Learning under Covariate Shift in the Data CE888 Assignment 2

Belia Maria Baez Molina, *1900952*

***Abstract*—The constant change in the information on real-life provokes a change in studying the data. A modification on this data consequently will generate problems while learning from it. This project suggests a method to detect those differences on the data and an approach to solve it using a discriminator to correct the change. It compares a shift data score on a Kaggle competition with another attempt without the shit. Key results on the project are the significant improvement on one dataset and a slight decrease in another one.**

***Index Terms*—Covariate shift, Classifiers, Adaptation Methods, Kaggle Competition**

I. Introduction

IN supervised learning, is normally used the same distribution of data in the training and test data sets, but there is a case where the input distribution P(x) is distinct in the training and test as commonly expected. However, in spite of that singularity, the conditional distribution of output $p(y|x)$ does not change, and that is called covariate shift [1]. Having a shift in the data does not means that it can be a problem on the accuracy of the results, it all depends on the amount of information and the abilities of the programmer [2].

This topic has gain popularity recently [3], especially around data analysis due to the suspected influence that this has over deep neural networks.

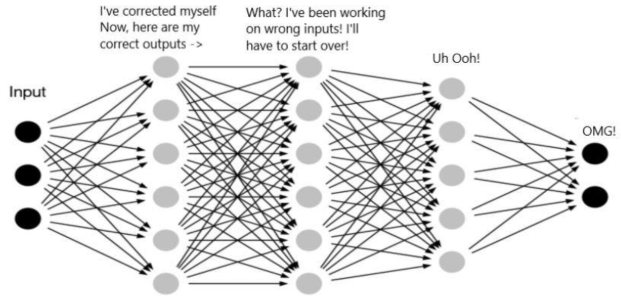

I've corrected myself Now, here are my correct outputs ->
What? I've been working on wrong inputs! I'll have to start over!

Fig. 1. Reaction of covariate shift on hidden layers of deep neural networks [2]

Researchers [2], found that there is a delay in the training due the alteration of the data. This is because the training data output is the input of a consecutive layer and this evoke a slower learning due the bewilderment. Demonstration in Fig 1. Also, covariate shift is commonly seen in credit card fraud [3], the shift happens in a different use of the card. With that, it can be identify if there is a fraud on the card or not.

In order to create a solution to the misestimation fomented due covariate shift, there hav been investigations on how to solve it. One of the methods is the ”Kullback-Leibler Importance Estimation Procedure (KLIEP)”[1]. This method focuses on returning a unique global solution. A key contribution of that research is the implementation of cross validation on the test input samples [1]. Another method is the ”Batch Normalization”, this method is know because of the accuracy it provides because it needs less steps [4]. The one used in this project is a simpler one, is by training discriminator model, this one consist on using a discriminator to correct for a difference in the distributions of datasets [5].

With a dataset with few values or if the information is not crucial, is acceptable to use a simpler adaptation method because the processing time does not make a big impact and having a sharp accuracy is not critical. As mention before, for those type of cases, is better to use a more complete method as KLIEP and Barch Normalization.

II. Background

A dataset shift is associated to transfer learning and inductive transfer [6], the difference is that transfer learning works with general problems and dataset shift is more specific.

The shift in the data can be presented in distinct forms those are Simple Covariate Shift, Prior Probability Shift, Sample Selection Bias, Imbalanced Data, Domain Shift and Source Component Shift [6]. Learning under covariate shift is a supervised learning. This type of learning does not know the input-output dependency of the training data, these is use to calculate unseen test inputs [1]. The shift happens only in the testing data, information that the user can not see, and it can be by many reasons and it can affect to one or more features and even the whole data [3].

In [1], they use a method to demonstrate the adaptive learning with covariate shift-detection for motor imagery-based brain- computer where interfaces are used with previously developed covariate shift detection to classify between mental tasks and generate neurofeedback. This feedback is planned to be used in the form of visual and exoskeleton motion.

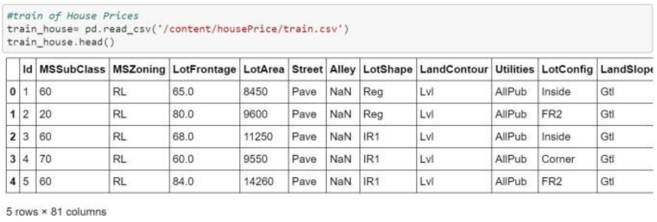
III. Methodology

In order to start working with a dataset, is important to inspect them mainly to see if there is a covariate shift in there. To identify that shift, the data was shown in a histogram to identify if there are changed between the train and the test data. Before plotting it, is necessary to filter and clean the columns. Sometimes, information can be missing and it appears as NaN, another thing that can happen is that the labels are strings and it not possible to plot them, that is why is important to first convert those pieces of information into digits.

In order to achieve the purpose of the project, there were selected two datasets from the community Kaggle.com; those were House Prices: Advanced Regression Techniques and Titanic: Machine Learning from Disaster. Both datasets were previously used in different challenges.

Continuously, to obtain the data from Kaggle, it was downloaded to the virtual machine of google colab via the API of the dataset. kaggle competitions download kaggle competitions download -c house-prices-advanced-regression-techniques and Kaggle competitions download -c titanic.

Once it is already downloaded, it is necessary to unzip them to start working. The process consists of after unzipping them, save the train.csv and test.csv file into variables for the easy manipulation. Is important to confirm that the data is correctly saved in the variables before continuing with the rest of the process. The example is shown in Fig2.

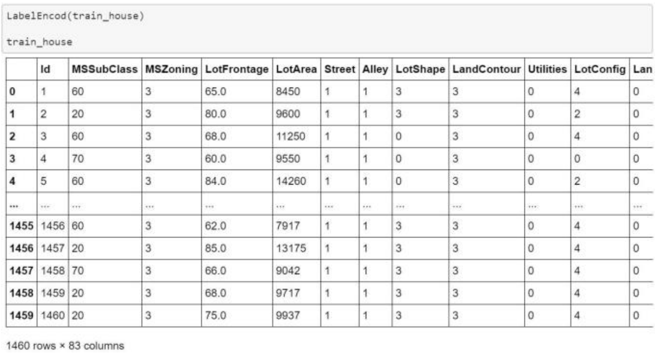


```
#train of House Prices
train_house= pd.read_csv('/content/housePrice/train.csv')
train_house.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl

5 rows × 81 columns

As mention before, the data can contain NaN and strings as values. In the case of The House Prices data, there was specific information to fill the Nan values. After that, there are only strings and integers on the data, so a label encoder function was used to convert everything into numbers in order to plot the information as shown in Fig 3. Is crucial to adapt the information instead of erasing it because, with missing values, the results can be wrong.



```
LabelEncod(train_house)

train_house
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	Lan
0	1	60	3	65.0	8450	1	1	3	3	0	4	0
1	2	20	3	80.0	9600	1	1	3	3	0	2	0
2	3	60	3	68.0	11250	1	1	0	3	0	4	0
3	4	70	3	60.0	9550	1	1	0	3	0	0	0
4	5	60	3	84.0	14260	1	1	0	3	0	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	3	62.0	7917	1	1	3	3	0	4	0
1456	1457	20	3	85.0	13175	1	1	3	3	0	4	0
1457	1458	70	3	66.0	9042	1	1	3	3	0	4	0
1458	1459	20	3	68.0	9717	1	1	3	3	0	4	0
1459	1460	20	3	75.0	9937	1	1	3	3	0	4	0

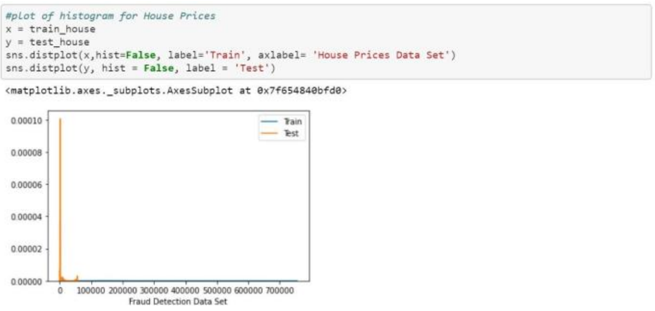
1460 rows × 83 columns

Fig. 3. Visualization of the data ready to be plotted.

These same steps were applied to the train data, for simplicity only the train data is going to be shown.

Afterwards, the plot the histogram is used to verify if there is any shift in the data. Those are made by the method sns.distplot(). This type of graphic helpful to observe the change easily in data because it is a continuous line. In Fig 4 is presented the shift between the train and test data.

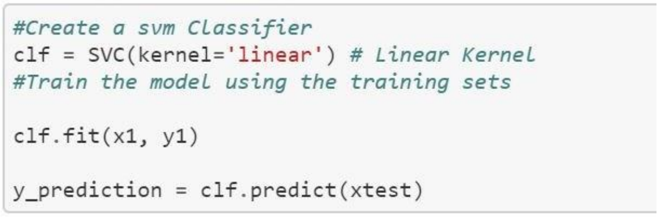
Now, the next step is to implement a basic classifier. It is used the Support Vector Machine, Fig 5, for both data sets due to the transformation of data to find the optimal boundary between the outputs. The first rmse score of submission on Kaggle for House sale: 0.3414 and the prediction of survival for Titanic was of 70% (score: 0.77511). More details about the scores will be on the Results Section.



```
#plot of histogram for House Prices
x = train_house
y = test_house
sns.distplot(x,hist=False, label='Train', axlabel= 'House Prices Data Set')
sns.distplot(y, hist = False, label = 'Test')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f654840bfd0>
```

Fig. 4. Visualization of the information plotted.

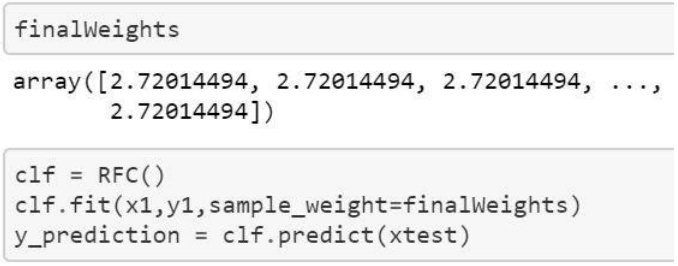


```
#Create a svm Classifier
clf = SVC(kernel='linear') # Linear Kernel
#Train the model using the training sets

clf.fit(x1, y1)

y_prediction = clf.predict(xtest)
```

Fig. 5. Support Vector Machine classifier.

The plots showed a shift in the data; that is why the scores were not optimal. Subsequently, there was the implementation of a covariate shift adaptation method on the data.

For this project, it was applied a discriminator to correct the shift between datasets. It was selected this method due to the simplicity of it and because there was not a lot of data, so the processing time was not relevant to select the method. Some adjustments were needed to be applied because of the values of the data. This method provides, as a result, a simple weight that is added to the fit() method of the classifier to have a new prediction, Fig6. An important aspect to notice is that for the final prediction, a different classifier was used. It was changed to Random Forest Classification due to the result of a better prediction.



```
finalWeights
```

```
array([2.72014494, 2.72014494, 2.72014494, ...,
       2.72014494])
```

```
clf = RFC()
clf.fit(x1,y1,sample_weight=finalWeights)
y_prediction = clf.predict(xtest)
```

Fig. 6. Final prediction without the shift.

IV. Results

The results of prediction where to obtain from Kaggle, it was manually uploaded it to the competition.

A. Predictions with shift on the data

For the Titanic Dataset: score of 0.77511, 70% of accuracy For the House Prices: rmse score of 0.34142

B. Prediction without a shift on the data

For the Titanic Dataset: score of 0.77033, 70% of accuracy For the House Prices: rmse score of 0.0.2084

V. Discussion

The score that the Kaggle competition provides is the most accurate results the project can have. For the House Price competition, the more the score is closer to cero, the better. For this one, there was a significant improvement on the score after applying the discriminative classifier; it decreases more than .1000. On the other hand, for the Titanic competition, the adaptation hampered the performance giving as result a minimum decrease on the score. A reason for this can be the difference between the amount of information on each dataset. For future work, a different method can be implemented on the Titanic data, make different attempts of each method to see which one gives a better prediction where there is less information that other data.

VI. Conclusion

This project has the objective to identify the performance under covariate shift using different classifiers. For the classifiers, there were no problems on implementations but as a limitation of the project was little knowledge on adaptation methods for covariate shift. There were a few tutorials for it and not so understandable into personal perspective, that is why it the method was selected, due to the simplicity of coding and understanding. A shortcoming presented on the project was the decrease in the score in the Titanic competition. It was expected to improve as the other case. Future work for the field can be the relation between classifiers and adaptation methods. There is information about the methods and tutorials for the classifiers, but there are few articles that present which classifier work better with which method.

References

[1] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert MÃžller. “Covariate shift adaptation by importance weighted cross validation”. In: *Journal of Machine Learning Research* 8.May (2007), pp. 985–1005.

[2] Matthew Stewart. *Understanding Dataset Shift*. Dec. 2019. URL: https://towardsdatascience.com/understanding-dataset-shift-f2a5a262a766.

[3] Jose G Moreno-Torres, Troy Raeder, RocıO Alaiz-RodrıGuez, et al. “A unifying view on dataset shift in classification”. In: *Pattern recognition* 45.1 (2012), pp. 521–530.

[4] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[5] Davidson Erlend. *covariate-shift-adaption*. https://github.com/erlendd/covariate-shift-adaption. 2016.

[6] Amos Storkey. “When training and test sets are different: characterizing learning transfer”. In: *Dataset shift in machine learning* (2009), pp. 3–28.