**Regression**

Tiati Thelen

Department of Computer Information Sciences, Minnesota State University, Mankato

CIS 677: Individual Studies

Dr. Guarionex Salivia

April 3, 2022

**Contents**

Introduction .….…………………………………..……………………………………………… 3

Linear Regression ……………...…….………………………………….………………………. 6

Random Forest ………………………………………………………….……………………….. 6

Gradient Boost .…….…………...……………………………………….………………………. 7

Results …………………………….…………….……………………….………………………. 8

References ...……………………….…………………………………….……………………... 10

# Introduction

# Machine Learning problems can be broken down into classification and regression. Regression problems have a goal to predict a target numeric value given a set of features. For this paper, the goal is to accurately predict how many bikes will be rented based on different predictors i.e. weather. The data set found was provided by (Fanaee, 2013). Three algorithms are implemented and evaluated to determine the best model for the dataset. Linear Regression, Random Forest, and Gradient Boost will be compared in terms of mean squared error (MSE), root mean squared error (RMSE), and R2. All data cleaning, algorithm information, and algorithm implementation was provided through the textbook *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (Géron, 2019).

The data is prepared by removing unnecessary variables. Specifically, the columns ‘casual’, ‘registered’ and ‘instant’ will be removed since it contains irrelevant data. These variables hold redundant information or repetitive information. Another data cleaning was to consider null values. Luckily, there were no null values within this dataset.

A picture containing table

Description automatically generated

Figure 1: Drop Unnecessary Variables

The next data cleaning steps were changing variables to the right type. For this data a handful of variables were changed into type category. This is recommended through the textbook and visualized in the figure below.

Table

Description automatically generated

Figure 2: Binary Values

Next, there is a case of multicollinearity occurring. When the correlation matrix is calculated the variables ‘atemp’ and ‘temp’ are highly correlated. Therefore, the variable ‘temp’ will be dropped. The figure following shows the code executed for this.

Graphical user interface, application, table

Description automatically generated

Figure 3: Multicollinearity

Lastly, before starting the model training, the variables were renamed to reduce confusion. Therefore, the y training and testing variables will hold the count of the bike rental for that day. I chose to have 33% of the data hidden to be held as testing data. This hidden data will be the consistently the same set across all three models. The figure below shows the implementation of this data split. I did implement this through a module in the sklearn library.

Graphical user interface, text, application

Description automatically generated

Figure 4: Training and Testing Data Sets

**Linear Regression**

Linear Regression is an approach for modelling data of a scalar response and on or more explanatory variables. For this project’s case it will be simple linear regression since it has one explanatory variable. This project did consider Logistic Regression, but it fell short in comparison. Linear Regression aims to find a straight line against the features to make accurate predictions (Al-Masri, 2021). This algorithm can extend further to create a function to predict on the test dataset. Linear Regression algorithm makes a prediction by computing a weighted sum of the input features. It will also add on a bias term. The function it creates is provided,

.

Theta represents the model’s parameter vector and x is the feature vector. As mentioned previously, the model will be trained to find the value of theta that finds the smallest RMSE value.

The complexity of this algorithm is dependent on which solver is used. In general, let n be the number of observations and p be the number of weights. It can be concluded that the complexity will be n2p + p3 (www.TheKernelTrip.com, 2018). It should be noted that when using Scikit library (which is being done), it seems to be a closer time complexity to 𝑛0.72𝑝1.3. It should be notes that the system could be solved using gradient descents.

**Random Forest**

Radom Forest was designed to be an ensemble of Decision Trees. It is trained usually by the bagging method. The bagging method is used during training by randomly selecting with replacement a sample of the training set. It then fits trees to these samples. It is done by a set number of times. After training, the predictions are then averaged from all individual regression trees.

When using Random Forest for regression tasks, the average prediction of individual trees is returned. Random Forests are advantageous because they correct for decision trees common problem of overfitting the training set. They usually perform better than decision trees but does have a lower accuracy than gradient boosted trees. An important key is that deep trees can detect irregular patters, yet they overfit training sets (low bias but high variance). Since Random Forests use averages of many deep trees, it has a goal of reducing variance. It will boost the overall model fit. In general forests pull together a team of decision trees to improve the overall performance of a singular tree.

The book explains to use a Radom Forest Classifier class. This is a more convenient and optimized class for Decision Trees. This algorithm introduces extra randomness when it grows trees. It does this since it does not search for the very best feature, but it searches for the best feature among a random subset of features. This process creates a more diverse tree that will trade a higher bias for a lower variance. It will yield a generally better model this way.

**Gradient Boost**

Gradient Boost is designed to sequentially add predictors to an ensemble. The form of ensembles is from weak prediction models, generally decision trees. Each addition will correct its predecessor. It will try to fit a new predictor to the residual errors made by the previous predictor. It is generalizable because it allows for optimizing different loss functions.

A problem to consider is overfitting the training set. This will lead to losing the generalization ability from the model. Different regularization techniques can be used to reduce the overfitting. One way to regularize is by setting the number of gradient boosting iterations. Increasing it will reduce the error on the training set. It could also lead to overfitting if increased too far. To find an optimal value, one can monitor the prediction error on different validation data. This is just one example of a way to fix this common issue. Another example is to reduce the number of trees if applicable. The higher the value the potential to overfit occurs again.

**Results**

The following figures show the code and implementation of the previously stated algorithms. These algorithms were compared in terms of accuracy of the hidden test data set. It can be concluded that linear regression performed the worst. It has a low R2 value in comparison to the others. The algorithm that did perform the best was Random Forest. It has the highest R2 value. It also has low RMSE and MSE values in comparison to Gradient Boost. Therefore, it can be concluded it fit the data best.

Graphical user interface, text, application

Description automatically generated

Figure 5: Linear Regression Results

Graphical user interface, text, application, email

Description automatically generated

Figure 6: Random Forest Results

Graphical user interface, text, application, email

Description automatically generated

Figure 7: Gradient Boost Results

**References**

Al-Masri, A. (2021, August 3). *How does linear regression actually work?* Medium. Retrieved April 10, 2022, from https://towardsdatascience.com/how-does-linear-regression-actually-work-3297021970dd

*Computational complexity of machine learning algorithms*. The Kernel Trip. (2022, February 25). Retrieved April 10, 2022, from http://www.TheKernelTrip.com/

Géron A. (2019). *Hands-on machine learning with scikit-learn and tensorflow concepts, tools, and techniques to build Intelligent Systems*. O'Reilly.