CO544 - Machine Learning and Data Mining Project

Group 04

Department of Computer Engineering

Submitted to: Dr. Damayanthi Herath

CO 544- Machine Learning and Data Mining

Department of Computer Engineering

University of Peradeniya

Group Members

E/15/180 - Karunathilake V.M.B.S.S.V

E/15/243 - Nisansala R.M.B.S

E/15/271 - Prasadika L.B.S.

Created on : 11/05/2020

Table of contents

1. Abstract

In this project we were asked to experiment with a real world dataset, and to explore the application of machine learning and data mining to solve a classification problem. Machine learning algorithms can be used to find the patterns in data. We were expected to gain experience using python language and python libraries, and were expected to submit a report about outlining the work in the project. After performing the required tasks on the dataset given, herein lies in the final report.

2. Introduction

"Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions."[1] As data sources proliferate along with the computing power to process them, going straight to the data is one of the most straightforward ways to quickly gain insights and make predictions.

Machine Learning can be thought of as the study of a list of sub-problems, viz: decision making, clustering, classification, forecasting, deep-learning, inductive logic programming, support vector machines, reinforcement learning, similarity and metric learning, genetic algorithms, sparse dictionary learning, etc. Supervised learning refers to a class of systems and algorithms that determine a predictive model using data points with known outcomes. In Supervised learning, we have a training set, and a test set. The training and test set consists of a set of examples consisting of input and output vectors, and the goal of the supervised learning algorithm is to infer a function that maps the input vector to the output vector with minimal error.

There are many different types of classification algorithms for modeling classification predictive modeling problems. There is no good theory on how to map algorithms onto problem types; instead, it is generally recommended that a practitioner use controlled experiments and discover which algorithm and algorithm configuration results in the best performance for a given classification task. There is no single algorithm that works for all cases, as merited by the No free lunch theorem[2]. In this project, we try and find patterns in a dataset to make predictions with more accuracy.

---

[1] "What is machine learning? Types of machine learning. - Medium." Accessed May 13, 2020. https://medium.com/@meetpatel12121995/what-is-machine-learning-types-of-machine-learning-31b5fbbda66b.

[2] "No free lunch theorem - Wikipedia." Accessed May 13, 2020. https://en.wikipedia.org/wiki/No_free_lunch_theorem.

3. Dataset

There are 15 attributes in the data set. The attributes could be continuous, nominal with small numbers of values, and nominal with larger numbers of values. The dataset that was used for this project is a subset of a much larger dataset, and has the following feature vectors: The attributes details are as follows.

A1: b, a
A2: continuous
A3: u, y, l
A4: g, p, gg
A5: continuous
A6: c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff
A7: continuous
A8: TRUE, FALSE
A9: v, h, bb, j, n, z, dd, ff, o
A10: continuous
A11: TRUE, FALSE
A12: continuous
A13: TRUE, FALSE
A14: continuous
A15: g, p, s
A16: Success,Failure (class attribute)

4. Explorations Conducted on features

Machine Learning algorithms are completely dependent on data because it is the most crucial aspect that makes model training possible. On the other hand, if we won't be able to make sense out of that data, before feeding it to ML algorithms, a machine will be useless.

Data Exploration is used to understand, summarize and analyse the contents of a dataset, usually to investigate a specific question or to prepare for more advanced modeling.

4. 1. Data exploration steps

Below are the steps involved to understand, clean and prepare the data for building our predictive model.

1. Variable Identification
2. Univariate Analysis
3. Bi-variate Analysis
4. Missing values treatment
5. Outlier treatment
6. Variable transformation
7. Variable creation

Finally, we need to iterate over steps 4 – 7 multiple times before we come up with our refined model.

- **Variable Identification :** First, identify **Predictor** (Input) and **Target** (output) variables. Next, identify the data type and category of the variables. In our case dataset target column A16 and A1-A15 are Predictor. Then we should identify the type of variable, data type and the data type category.
- **Univariate Analysis** :For continuous variables, it helps us find mean, median, mode, min, max. For categorical variables, we use a frequency table to understand distribution of each category. Univariate analysis is also used to highlight missing and outlier values.[3]
- **Bi-variate Analysis:** While doing bi-variate analysis between two continuous variables, we should look at scatter plots. The relationship can be linear or nonlinear. To find the strength of the relationship, we use Correlation. Correlation varies between -1 and +1.There are different models which determine correlation between types of variables:

Feature/Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

[3] "Data Preparation and Exploration - Towards Data Science." Accessed May 14, 2020. https://towardsdatascience.com/data-preparation-and-exploration-5e09b92cf00e.

4. 2. Missing Value Treatment

Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification. Occurrence of these missing values may occur at two stages:

1) **Data Extraction:**
   It is possible that there are problems with the extraction process. In such cases, we should double-check for correct data with data guardians. Some hashing procedures can also be used to make sure data extraction is correct. Errors at the data extraction stage are typically easy to find and can be corrected easily as well.
2) **Data collection:**
   These errors occur at time of data collection and are harder to correct. They can be categorized in four types:

- Missing completely at random:
  This is a case when the probability of missing a variable is the same for all observations.
- Missing at random:
  This is a case when a variable is missing at random and missing ratio varies for different values / levels of other input variables.
- Missing that depends on unobserved predictors:
  This is a case when the missing values are not random and are related to the unobserved input variable.
- Missing that depends on the missing value itself:
  This is a case when the probability of missing value is directly correlated with missing value itself.

There are various ways to deal with each of these missing values. It is better to get rid of all those nasty NaN values as soon as possible. There are some methods to handle missing values.

- **Deleting entry**, this is where we can remove the missing/null values by removing the entire row of data but also in the process lose quite a lot of data which could create serious amounts of variance.

- **Mean/ Mode/ Median Imputation,** we can replace the missing/null values with either of 3 M's depending on the possible values of the given column. The method consists of replacing the missing data for a given attribute by the mean or median (quantitative attribute) or mode (qualitative attribute) of all known values of that variable.
- **Prediction model,** here we divide the dataset into two datasets, one with no missing values(to train) and one with missing values. This helps us figure out the values to be introduced in the dataset with missing values. This model can still be less precise in cases where there is no relation between missing values and the attributes in the dataset.

4.3. Steps done to process data

- In our data set we have replaced all the missing values with NaN.
  dataset.replace('?', np.nan, inplace=True) ;

- Then replaced missing values of non integers with high frequency values and we used Mean/ Mode/ Median Imputation to replace missing values of integers & floats.

  For an example, if the high frequency value for column A1 is b, then the missing values of that column would be filled as follows.

  dataset = dataset.fillna({"A1": "b"})

- In machine learning, we usually deal with datasets which contain multiple labels in one or more than one columns. These labels can be in the form of words or numbers. To make the data understandable or in human readable form, the training data is often labeled in words. Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important preprocessing step for the structured dataset in supervised learning. The following code shows how we encoded the categorical data in our dataset.

  Ex: if it is for column A1

  lb_make = LabelEncoder()

  dataset["A1"] = lb_make.fit_transform(dataset["A1"])

- Then We will split the loaded dataset into two, 80% of which we will use to train, evaluate and select among our models, and 20% that we will hold back as a validation dataset.

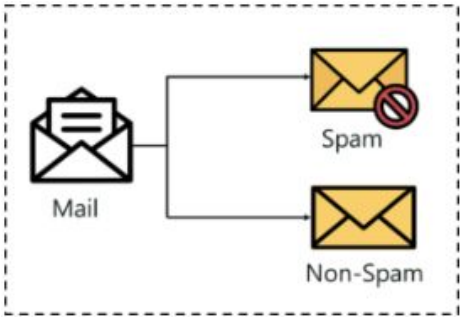
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.20, random_state=1)

- Then the training dataset is used to prepare a model, to train it and make predictions.

5. Classification

Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.

The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.



As an example heart disease detection can be identified as a classification problem, this is a binary classification since there can be only two classes i.e has heart disease or does not have heart disease. The classifier, in this case, needs training data to understand how the given input variables are related to the class. And once the classifier is trained accurately, it can be used to detect whether heart disease is there or not for a particular patient.[4]

[4] "Linear Regression Explained - Towards Data Science." 26 Jan. 2020, https://towardsdatascience.com/linear-regression-explained-d0a1068accb9. Accessed 15 May. 2020.

Since classification is a type of supervised learning, even the targets are also provided with the input data. Let us get familiar with the classification in machine learning terminologies.

5.1. Classification Terminologies

- **Classifier** – It is an algorithm that is used to map the input data to a specific category.
- **Classification Model** – The model predicts or draws a conclusion to the input data given for training, it will predict the class or category for the data.
- **Feature** – A feature is an individual measurable property of the phenomenon being observed.
- **Binary Classification** – It is a type of classification with two outcomes, for eg – either true or false.
- **Multi-Class Classification** – The classification with more than two classes, in multi-class classification each sample is assigned to one and only one label or target.
- **Multi-label Classification** – This is a type of classification where each sample is assigned to a set of labels or targets.
- **Train the Classifier** – Each classifier in sci-kit learn uses the fit(X, y) method to fit the model for training the train X and train label y.

In our project we used **binary classification** because we had two outcomes either Success or Failure.

5.2. Classification Algorithms

In machine learning, classification is a supervised learning concept which basically categorizes a set of data into classes. The most common classification problems are – speech recognition, face detection, handwriting recognition, document classification, etc. It can be either a binary classification problem or a multi-class problem too. There are a bunch of machine learning algorithms for classification in machine learning. Let us look at our classification algorithm in the machine learning project that is **Logistic Regression** which we used for the final approach of our project.[5]

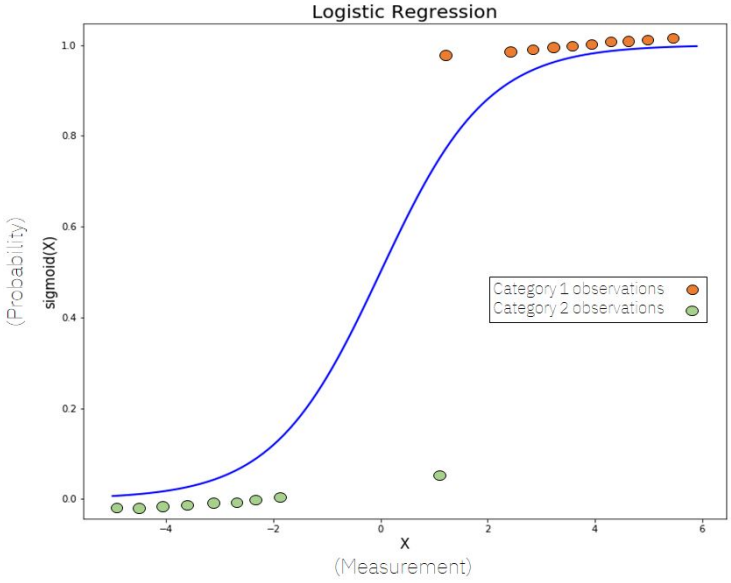
[5] "Machine Learning Classifiers - Towards Data Science." 11 Jun. 2018, https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623. Accessed 15 May. 2020.

6. Logistic Regression

In the Machine Learning world, Logistic Regression is a kind of parametric classification model, despite having the word '*regression*' in its name. And it is a very useful algorithm for binary classifications.

This means that logistic regression models are models that have a certain fixed number of parameters that depend on the number of input features, and they output categorical prediction, like for example if a plant belongs to a certain species or not.

In reality, the theory behind Logistic Regression is very similar to the one from Linear Regression. In Logistic Regression, we don't directly fit a straight line to our data like in linear regression. Instead, we fit a S shaped curve, called *Sigmoid*, to our observations.



First of all, like we said before, Logistic Regression models are classification models; specifically binary classification models (they can only be used to distinguish between 2 different categories — like if a person is obese or not given its weight, or if a house is big or small given its size). This means that our data has two kinds of observations (Category 1 and Category 2 observations) like we can observe in the figure.

Secondly, as we can see, the Y-axis goes from 0 to 1. This is because the *sigmoid* function always takes as maximum and minimum these two values, and this fits very well our goal of classifying samples in two

different categories. By computing the *sigmoid* function of X (that is a weighted sum of the input features, just like in Linear Regression), we get a probability (*between 0 and 1 obviously*) of an observation belonging to one of the two categories.
The formula for the *sigmoid* function is the following:

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

As a example If we wanted to predict if a person was obese or not given their weight, we would first compute a weighted sum of their weight and then input this into the *sigmoid* function.

1) Calculate weighted sum of inputs

$$x = \Theta \cdot weight + b$$

Weighted sum of the input features (feature in this example)

2) Calculate the probability of Obese

$$probability\ of\ obese(x) = \frac{1}{1 + e^{-x}}$$

Use of the sigmoid equation for this calculation

There are multiple ways to train a Logistic Regression model (fit the S shaped line to our data). We can use an iterative optimisation algorithm like ***Gradient Descent*** to calculate the parameters of the model (the weights) or we can use probabilistic methods like ***Maximum likelihood***.

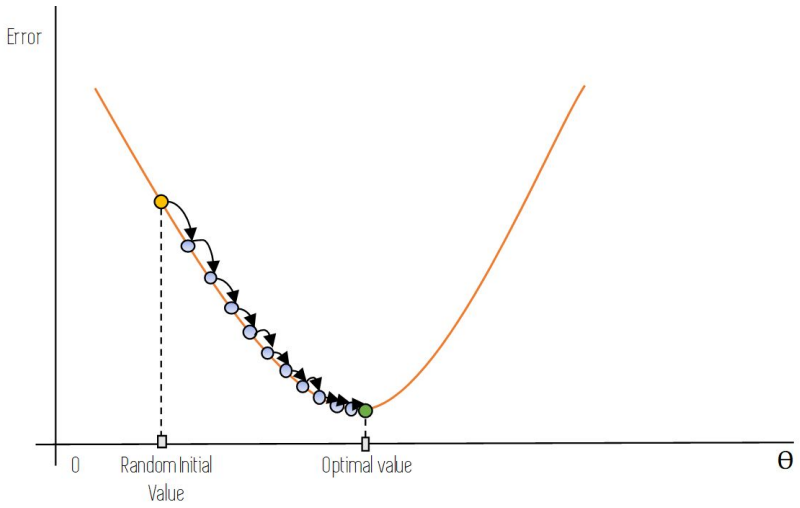
6.1. Training with Gradient Descent

Gradient Descent is an optimisation algorithm that can be used in a wide variety of problems. The general idea of this method is to iteratively tweak the parameters of a model in order to reach the set of parameter values that minimises the error that such a model makes in its predictions.

After the parameters of the model have been initialised randomly, each iteration of gradient descent goes as follows: with the given values of such parameters, we use the model to make a prediction for every instance of the training data, and compare that prediction to the actual target value.

Once we have computed this aggregated error (known as ***cost function***), we measure the local gradient of this error with respect to the model parameters, and update these parameters by pushing them in the direction of descending gradient, thus making the cost function decrease.[6]

The following figure shows graphically how this is done: we start at the orange point, which is the initial random value of the model parameters. After one iteration of gradient descent, we move to the blue point which is directly right and down from the initial orange point: we have gone in the direction of descending gradient.



Iteration after iteration, we travel along the orange error curve, until we reach the optimal value, located at the bottom of the curve and represented in the figure by the green point.

In practice, what happens when we train a model using gradient descent is that we start by fitting a line to our data (*the Initial random fit line*) that is not a very good representation of it. After each iteration of gradient descent, as the parameters get updated, this line changes its slope and where it cuts the y axis. This process is repeated until we reach a set of parameter values that are good enough (these are not

[6] "Linear Regression Explained - Towards Data Science." 26 Jan. 2020, https://towardsdatascience.com/linear-regression-explained-d0a1068accb9. Accessed 15 May. 2020.

always the optimal values) or until we complete a certain number of iterations. These parameters are represented by the green *Optimal fit line*.

Once we have used one of these methods to train our model, we are ready to make some predictions. Let's see an example of how the process of training a Logistic Regression model and using it to make predictions would go:

1. **First,** we would **collect a Dataset** of patients who have and who have not been diagnosed as obese, along with their corresponding weights.
2. **After this**, we would **train our model**, to fit our S shape line to the data and obtain the parameters of the model. After training using Maximum Likelihood, we got the following parameters:

Parameters:
θ = 1/12
b = - 7
Equation:
$X = \frac{1}{12} \times \text{weight} - 7$

Parameters and equation of X

3. **Now**, we are ready to **make some predictions**: imagine we got two patients; one is 120 kg and one is 60 kg. Let's see what happens when we plug these numbers into the model:

Patient 1; 60 Kg:

$$x = \frac{1}{12} \cdot 60 - 7 = -2$$

$$probability\ of\ obese(x) = \frac{1}{1 + e^{-(-2)}} = 0.119$$

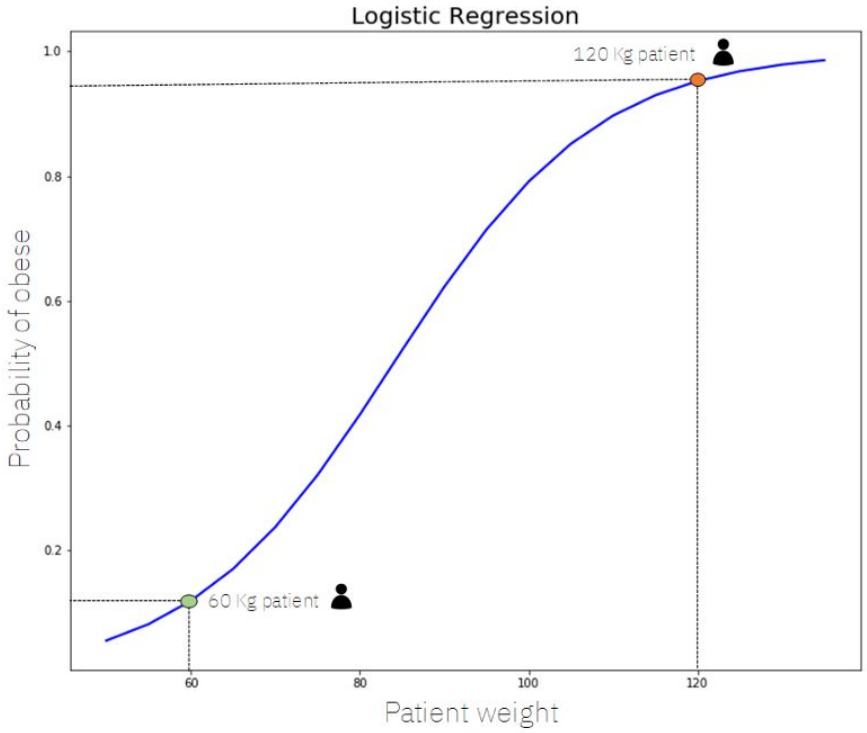
Patient 2; 120 Kg:

$$x = \frac{1}{12} \cdot 120 - 7 = 3$$

$$probability\ of\ obese(x) = \frac{1}{1 + e^{-3}} = 0.95$$

Results of using the fitted model to predict obesity given patient weight

As we can see, the first patient (60 kg) has a very low probability of being obese, however, the second one (120 kg) has a very high one.



Logistic Regression results for the previous examples.

6.2. Advantages and Disadvantages

Logistic regression is specifically meant for classification, it is useful in understanding how a set of independent variables affect the outcome of the dependent variable.

The main disadvantage of the logistic regression algorithm is that it only works when the predicted variable is binary, it assumes that the data is free of missing values and assumes that the predictors are independent of each other.

6.3. Use Cases

- Identifying risk factors for diseases
- Word classification
- Weather Prediction
- Voting Applications

7. Alternatives

Curious about why the data was behaving the way it was, we did use other classifiers on the said dataset. We have tried out several models which are Logistic Regression (84.35%), Linear Discriminant Analysis(84.13%), kNeighborsClassifier(68.70%), Decision Tree Classifier(80.94 %), GaussianNB(81.16 %) and SVC(51.69 %). The only thing we could now think of is that the input space was incomplete, and needed more dimensions for better predictions, and with the given feature vectors.

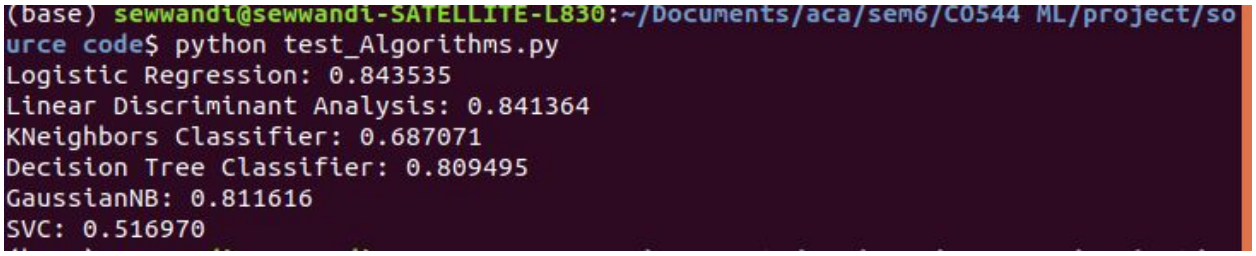


Figure: results of running testAlgorithms.py for training set

And all of them the highest accuracy was given by Logistic Regression. Therefore the model Logistic Regression was chosen as the final approach.

8. Results of final approach

For training data set;

Accuracy: 0.8288288288288288

Confusion_matrix: [[51 13]
[6 41]]

Classification_report:

	Precision	recall	fi-score	support
0.0	0.89	0.80	0.84	64
1.0	0.76	0.87	0.81	47
Accuracy			0.83	111
Macro avg	0.83	0.83	0.83	111
Weighted avg	0.84	0.83	0.83	111

9. Conclusion

We conclude that the dataset is not a complete space, and there are still other feature vectors missing from it. What we were attempting to generalize is a subspace of the actual input space, where the other dimensions are not known, and hence none of the classifiers were able to do better than 84.13% (Logistic Regression) in the training set. Therefore more feature vectors need to be calculated so that the classifiers can form a better idea of the problem at hand.