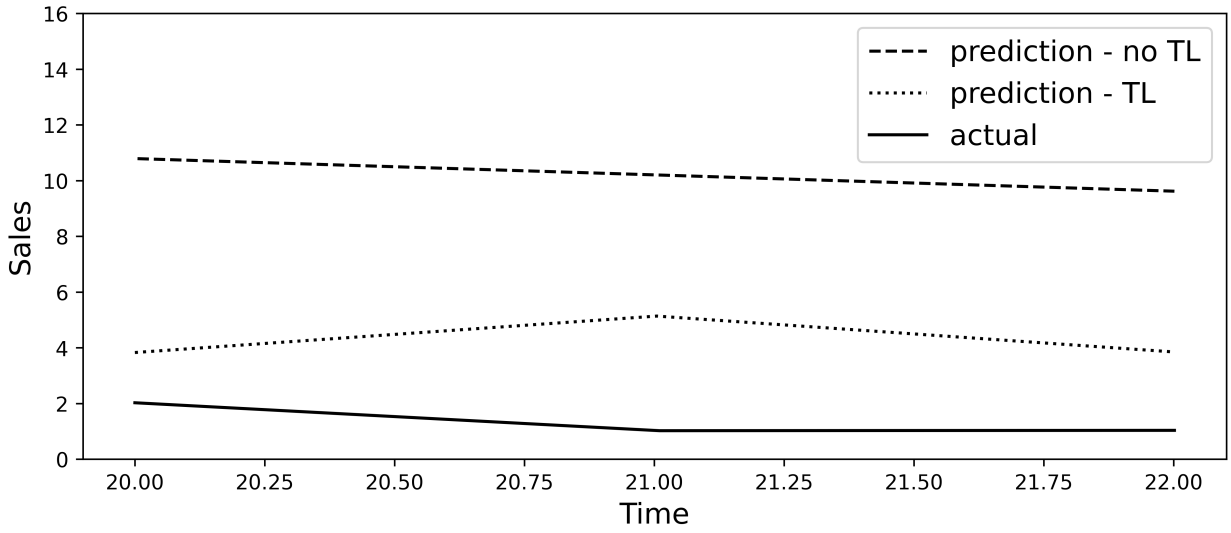


Figure 11: Prediction for Item 3731 - With TL MSE 9.3226 - Without TL MSE 77.0268

At a first glance, it looks like transfer learning helped the model make more accurate predictions even with the lack of data. However looking at the nature of the data it can be seen that the last 19 days are roughly the same value. With the limited historical data, it is also likely that the model may overfit or underfit by simply predicting the same value for the next few days.

Figure 12 shows the performance of the models with transfer learning (TL) and without transfer learning (no TL) during training. It can be seen that with transfer learning applied, the model has a lower start MSE, and converges faster, while the model with no transfer learning plateaued and could not reduce MSE after the second epoch.

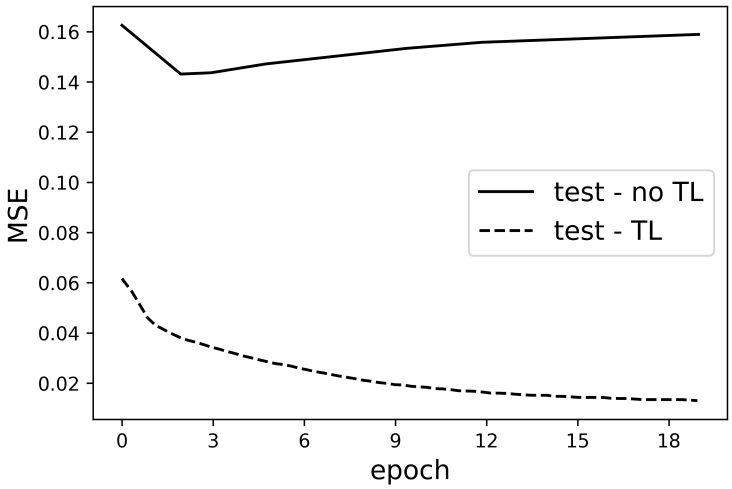


Figure 12: Mean Squared Error loss comparison

3.3.2 PART TWO: ADAPTIVE LEARNING

Figure 13 shows examples of prediction performance of different models for different items in the dataset:

Figure 13: Comparison of model performance in predicting different item sales

It can be seen from Figure 13 that there is a concept shift for items 7894, and 17117, since the performance of the linear model that was trained for year 2014 performs very poorly when used to predict year 2015. For those items it can be seen that using adaptive model results in an improvement of performance, particularly for item 17717, which exhibits a change behaviour in 2015 compared to 2014. For items that do not exhibit concept shift, the results of both the adaptive and non-adaptive items are comparable.

3.3.3 LIMITATIONS

- Adaptive models may not improve the performance if there is minimal concept shift.
- Existing Implementations of adaptive models are not as fast as standard non-adaptive models.
- There is no guarantee that transfer learning would improve the performance of the model.

3.4 CONCLUSION

According to the results, adaptive methods outperform non-adaptive methods when there is a concept shift, but results are comparable when there is no dataset shift. There is no clear winner in terms of forecasting, different models needs to be explored. Different models must be tested in order to obtain a deeper understanding of the behaviour of adaptive methods. It can be also concluded that applying transfer learning to a new model may allow it to outperform models that do not use transfer learning, as well as accelerate training. When limited data is available, transfer learning can reduce the cost of retraining a model.

4 FOCUS AREA THREE: IMAGE

4.1 OBJECTIVES

The objectives of this project are:

- Data collection, pre-processing, and visualization.
- Reproduce the results of Prototypical Networks on omniglot dataset [21]. The official code of the original paper [22] was used for this step.
- Reproduce the results of Prototypical Networks on mini-image dataset [23].
- Run prototypical Networks experiments on different combinations of omniglot dataset and mini-imagenet dataset. This step is to verify the results of Prototypical Networks when training on one dataset and testing on another.
- Reproduce the results of Model-agnostic meta-learning (MAML) algorithm [22] on Omniglot dataset. The official code of the original paper [22] was used for this step.

4.2 METHODS

In this focus area, the group focused on few-shot learning on image classification where only a small number of examples of each new class is given and train a classifier that can generalize to new classes not seen in the training set.

Prototypical networks learn a metric space in which classification can be performed by computing distances to prototype representations of each class. Compared to recent approaches for few-shot learning, they reflect a simpler inductive bias that is beneficial in this limited-data regime,

and achieve excellent results. MAML is an algorithm for meta-learning that is model-agnostic, in the sense that it is compatible with any model trained with gradient descent and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning. The goal of meta-learning is to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples. In this approach, the parameters of the model are explicitly trained such that a small number of gradient steps with a small amount of training data from a new task will produce good generalization performance on that task.

The architecture that was used in this study is the same as the embedding function used by Vinyals et al. [23], which has 4 modules with 3 × 3 convolutions and 64 filters, followed by batch normalization, a ReLU nonlinearity, and 2 × 2 max-pooling. The Omniglot images were downsampled to 28 × 28, so the dimensionality of the last hidden layer is 64. As in the baseline classifier used by Vinyals et al., the last layer is fed into a softmax. For Omniglot datest, strided convolutions were used instead of max-pooling. For MiniImagenet dataset, 32 filters per layer was used to reduce overfitting.

For mini-imagenet, the group used the splits introduced by Ravi and Larochelle [24] in order to directly compare with state-of-the-art algorithms for few-shot learning. Their splits use a different set of 100 classes, divided into 64 training, 16 validation, and 20 test classes. The group followed their procedure by training on the 64 training classes and using the 16 validation classes for monitoring generalization performance only.

For the omniglot dataset, the group followed the procedure of Vinyals et al [23] by resizing the grayscale images to 28 × 28 and augmenting the character classes with rotations in multiples of 90 degrees. They used 1200 characters plus rotations for training (4,800 classes in total) and the remaining classes, including rotations, for testing.

The default parameters of the algorithms were used to retrain the model and report the results. All of the models were trained via SGD with Adam and initial learning rate of 1×10^{-3} (cut the learning rate in half every 2000 episodes). No regularization was used other than batch normalization. For MAML, the group used one gradient update with $K = 10$ examples with a fixed step size $\alpha = 0.01$, and used Adam as the meta-optimizer. One Nvidia Titan V GPU was used for training. The training of Prototypical Networks takes 3 hours to finish, and the training of MAML takes 6 hours to finish. Due to the randomness of the model initialization, the models trained several times and reported the best results.

To train the model on one dataset and test it on another, both datasets were preprocessed such that they could be fed into the same model (i.e., to have the same dimensions). The model was then trained using one of the datasets and was tested on the other dataset (e.g., trained on Omniglot, tested on Mini-imagenet).

4.3 RESULTS AND DISCUSSION

4.3.1 PART ONE: PROTOTYPICAL NETWORKS

Performance on Omniglot dataset: The Table 4 shows comparison of the reproduced results and reported results of Omniglot dataset. It can be seen that similar results were obtained as the reported performance except for 20-way 1-shot setting, which the reproduced result is around 3.1% lower than the reported result.

Models	5-way Accuracy		20-way Accuracy	
	1-shot	5-shot	1-shot	5-shot
Reported [22]	98.8%	99.7%	96.0%	98.9%
Reproduced	98.0%	99.6%	92.9%	98.5%

Table 4: Performance of prototypical networks on Omniglot dataset

Performance on Mini-imagenet dataset: The Table 5 shows comparison of the reproduced results and reported results of omniglot dataset. It can be seen that the reproduced results are similar to the reported results.

Models	5-way Accuracy	
	1-shot	5-shot
Reported [23]	49.4%	68.2%
Reproduced	49.0%	66.9%

Table 5: Performance of prototypical networks on Mini-imagenet dataset

Performance on different combination of training and test dataset: The group ran prototypical networks on different combinations of datasets, such as training on omniglot dataset and testing on mini-imagenet dataset or vice versa. The distribution of dataset shift occurs when training a model on one dataset and testing it on another. In this experiment, the images from omniglot dataset are simple characters with clean background, while the images from mini-imagenet dataset are real objects with complicated backgrounds. The model needs to learn how to transfer the knowledge learned from one dataset to another.

Table 6 shows the results of different combinations of the datasets. Due to the dataset shift between two datasets, it can be seen that the performance drops significantly when train on one dataset and test on the other, comparing to training and testing on the same dataset (Table 4, Table 5). When the model was trained on the combined dataset (omniglot dataset + mini-imagenet dataset) and tested on a single dataset, the performance downgrade was observed due to the large dissimilarities between the two source domains.

Train on	Test on	5-way Accuracy	
		1-shot	5-shot
Omniglot	Omniglot	98.0%	99.6%
Mini-imagenet	Mini-imagenet	49.0%	66.9%
Omniglot	Mini-imagenet	27.4%	34.5%
Mini-imagenet	Omniglot	65.0%	89.3%
Omniglot + Mini-imagenet	Omniglot + Mini-imagenet	95.3%	98.9%
Omniglot + Mini-imagenet	Omniglot	95.2%	99.1%
Omniglot + Mini-imagenet	Mini-imagenet	29.8%	39.3%

Table 6: Performance on different combination of training and test dataset

4.3.2 PART TWO: MODEL-AGNOSTIC META-LEARNING

Performance on Omniglot dataset: The Table 7 shows comparison of the reproduced results and reported results of MAML networks using omniglot dataset. Similar results were obtained as the reported performance except for 20-way 1-shot setting, which the reproduced result is around 5.7% lower than the reported result.

Models	5-way Accuracy		20-way Accuracy	
	1-shot	5-shot	1-shot	5-shot
Reported [22]	98.7%	99.9%	95.8%	98.9%
Reproduced	98.6%	99.1%	90.1%	97.2%

Table 7: Performance on Omniglot dataset

4.3.3 LIMITATIONS

When replicating the reported results, the group faced a number of challenges regarding preprocessing of dataset, lack of reported experimental settings and training environments. In order to minimize the effect of these challenges, the group carefully read the official code, and preprocessed the dataset following the guideline provided by the original publications. Although both Prototypical Networks and MAML produce good results on Omniglot dataset, the performance on mini-imagenet dataset still needs to be improved.

4.3.4 BEST PRACTICES

- Prototypical networks are far simpler and more efficient than MAML.

- Performance of prototypical networks can be greatly improved by carefully considering the chosen distance metric, and by modifying the episodic learning procedure.
- MAML is also simple and does not introduce any learned parameters for meta learning.

4.4 CONCLUSION

The goal of few-shot learning is to classify new data having seen only a few training examples. Many few-shot learning approaches have been designed under the meta-learning framework, which achieves the expected performance in the scenario where all samples are drawn from the same distributions. In this project, the group experimented with prototypical networks where the few-shot learning method was experimented to classify new data having seen only a few training examples using two different image datasets. The group demonstrated that the prototypical networks can be a viable tool to tackle data distribution shifts in few-shot learning.

5 CONCLUSION AND FUTURE DIRECTION

Through the dataset shift project, launched in the Summer of 2020, the Vector Institute brought together academic and industry sponsors to explore dataset shift problems and experiment potential remedies. The primary objectives of this project were to foster and widen collaboration between academic researchers and industry applied scientists on several projects, and to build capacity for further advances in our ecosystem. The project focused on three data types: cross-sectional, time series, and image.

- **Cross-sectional:** Experiments focusing on the evaluation of covariate shift in cross-sectional data. The working group demonstrates that adaptation may not necessarily improve the performance results in all cases. That means that the best adaptation models or transformations cannot be generalized for different use cases.
- **Time series:** Experiments focusing on transfer learning and adaptive learning as a means to tackle the dataset shift problem using retail data. The working group demonstrates that adaptive methods can outperform non-adaptive methods when there is a concept shift present, and may have comparable performance when no concept shift is present.
- **Image:** Experiments focusing on few-shot learning method to classify new data having seen only a few training examples using two different image datasets. The working group demonstrated that the prototypical networks can tackle data distribution shifts in few-shot learning by computing distances to prototype representations of each class.

Overall, because of rising global threats such as global warming and climate change, as well as the recent impact of the COVID 19 pandemic on businesses and decision-making models, industries that use ML models can benefit greatly from detecting and correcting dataset shift in ML pipelines. This project demonstrates that more joint efforts between industry and academia are needed to build a deeper understanding of dataset shift in order to mitigate its effects in practical settings.

6 REFERENCES

[1] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[2] Shai Ben-David and Ruth Urner. Domain adaptation–can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence*, 70(3):185–202, 2014.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[4] Ali Pesaranghader, Herna L Viktor, and Eric Paquet. Mcdiarmid drift detection methods for evolving data streams. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2018.

[5] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

[6] Isvani Frias-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jimenez, Rafael Morales-Bueno, Agustin Ortiz-Diaz, and Yailé Caballero-Mota. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):810–823, 2014.

[7] Ali Pesaranghader and Herna L Viktor. Fast hoeffding drift detection method for evolving data streams. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 96–111. Springer, 2016.

[8] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3752–3761, 2018.

[9] Iowa house prices. https://www.kaggle.com/c/iowa-house-prices-regression-techniques/data.

[10] Kedar Potdar, Taher S Pardawala, and Chinmay D Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, 175(4):7–9, 2017.

[11] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[12] Stephan Rabanser, Stephan Günnemann, and Zachary C Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *arXiv preprint arXiv:1810.11953*, 2018.

[13] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[14] Eric W Weisstein. Bonferroni correction. *https://mathworld. wolfram. com/*, 2004.

[15] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[16] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[17] Fang Chu, Yizhou Wang, and Carlo Zaniolo. An adaptive learning approach for noisy data streams. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 351–354. IEEE, 2004.

[18] Haider Raza, Girijesh Prasad, and Yuhua Li. Adaptive learning with covariate shift-detection for non-stationary environments. In *2014 14th UK Workshop on Computational Intelligence (UKCI)*, pages 1–8. IEEE, 2014.

[19] Predict future sales. https://www.kaggle.com/c/competitive-data-science-predict-future-sales.

[20] Albert Bifet and Ricard Gavalda. Adaptive learning from evolving data streams. In *International Symposium on Intelligent Data Analysis*, pages 249–260. Springer, 2009.

[21] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

[22] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[23] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2017.

[24] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.