Project 1: Machine Learning Pipeline

Sarah Wilson SAWI117@JHU.EDU
303-921-7225
Engineering Professionals Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA

1. Introduction

Regression and classification are both common tasks in the realm of Machine Learning. Regression and classification are both supervised learning problems. Supervised learning is where the system is given an input and output and then asked to learn or predict the mapping of input to output.

Regression is used to solve problems where the outcome is a number. An example of a Regression problem would be if a system needed to be built to predict the price of a car based off of certain attributes, such as, mileage, accident history and age.

Classification is used to solve problems where the outcome is a classifier or string. An example of a Classification problem would be if a system needed to built to predict if a loan was 'high' or 'low' risk, based off of certain attributes of the person applying for that loan, such as, credit score, previous loan history and income.

In order to implement Regression and Classification algorithms it first must be noted that there needs to be data for these algorithms to run on. A crucial component in Machine Learning is the pre-processing of the data sets that the algorithms are intended to run on. The primary motivation behind this project was to develop a Machine Learning Pipeline that could be used to pre- process multiple unique data sets in order to pass the data to the algorithms. Due to the fact the primary objective was proper data handling the only algorithms that will be discussed in this report are: for Classification problems a Naive Majority predictor and for Regression problems a Naive Average predictor. These algorithms will be evaluated by using the k-fold cross validation method.

The algorithms implemented for both Classification and Regression are very simple. This leads to the hypothesis that the results from these simplistic algorithms will be highly inaccurate and produce large errors.

Section 2 will discuss more examples of ways that data needs be pre-proccesed before entering the algorithms, the data-sets that were leveraged, the algorithms themselves and the k-fold cross validation method. Section 3 will present the results obtained by the Classification task using a Naive Majority predictor and for Regression task using a Naive Average predictor. Section 4 will discuss the results that were obtain and compare that to the hypothesis that was outlined in the introduction. This report will conclude in Section 5 with a discussion of lessons learned and areas of possible future work.

2. Algorithms and Experimental Methods

The primary objective of this project was to create a Machine Learning Pipeline that was capable of prepping data sets for insertion into algorithms. The primary methods that were developed to be used in the future were: The handling of missing data, handling categorical data, discretization and standardization.

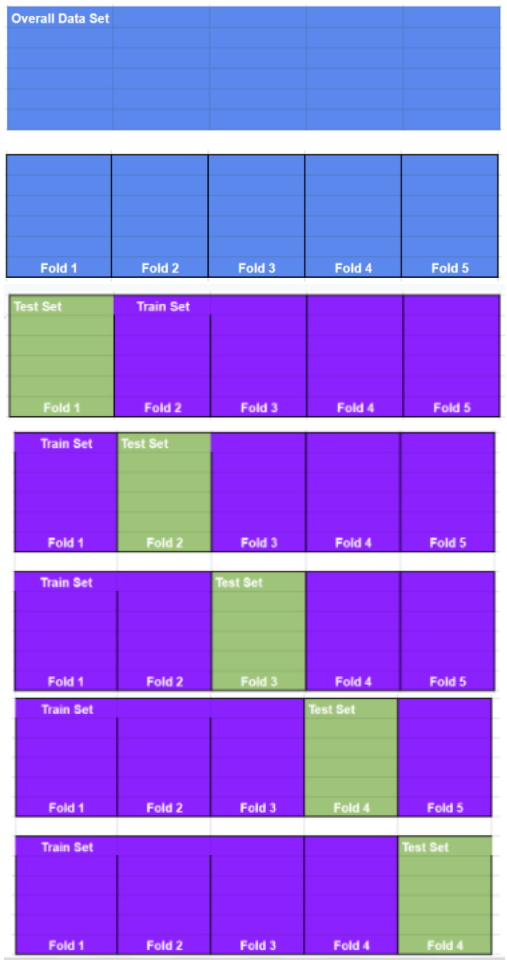
Handling of missing data is needed as real life data will in many cases be incomplete for certain attributes. For the purposes of this project if missing data was found in any particular feature column, the average of that feature column was taken and then the missing element was filled with that average.

Handling of categorical data was addressed in two ways. For ordinal data where the order of the categories was important to preserve, those categorical values were converted to integer values that keep the ordering of the original categories. For nominal data, which is data that has no order, all that was applied was a binary one-hot encoding, where the value of the category is encoded in set of 1s or 0s. Handling this data is important as algorithms that are explored that need to run on categorical data but can not handle strings as the input type.

Dicretization is the process of taking an attribute and binning that attribute. The binning process can be described into two ways, either with equal width binning on with equal frequency. In the equal width case, the bins are of equal width and may have different numbers of data points that fall into each bin. In the equal frequency case, bins may have different widths but typically have the same amount of data points in each bin. Both these methods can cause a reduction in the amount of data available but is useful for some algorithms.

Standardization is the process of getting all the data points in the data set on the same scale. One method of Standardization is z-Standardization, this can be expressed as the following equation: $\frac{(x-\mu)}{\sigma}$. Where μ is the mean and σ is the standard deviation. This z-Standardization is derived from the the training set and then applied to the test set. Note for results presented in Section 3, there was no z-Standardization applied on the data prior to being fed into the algorithms.

The secondary objective of this project was to implement a Naive Majority and Average predictor for Classification and Regression tasks respectively. The data sets that were provided were small and it is important to optimize the amount of data available to train the predictors on. The method used in this project is called k-fold cross validation. At a high level the k-fold validation process can be summarized as taking the overall data set, breaking in up into a defined number of unique and separate folds. A partition of those folds will be called the test set and another larger partition will be called the training set. The algorithm is then run on the train set and test set, results are returned and compared to the test set to determine accuracy or error parameters. The test set is then moved to a different fold of the n number of folds available and the train set is compromised of what is left. The algorithms are then run again. This cycle repeats until the test set has been in all available spots in the overall fold structure. This process is summarized pictorially below. There can also be a validation partion that is defined from the original data set. Note for results presented in Section 3, there was no validation partition in place, there were only test and train sets used.



Data Sets

The following data sets were used during the classification and regression tasks for this project.

Breast Cancer

Description:
Task: Classification
Predictor: Diagnosis (Malingant or Benign)
Link:
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

Car Evaluation

Description:
Task: Classification
Predictor: Car Evaluation (Unacctable, Acctable, Good, Very Good)
Link:
https://archive.ics.uci.edu/ml/datasets/Car+Evaluation

Congressional Vote

Description: 1984 United Stated Congressional Voting Records
Task: Classification
Predictor: Party (Reblican / Democrat)
Link:
https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

Albalone

Description: Physical merasurements of Albalone
Task: Regression
Predictor: Rings (int)
Link:
https://archive.ics.uci.edu/ml/datasets/Abalone

Computer Hardware

Description: Realative CPU performance data.
Task: Regression
Predictor: PRP
Link:
https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

Forest Fires

Description: Forest Fire burn area data
Task: Regression
Predictor: Area (float)
Link:
https://archive.ics.uci.edu/ml/datasets/Forest+Fires

3. Results

The following results were obtained from the Classification task data sets. Tables 1-3 display the results from the Breast Cancer, Car Evaluation and Congressional Vote data sets. These tables show the result from the train set and the test set during each fold of the k-fold validation process. The tables also display the accuracy and error obtained during each fold, and the total accuracy and error as averaged across each of the 5 k-folds. An accuracy(scale of 0-1) of 1 indicates that the train set had the most frequently occurring classifier equal to the most frequently occurring classifier in the test set. An error(scale of 0-1) of 0 indicates that the test set and train set had the same most frequently occurring classifier. Note in the Breast Cancer data set a value of 2 was mapped to the classifier of benign.

Table 1: Breast Cancer: Naive Majority Predictor Results

Breast Cancer	**Test Set**	**Train Set**	**k-fold Accuracy**	**k-fold Error**
k-fold[1]	2	2	1	0
k-fold[2]	2	2	1	0
k-fold[3]	2	2	1	0
k-fold[4]	2	2	1	0
k-fold[5]	2	2	1	0
Total Average Accuracy	**1**			
Total Average Error	**0**			

Table 2: Car Evaluation: Naive Majority Predictor Results

Car Evaluation	**Test Set**	**Train Set**	**k-fold Accuracy**	**k-fold Error**
k-fold[1]	Unacceptable	Unacceptable	1	0
k-fold[2]	Unacceptable	Unacceptable	1	0
k-fold[3]	Unacceptable	Unacceptable	1	0
k-fold[4]	Unacceptable	Unacceptable	1	0
k-fold[5]	Unacceptable	Unacceptable	1	0
Total Average Accuracy	**1**			
Total Average Error	**0**			

Table 3: Congressional Vote: Naive Majority Predictor Results

Congressional Vote	**Test Set**	**Train Set**	**k-fold Accuracy**	**k-fold Error**
k-fold[1]	Democrat	Democrat	1	0
k-fold[2]	Democrat	Democrat	1	0
k-fold[3]	Democrat	Democrat	1	0
k-fold[4]	Democrat	Democrat	1	0
k-fold[5]	Democrat	Democrat	1	0
Total Average Accuracy	**1**			
Total Average Error	**0**			

The following results were obtained from the Regression task data sets. Tables 4-6 display the results from the Albalone, Computer Hardware and Forest Fire data sets. These tables show the result from the train set and the test set during each fold of the k-fold validation process. The tables also display the error obtained during each fold, and the total error as averaged across each of the 5 k-folds The error is the absolute error and was calculated as: $|TestSetAverage - TrainSetAverage|$. This is meant as a measure of how far off the train set average was from the test set average.

Table 4: Albalone: Naive Mean Predictor Results

Albalone	**Test Set**	**Train Set**	**k-fold Error**	
k-fold[1]	10.00	9.56	0.43	
k-fold[2]	9.00	10.26	1.26	
k-fold[3]	9.00	9.79	0.79	
k-fold[4]	9.00	9.99	0.99	
k-fold[5]	9.00	10.04	1.04	
Total Average Error	**0.90**			

Table 5: Computer Hardware: Naive Mean Predictor Results

Computer Hardware	**Test Set**	**Train Set**	**k-fold Error**	
k-fold[1]	33.00	94.91	61.91	
k-fold[2]	36.00	119.29	83.29	
k-fold[3]	66.00	111.65	45.65	
k-fold[4]	45.00	105.31	60.31	
k-fold[5]	30.00	96.98	66.98	
Total Average Error	**63.63**			

Table 6: Forest Fire: Naive Mean Predictor Results

Forest Fires	**Test Set**	**Train Set**	**k-fold Error**	
k-fold[1]	0.00	16.08	16.08	
k-fold[2]	0.00	15.20	15.20	
k-fold[3]	0.00	7.85	7.85	
k-fold[4]	0.00	13.81	13.81	
k-fold[5]	0.00	11.28	11.28	
Total Average Error	**12.84**			

4. Discussion

The hypothesis was that since the algorithms implemented for Classification and Regression were very simplistic the prediction results from these algorithms will be highly inaccurate and produce large errors.
The Classification results seemed to indicate otherwise, for all Classification data-sets run through the Naive Majority predictor, for each fold the training set matched the test set. Due to this result occurring across three different data sets, the Naive Majority predictor should under-go more review. It was validated that the test set was unique and contained no duplicate data from the train set, thus ruling out an issue in the k-fold process it self. However, each of these data set was not shuffled prior to entering into the k-fold generation process. It is possible that the storage structure of the data as it's pulled down raw from the source, had some intrinsic ordering that impacted the results.

The Regression results also indicated that there was is most cases lower error than originally hypothesized. The error was relatively low k-fold to k-fold and on average across folds for the Albalone and Forest Fire data set. The computer hardware data-set seemed to have a larger magnitude of error. It is possible that the computer hardware had a larger spread of values in the Attribute PRP that was used to perform the Naive Average predictor on. This large spread would then transfer to the test and train sets more often. In other words, as the data get cut sliced for the test set there is a higher chance that more of the higher PRP values fall into the test set, which can attribute to the error, as the average of higher values will be higher than the average of lower values, and vice versa. The Forest Fire data set did not have a logarithmic transform applied, leading to the error being directly equal to the average that was computed by the train set.

5. Conclusion

From this project it can be concluded that pre-processing the data in order to feed algorithms is an important step in Machine Learning. This project also implemented a Naive Majority and Average predictor on data sets using k-fold cross validation. Reviewing the results, it can be seen that trends that are apparent in the raw data can have an impact on the result of our predictors. This was most noticeable in the Forest Fire data set, where many of the results were skewed towards zero. Without a logarithmic transform in place, to represent all data points from this data set on a representative scale, the overall error in our mean predictor was directly equal to the result average of our training sets. For future would it would be useful to implement a logarithmic transform of the data and in addition it would be helpful if the Naive Majority and Average predictor algorithms were improved to reduce error.