CE888 Assignment 2
Project 1: Learning under Covariate Shift in the Data (Supervised Learning)

NIRUPMA PANDEY
Registration number: 1906463

Abstract—Internet availability in portable devices has boomed the steaming fashion and made it easy to collect data, encouraging the need for data mining to extract information to predict consumers behavior results in developing a smarter market campaign. In this project, we are aiming to do further analysis to deal with covariate shift, which is one of the types of non-environmental data shift. We are going to implement techniques proposed in the last report to find drift in two different datasets and work on a proposed methodology for treatment and enhancement of output, followed by a comparative study showing a performance difference with and without covariate drift.

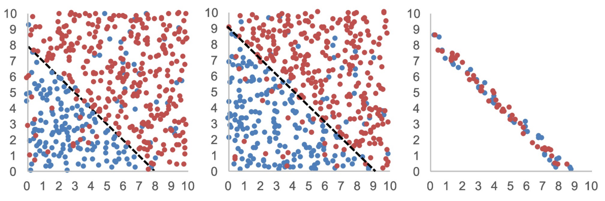
Index Terms—Data Shift, Covariate Shift, ROC curve, AUC score, Phi coefficient.

1 Introduction

With the rapid advancement in technology such as the internet of things, faster and cheaper internet access, social networking platforms, online recommendation system, and digitization of services have encouraged net surfing. Willingly or unwillingly every day we are sharing our personal information and preferences, resulting in pulling off a massive amount of data. Easily accessibility of personal data and preferences has promoted data analysis using machine learning techniques and mathematical modeling for behavioral analysis of customers, resulting in a promotion and recommendation system.

Sometimes during data collection due to the presence of some non-stationary environment factors or unavailability of complete data (user willing not hide some of the information), causes date drift or data shift. Data drift is a situation when the joint distribution of input and output in the test and train dataset is uneven. Mathematically, for a predicted variable y and response variable x:

$P_{train}(y, X) \neq P_{test}(y, X)$



(a) Data at time t (b) Data at time $t+1$ (c) Data in drift regions

Fig. 1. Figure 1 showing drift in data set with respect to time, (a) shows the data distribution at time t, (b) shows distribution at t+1, (c) data distribution in drift region

Based on distribution, types of data shifts are:

(a) Covariate Shift - shift in independent variables in train and test datasets, also known as covariates.

(b) Prior Probability Shift - shift in target variable in train and test datasets.

(c) Concept Shift - shift in relationship between independent variables and target variables.

Analysing a dataset is procedure includes preprocessing, feature engineering, model selection, parameter tuning, validation score, and test prediction. During manipulation, if we build a model only considering the features present is the training file, and ignoring the different information/properties of features present in the test file, could result in predicting the biased result. Hence before fitting a model a prior investigation of finding data shift, if any. In the project, we are going to look at the methods which can be used to detect data drift in a given dataset such as plotting a histogram with features having drift, applying cross-validation, plotting Receiver Operating Characteristic Curves (ROC) and Area Under Curve (AUC). Later in this project, we are going to look at the methods to deal with the same problem and improve performance score using appropriate methods, discussed under the methodology section in detail.

2 Background

In supervised learning, the learning aim is to infer an input-out dependency from training data, using which output of an unknown test data can be predicted. We generally assume that input points of both train and test are having the same probability distribution[3]. Covariate shift is a specific class of selection bias that arises when the marginal distributions of the input features X are different in the source and the target domains while the conditional distributions of the target Y given X are the same, mathematically

$P_{train}(X) \neq P_{test}(X) and P_{train}(Y|X) = P_{test}(Y|X)$,
where X is a feature [1]

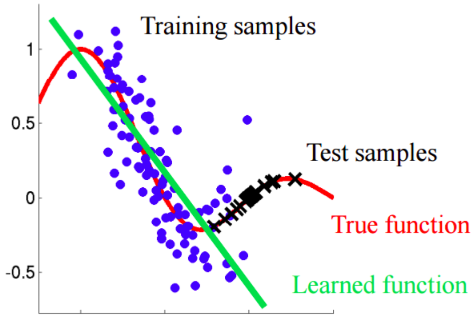


Fig. 2. Covariate shift in data

The problem of covariate shift can be found in real-world instants very easily, for example, data of working-age group for an insurance policy, data collected over a long period on working efficiency of machinery, filtering spam emails, brain-computer interfacing[3]. In the first example, if data of people aged between 25-60 has been collected and distributed such that training contains fairly young candidates, whereas elderly people are classified as a test, it results in a significant difference in mean distribution.

Similarly, if records collected on the working efficiency of machines in a production plan over a time period, may result in a significant difference in distribution, due to wear and tear of some/most of the parts.

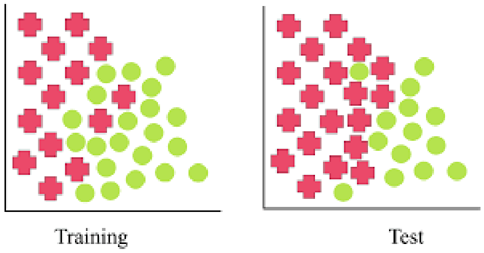


Fig. 3. Example showing distribution difference in test and training data

3 Methodology

In this section, we are going to discuss some of the proposed and well-known methods used for identifying covariate shift, techniques to treat the problem, and improve performance with fair prediction.

3.1 Identification of covariate shift

As mentioned earlier in the above sections, in covariate shift the joint distribution of train and test are not the same. The easiest way to find a covariate shift is by combining both train and test files as a single dataset and still being able to classify each instant accurately to the class it belongs originally. The procedure is divided into below-mentioned steps as:

- Preprocessing: The first step is missing values imputation in both train test and labeling converting categorical variable.
- Drop target variable from train to remain with the same number of features with test.
- Create a new feature Origin, add Origin = 0 in train and Origin = 1 in test, which will help us to identify the actual class of a random instant picked from merging both dataset.
- Take a random sample from train and test and merge them into a single dataset,then drop Origin column
- Now fit a model taking one feature at a time with considering Origin as target variable on more than 70% of the data
- Predict on rest 30% of the data and calculate AUC-ROC score.
- Lets fix a threshold for AUC-ROC score, feature with a score greater than threshold value is having drift.

For this project, we are working on two different datasets. The first dataset is 'Bank Load Sanction' and the second 'Sberbank Russian Housing Market' dataset from kaggle. After finding the variables with drift, we have plot histogram for visualization:

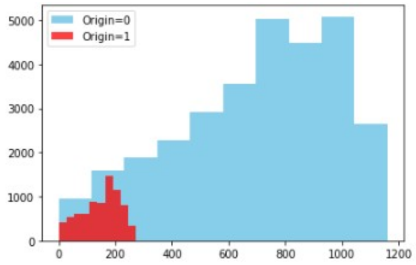
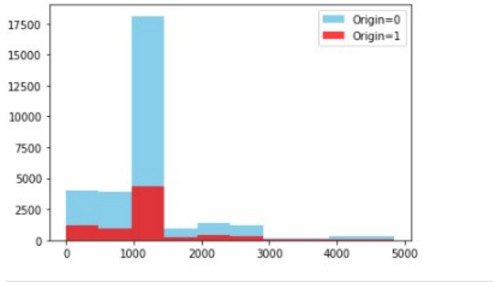


Fig. 4. Histogram showing drift feature 'timestamp', where Origin =0 for train and Origin =1 for test



3.2 Treatment of drift in data

Once we found the features with drift, we will check for possible methods to treat the problem to improve performance. Broadly mentioned two methods are suggestive:

- Dropping of variable with drift, which are less importance
- Density Ratio Estimation

3.2.1 Dropping features with drift

The first method is checking the importance of each feature in prediction, using Random Forest feature importance ('rf.feature_importances_' code). For our datasets, we have generated an array of features with drift and using the 'rf.feature_importances_' code to generate a plot showing features playing an important role in predicting outcome. Then we have manually check for less important features and drop them to improve the score.

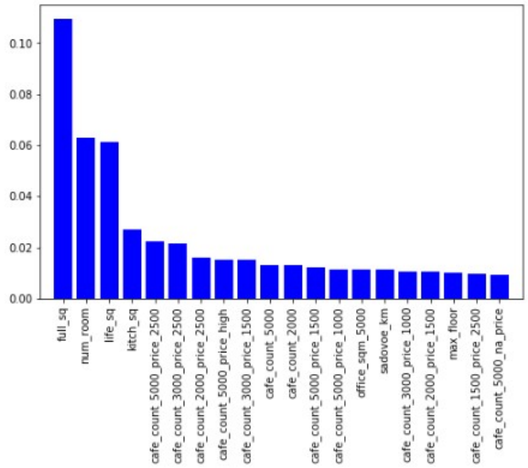


Fig. 6. Plot showing the features with their importance in prediction

3.2.2 *Density Ratio Estimation*

In the technique, we will first calculate the densities of both train and test datasets separately and the second part is to estimate importance by calculating the ratio of estimated densities of train and test. These ratios are considered as the weighing of each row in the training dataset. Due to the long processing time for calculating the density ratio of each feature, we are avoiding this method for this project.

3.3 Comparative study of datasets before and after treatment

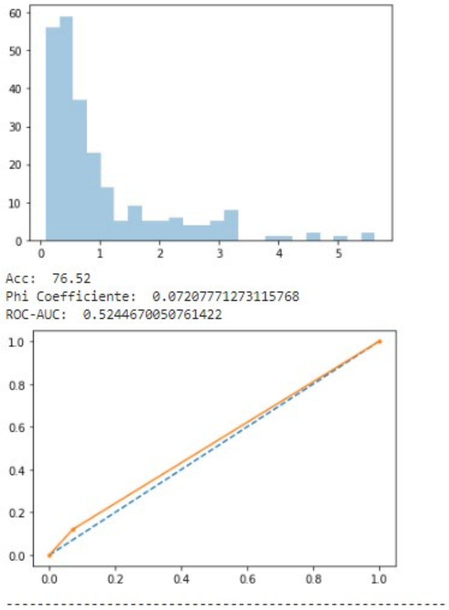
After applying the treatment to minimise the effect of shift, we will perform some comparative study on the data set to measure the accuracy or correctness of the model. One of the commonly used methods is calculating the AUC score for ROC. The receiver operating characteristic curve (ROC) is a graphical representation of the performance of the classification model at all classification threshold, by plotting two parameters- True positive rate and False positive rate.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

Whereas Area Under Curve (AUC) is calculated from ROC and it measures two-dimensional area for ROC, between (0,0) to (1,1). In this assignment to determine variables with covariate shift, we have fixed a threshold of 0.3 and 0.8 for the Phi coefficient and AUC-ROC score respectively.

The last step is to calculate the accuracy classification score of the prediction of the model before and after the removal of features with drift. Accuracy classification score is the percentage of test data instants being correctly classified. Accuracy score is also helpful in guessing if there is an under-fitting or over-fitting or unbiased prediction.



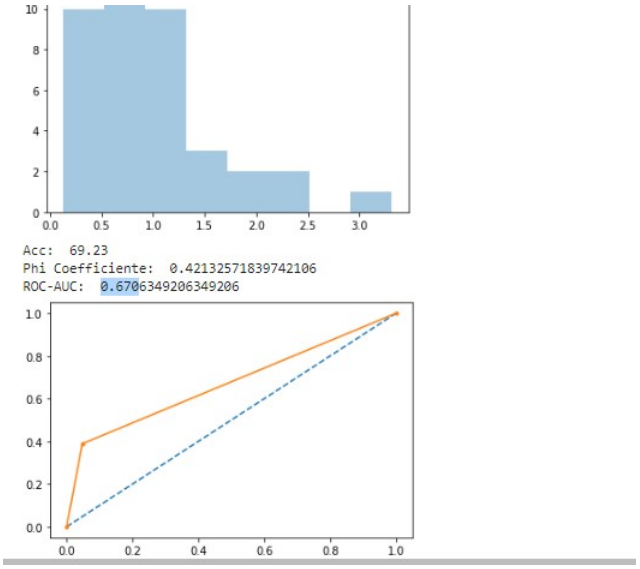
```
Acc:  76.52
Phi Coefficiente:  0.07207771273115768
ROC-AUC:  0.5244670050761422
```

Fig. 7. Showing the ROC curve and mathematical value of accuracy, Phi coefficient and AUC-ROC score

4 Result

In this assignment, we have worked on two different datasets from Kaggle to investigate the covariate shift and to treat them. The first dataset we have picked from Kaggle is ’Bank Load Sanction’ with a target variable Loan_Status. Test and train files are loaded on Faser named as test_3.csv and train_3.csv respectively and coded under jupyter notebook CE888_2.2.ipynb. For the analysis of drifted features, we have set a threshold of 0.5 for roc_auc. Using the Random Forest classifier to calculate roc_auc score of each feature, a total of six features were detected with a score higher than a threshold of 0.5. Before starting with the treatment part, our model was predicting with an accuracy of 58.97%, rocauc score of 0.579. Then we have checked for 8 variables covering maximum information of the dataset and then compare the list with the array containing features with drift. Then we have removed less important features, which are shown to have a covariate shift. After removing such two features, the accuracy of our model has a significant improvement with 69.23% and rocauc score of 0.671.

The second dataset we have picked from Kaggle is ’Sberbank Russian Housing Market’ with a target variable price_doc. Test and train files are loaded on Faser named as test.csv and train.csv respectively and coded under jupyter notebook CE888_2.1.ipynb. For the analysis of drifted features, we have set a threshold of 0.8 for roc_auc. We have applied the Random forest classifier to calculate roc_auc score of each feature, a total of eight features were detected with a score higher than a threshold of 0.8. Before starting with the treatment part, our model was predicting with an accuracy of 61.16%. Then we have checked for 20 variables covering maximum information of the dataset and then compare the list with the array containing features with drift. Then we have to remove less important features, which are shown to have a covariate shift. After removing such five features, the accuracy of our model has slightly decreased to 61.13%,



```
Acc:  69.23
Phi Coefficiente:  0.42132571839742106
ROC-AUC:  0.6706349206349206
```

Fig. 8. Showing the ROC curve and mathematical value of accuracy, Phi coefficient and AUC-ROC score for first dataset after removing less important features with drift

instead of expected improvement.

5 Conclusion

During this project, we have worked on two different datasets, with an aim of understanding under covariate shifts, in the later steps we have done a comparative study to measure significant improvement in prediction after removing features which are causing drift. The main finding of our work can be summaries as:

- working a dataset with covariate shift could result misleading outcomes or machine learning model failure.
- plotting histogram is an easy way to find and visualise feature wise distribution difference in train and test.
- a good analysis of dataset is required before setting threshold values.
- for the first dataset the method of removing less important features with drift has shown a significant improvement(by 10.26%).
- however, for second dataset, drifted feature dropping has slightly decrease the performance (by 0.03%).

6 Discussion

Our study has suggested that further research is required in this area. Available methods are not always helpful in improving the prediction, as seen in our findings with the second dataset. It indicates that we need to look for other techniques for treatment, which could reduce the processing time and generate measurable improvement in most of the cases. In the first dataset, we have seen that features with drift are containing maximum information, hence we have removed only two features which is showing the least importance. A further investigation is required in the area to check if the improvement in the first dataset accuracy is a result of overfitting.

References

G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, “Learning in nonstationary environments: A survey,” IEEE Computational Intelligence Magazine, vol. 10, no. 4, pp. 12–25, 2015.

H. Raza, G. Prasad, and Y. Li, “Adaptive learning with covariate shift-detection for non-stationary environments,” in 2014 14th UK Workshop on Computational Intelligence (UKCI). IEEE, 2014, pp. 1–8.

M Sugiyama, M Krauledat, KR MÃžller, ”Covariate shift adaptation by importance weighted cross validation” - Journal of Machine Learning . . . , 2007