# ECE532 - Final Project Report

**Lewis Francisco Handy-Cardenas***
Department of Mechanical Engineering
University of Wisconsin - Madison
Madison, WI
handycardena@wisc.edu

## Abstract

The focus of this project is to design and evaluate three different classifiers using three different machine learning algorithms (i.e. Linear Classifier: Least Squares, Truncated SVD, Neural Networks) to predict an online buyers intention.

## 1 Introduction

Machine learning techniques aid data analysis in different fields and comprise two general categories of problems, some relying on supervised learning and others unsupervised learning. The data analysis of a given dataset can be broken into five steps: Data colletion, Pre-processing, Feature extraction, Training data collection, Machine learnig algorithm implementation.

One could say that optimization is a sixth step essential to a proper project development. A very typical algorithm used nowadays is a regression based one, in which data is classified shaping boundaries with respect to feature distribution (i.e. Least Squares to solve Binary Classifier with a linera boundary). Different datasets with different attributes and distributions can be tackeld with different algorithms, this projects explores the performance of three different algoithms: Least Squares – Linear Regression, Truncated Singular Value Decomposition (SVD), and multi-layered Neural Networks.

The goal is to determine which algorithm is better to solve the given classification data. Higher dimensional feature spaces require a more complex algorithm. When using regularizing parameters, the designer has to keep the variance and bias of the classifier in mind as well, as these can lead to inproper classification, or overclassification.The ultimate goal of this research is to optimize and benchmark the three algorihtms by classifying a dataset involving an online buyers intention.

### 1.1 Dataset

The dataset that was chosen to be used is called **online shoppers intention** and can be found under:

`https://www.kaggle.com/roshansharma/online-shoppers-intention?select=online_shoppers_intention.csv`

The 18attributes are comprise different data types. There are three attributes (Administrative, Informational, Product related) which represent the different types of pages that were visited by the customer in a single session. Each of these attributes has a corresponding attribute which represents the total time spent in each of the page categories (e.g Administrative Duration).

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

According to the data source, some of the attributes are obtained from “Google Analytics” metrics; these are the “Exit Rate”, “Bounce Rate” and “Page Value”. The rest of the attributes are self-explanatory, and are as follows:

- “Special day”. Indicates if the customer is viewing items online for potential purchase on a holiday day such as Christmas, Black Friday, Valentine’s day, etc.
- “Month”. Specifies the month of the year the customer started the session on.
- “Operating syste”. Specifies the browser used by the customer, for which a number is analogous to some known operating system (e.g. 1 →Windows)
- “Browser”. Specifies the browser used by the customer, for which a number is analogous to some known browser (e.g. 1 →Google Chrome).
- “Traffic type”. An attribute related to some internal measure for which a number is analogous to some known label.
- “Visitor type”. This specifies if the customer is a returning or new visitor.
- “Weekend”. This specifies if the user is shopping on a weekend (Saturday or Sunday) or during the week (Monday-Friday).The classification problem that will be explored is to predict if a user will purchase something onlinewhen browsing duringa session.

## 2 Preprocessing

Because of the string and Boolean data types values found in the dataset, research out of the scope of the course had to be performed in order to find an appropriate way to pre-process the data. For this reason, the Pandas libraries were used in the pre-processing step. Documentaoitn was extensive and easy to follow. Initially, singular matrices and errors were being encountered after the data processing step, it was later found that the data contained some missing entries which conflicted in the several matrix operations. These values were searched and taken out of the dataset, the attributes that were given in string values such as “day of the week” were recoded to be interpateed with a numerical value. The following figures include the datasettypes after the pre-rpocessing step, and tables with the before and after.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12330 entries, 0 to 12329
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Administrative           12316 non-null  float64
 1   Administrative_Duration  12316 non-null  float64
 2   Informational            12316 non-null  float64
 3   Informational_Duration   12316 non-null  float64
 4   ProductRelated           12316 non-null  float64
 5   ProductRelated_Duration  12316 non-null  float64
 6   BounceRates              12316 non-null  float64
 7   ExitRates                12316 non-null  float64
 8   PageValues               12330 non-null  float64
 9   SpecialDay               12330 non-null  float64
 10  Month                    12330 non-null  object
 11  OperatingSystems         12330 non-null  int64
 12  Browser                  12330 non-null  int64
 13  Region                   12330 non-null  int64
 14  TrafficType              12330 non-null  int64
 15  VisitorType              12330 non-null  object
 16  Weekend                  12330 non-null  bool
 17  Revenue                  12330 non-null  bool
dtypes: bool(2), float64(10), int64(4), object(2)
memory usage: 1.5+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12316 entries, 0 to 12315
Data columns (total 20 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Administrative           12316 non-null  float64
 1   Administrative_Duration  12316 non-null  float64
 2   Informational            12316 non-null  float64
 3   Informational_Duration   12316 non-null  float64
 4   ProductRelated           12316 non-null  float64
 5   ProductRelated_Duration  12316 non-null  float64
 6   BounceRates              12316 non-null  float64
 7   ExitRates                12316 non-null  float64
 8   PageValues               12316 non-null  float64
 9   SpecialDay               12316 non-null  float64
 10  Month                    12316 non-null  int64
 11  OperatingSystems         12316 non-null  int64
 12  Browser                  12316 non-null  int64
 13  Region                   12316 non-null  int64
 14  TrafficType              12316 non-null  int64
 15  Weekend                  12316 non-null  int64
 16  Revenue                  12316 non-null  int64
 17  V_New_Visitor            12316 non-null  uint8
 18  V_Other                  12316 non-null  uint8
 19  V_Returning_Visitor      12316 non-null  uint8
dtypes: float64(10), int64(7), uint8(3)
memory usage: 1.6 MB
```

Figure 1: Data types before (left) and after (right) Preprocessing step.

### 2.1 Data Splitting: Training Set and Test Set

The data was divided into training data and testing (i.e validation) data sets. For this, the sklearn librariers were used to automate the process. This way the user only needs to select the ratio by which they desire to divide data, and that training and testing sets are made. This process was later implmented using the numpy library functions to perform cross validation studies.

## 3 First Classifier - Least Squares

The least squares (LS) linear regression was implemented in order to verify performance on the classified data. Given the nature of the data, high dimensionality, and the number of features, a poor behaviour was expected. The hypothesis held true even at the training stage of the data.

The variance found in some of the features for specific users/individuals is very high, in some instances by a factor of over 10. This greately affects the classification of the data with a linear boundary, a more complex boundary (i.e. high order polynomial) is needed in order to properly weight the attributes that experience such conditions. The least squares linear solution follows the equation that is here presented.

$$W = (X^T X)^{-1} X^T y$$

Here, W represents the weight vector, X representes the feature matrix (e.g. training data), and y represents the training labels corresponding to those given features. At its best, the classifier performed only withing $\sim 10\%$ of the correct data classification. This value was worse when looking at the predicted values and missclassifications for the test data. Given the poor nature and performance of the linear classifier, no optimization was further performed as it was shown that at its best, data classification of $\sim 10\%$ was as good as it got.

## 4 Second Classifier - Truncated SVD

Because of the shape of the data, the truncated singular value decomposition represented an attractive algorithm to implemente for classification. By using the truncated version of the SVD decomposition, the computational time and speed required by the code to run the functions diminishes considerably. This method relies on finding the eigen values which are particular to the data, and sets the groundwork for other algorithms such as PCA (Principal Component Analysis) which results helpful in cases where there is big variance in the data. It was found that the Truncated SVD on its own was not much better at creating a proper weight vector to classify the data. At its best, the truncated SVD performed shy of the $\sim 10\%$ accuracy.

Cross-validation techniques were explored with these across multiple combinations of training datasets derived from the total dataset, which representeded $25\%$ of the data. Larger training datasets were computationally demanding to loop through during the cross validation optimization, hence the chosen $25\%$. Even thought it was found that the Truncated SVD was not much better than the linear regression least squares model, this still remains a better estimator given it sets the basis for other algorithms such as PCA which with a Regularizer, which could improve performance of the data classification.

## 5 Third Classifier - Neural Networks

Neural Network Optimization

| Hidden Layers [dim] | Epochs [dim] | Alpha [dim] | Training Sample Size [% of total] | Number of Missclassifications [dim] | Mean Squared Error [%] |
|---|---|---|---|---|---|
| 2 | 10 | 0.1 | 10 | 185 | 15.03% |
| 2 | 20 | 0.2 | 10 | 185 | 15.03% |
| 2 | 10 | 0.1 | 30 | 582 | 15.76% |
| 3 | 15 | 0.3 | 15 | 280 | 15.01% |
| 3 | 15 | 0.1 | 20 | 388 | 15.75% |
| 3 | 15 | 0.2 | 20 | 388 | 15.75% |

Figure 2: Neural Network Optimization variation of parameters and calculated error.

The neural network classifier was built around the activation function $\sigma(z) = (1 + e^{-z})^{-1}$. Three parameters were varied in a randomized way, mostly limited by the computational time and speed given the training data size. Those parameters include the number of iterations (i.e. epochs), the number of hidden layers in the network (2,3), and the value for sigma. The value for sigma was left constant to optimize around the number of iterations and number of hidden layers in the network. Something important to note is that contrary to the other two classifiers, the training data had to be

limited to $\sim 25\%$ of the total. This was mainly due to the fact that the computational time would otherwise greately increase, demanding long wait times to properly optimize around values.

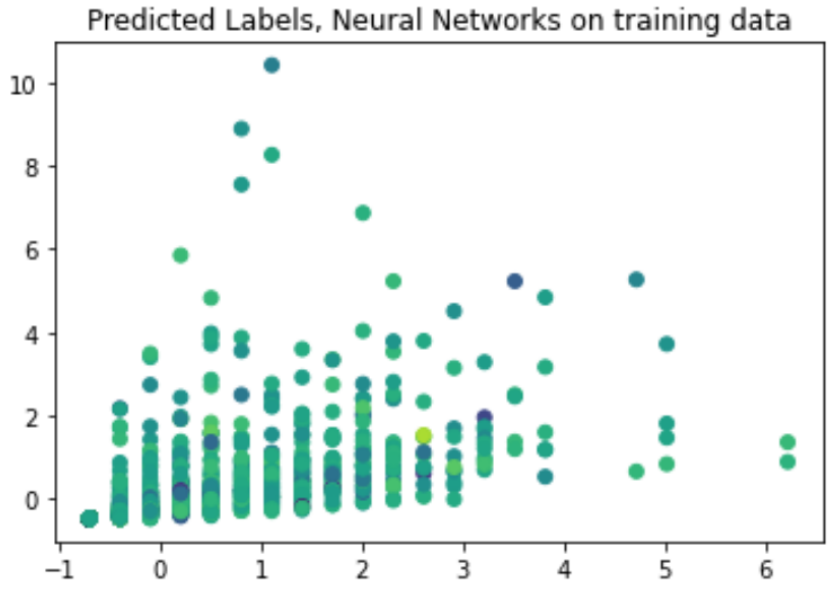


Figure 3: Classified data with optimized predicted weight values from Neural Network algorithm. Predicted labels for feature 1 and feature 2. Yellow colored data points represent correctly classified data, blue colored data points represent incorrect classified data.

The attached table inclues the number of missclassifications with each parameter variations. Becuase of the complexity of the computation, a looped iteration over a range of values for the neural network was not possible. The iterations had to be run individually modifying the parameters to the desired values. The attached plots comprise the results from the optimization. The figure legends detail the differences, and represent the visualization corresponding to the optimization values.

## 6 Analysis and Conclusions

Summarizing the results, the following was found:

- For Neural Networks a $\sim 15\%$ mean squared error was found for a dataset of $15\%$ to $30\%$ training samples and an alpha of $0.1$, which didn´t seem to change the results much with the chosen variation presented in the table.
- For LSE, $30\%$ of the total data used for training samples yielded $93.95\%$ mean squared error (really bad).
- For Truncated SVD, $30\%$ of the total data used for training samples yielded $91.39\%$ mean square error (bad but better), and the cross validation showed a performance of up to $16.62\%$ error (equivalent to $\sim 84\%$ mean squared error) on the training set, which comprised $20\%$ of the total dataset.

It was very clear that a linear boundary does not do well at classifying the dataset in study. A higher dimensional boundary is required; the next step in optimizing the least squares classifier and Truncated SD would be to include a more complex boundary. In the current study, it was found that both of these were outperformed by the neural networks, which showed classification accuracies of over $85\%$ on the training set. More time needs to be spent optimizing all three classifiers to find their true potential, but the main takeaway is that the variance found within the different feature vectors difficulty sets a clear boundary for lots of middle points within each of the features. It would also be important to evaluate the performance of the neural network classifier closer to discard any possibility

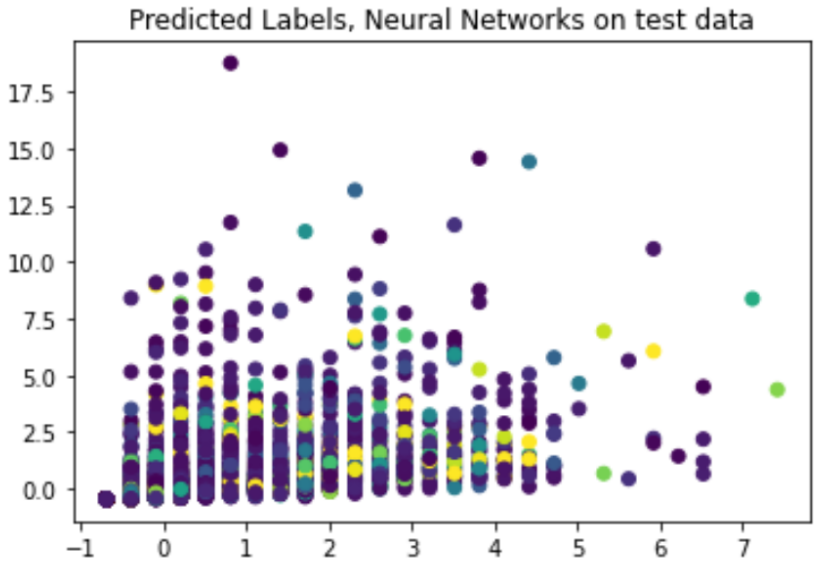


Figure 4: Classified data with optimized predicted weight values from Neural Network algorithm. Predicted labels for feature 1 and feature 2. Yellow colored data points represent correctly classified data, blue colored data points represent incorrect classified data.

of overfitting; which would diminish the total accuracy of the algorithm which was found within this study.

A link to the project files(including updates)can be found under the following links:

`https://github.com/handycardena/ECE532_Final_Project_Handy`

`https://github.com/handycardena`

The file called **Final Compiled** contains the python code, figures and results here presented for the first and second classifiers.

The file called **Neural Networks Final** contains the python code, figures and results here presented for the third classifier.