Research paper on the subject of Multiple linear regression and K-nearest neighbors

Nemanja Milosavljevic
University of Novi Sad Faculty of tehnical science
Faculty of tehnical science, FTN
Novi Sad, Serbia
nemanjam2008@gmail.com

Predrag Falcic
University of Novi Sad Faculty of tehnical science
Faculty of tehnical science, FTN
Novi Sad, Serbia
falcicp@gmail.com

Predrag Kaljevic
University of Novi Sad Faculty of tehnical science
Faculty of tehnical science, FTN
Novi Sad, Serbia
pkaljevic01@gmail.com

Abstract: In this paper we will use Multiple linear regression and K-nearest neighbor to predict the grade of student based on some other features. With the given features like, sex, address, family size and many others, we will predict the final grade. This problem can be solved on many ways, but we will use linear regression and k-nearest neighbors. Today it is more and more common to train a program to be able to predict some values for us, it is known as Machine learning. Machine learning has several very practical applications that drive the kind of real business results, such as time and money savings, that have the potential to dramatically impact the future of many organizations. Multiple linear regression and K-nearest neighbor are used to solve our particular problem, because we have more than one variable that has impact on our predicted value, in this case it is students grade. In this paper we will see that both of this methods will give almost the same result, with K-nearest neighbor being a little better with its prediction of the grade. The two algorithms will bot get the same dataset and predict the grade. Than we will measure the mean squar error and compare them to see wich one made the better prediction.

Keywords: regression, linear, knn, predicton.

I. INTRO

The problem described in this document is determining the level of influence of different factors describing a student, over his final grade. We have a dataset of students, each being described by 26 attributes, as well as their final grades. Using that information, the goal is to find out how each and every one of those attributes influences the grade of the student. Idea is to weight the attributes according to the amount of impact on the student's grade. The bigger the impact of a given attribute over student's final grade is, higher will its priority be when trying to predict the grade of a student with an unknown grade.

Managing to achieve a respectable accuracy in final grade prediction based on this information about a student, could prove to be very useful and helpful in education. For example, students whose grades are predicted on the lower end of the spectrum, could be given additional tutoring in their studies to try offsetting their predicted difficulties.

Two different algorithms were used to predict the final grade: Multiple Linear Regression and K-Nearest Neighbour. Both will be explained in detail later in the document.

II. MACHINE LEARNING

Maching learning is a type of artificial intelligence also known as AL that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data. The process of machine learning is similar to that of Data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension, as is the case in data mining applications, machine learning uses that data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervised. Suprevized algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from dataset.

For example, Facebook’s feed uses machine learning to personalize each member’s feed. If a member frequently stops scrolling in order to read or like a particular friend’s posts, the News Feed will start to show more of that friend’s activity earlier in the feed.

Machine learning is closely related to computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical oprimization, which delivers methods, theory and application domains to the field. Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction.

Machine learning also has intimate ties to optimization, many learning problems are formulated as minimization of some loss function on a training set of examples.

III. Simple and Multiple linear regression

Linear regression is an approach for modeling the relationship between a dependent variable, in our case that is student's grade, and one or more independent variables marked as X. X in our case is a list of the following features, sex, address, family size and others. The case of one independent variable is called Simple linear regression. Simple linear regression we used to solve this problem by so called brute force. Except of Simple linear regression there is one more type that we will focus on in this paper and it is called Multiple linear regression. In this paper we will more focus on multiple linear regression than simple regression.

Multiple linear regression attempts to model the relationship between two or more expmanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y. We can use multiple linear regression to find interesting relations between data that might not be obvious on first sight.

In most problems, more then one predictor variable will be available. This leads to the following multiple regression function: **F(X|Y) = a + b1*x1 + b2*x2 + … + bi*xi**, where a is called the intercept and the bi are called slopes or coefficients. Later in this paper when we discuss how we solved the problem, we will give an example how thi equation looks.

IV. K-Nearest neighbor

K-Nearest Neighbour (KNN) is a classification algorithm that basically involves making a decision. Which group does an element belong to? We find that out by looking around for it's closest neighbours. First we decide how many of it's closest neighbours are we looking for. Once we choose how many, then we look for them. When they have been found, we check which group does each of them belong to. Group in which the majority of neighbours belongs to, will be the group that the original element will be placed in.Choosing the value of k (number of neighbours to look for) is another important thing to note. If a problem consists of 2 classes, k should be an on odd number. This is done in order to prevent situations where there's an equal number of neighbours belonging in both classes, which will confuse the algorithm. K shouldn't also be a multiple of the number of classes, for the same reason. One of the main drawbacks of the KNN algorithm is the potential complexity in searching the nearest neighbour for each sample. When the dataset is very large, with a large amount of attributes for each sample, we have to calculate the distance based on each of those attributes.Even though K-Nearest Neighbour is a classification algorithm, it can be used in both classification and regression predictive problems. In KNN regression, it is used for estimating continuous variables. For example, we are trying to predict the value of an element. Again, we look for a set amount of nearest neighbours. After finding all of the nearest neighbours, we calculate their average value, and set the element to that value. It's worth mentioning that simply increasing the amount of nearest neighbours looked for won't necessarily give better results. There is usually a number of neighbours looked for after which the accuracy stops increasing, and starts getting worse. Searching for only one neighbour will just copy that neighbour's value to the element, while searching for all elements is just going to calculate the mean.

V. Solving a specific problem

In our problem, we have a dataset, it is saved in a csv file, and it is call train.csv. The problem can be solved using linear regression, regularization and some non-parametric approach. We decided to use linear regression, to be more specific we used simple linear regression, that gave us the highest mean square error, we also used multiple linear regression and knn that gave us better results.

The following table Table 1 **Description of dataset** will show data from the dataset.

Attribute	Description	Values
sex	Sex (male, female)	Binaries: F,M
age	ages	Numerical: [15, 22]
Address	Address(Ural, urban)	Binaries: U, R
famsize	Number of family members (<=3, or >3)	Biranaries: LE3, GT3
Pstatus	Parents live together or not	Binary: T, A
Medu	Mother education	Numerical:0 – 4
Fedu	Father education	Numerical: 0 -4
Reason	Reason for choosing that school	Nominal: home, reputation, course, other
Guardian	Mother, father, other	Nominal: mather, father, other
Traveltime	Time to travel to school	Numerical: 1-4
Srudytime	Number of hours spent learning	Numerical: 1-4
Failures	Number of time listening to the same course	Numerical: 1-4
Schoolsup	Addiotional classes	Binary:yes, no
Famsup	Family support	Binary:yes, no
Paid	Paid extra hours	Binary:yes, no
Activiries	Out of school activiry	Binary:yes, no

Higher	Want to go on faculty	Binary:yes, no
Internet	Has access to internet	Binary:yes, no
Romantic	In a relationship	Binary:yes, no
Famrel	Family relationship	Numerical: 1-5
Freetime	Amount of free time	Numerical: 1-5
Goout	How frequent he is going out with friends	Numerical: 1-5
Dalc	The amount of alcohol taken	Numerical: 1-5
Walc	The amountof alcohol taken on weekends	Numerical: 1-5
Health	Current health	Numerical: 1-5
Absences	Number of absence from class	Numerical: 0-93
Grade	Students grade	Numerical: 0-20

Table 1 Description of dataset

The problem was originally solved using Simple Linear Regression. Basically, going through all of the student's attributes, and finding out which one of them has the highest impact on the final grade. After that attribute is found, it is used for prediction.

First, data needs to be prepared for calculation. Binary attributes that have one out of two values, are numbered as 1 or 0, depending which of those two values they have. It's similar with attributes that have more than 2 values, they are numbered 0,1,2,3,4 etc. For attributes that have continuous values in a given range, their value is used as it is given, without the need to be changed. After the data has been passed through the algorithm, we calculate the rmse (root mean squared error) by substracting the actual value for each student's grade from the predicted value. We sum the squares of all errors, and divide them by the amount of dataset samples, in order to find the average prediction error.

Using this, we found out that the most impactful attribute is the number of classes that the student has failed. When predicting students' final grades based solely on that one attribute, we have managed to achieve the error of 2.99716141918.

As expected, this method cannot be as accurate as using all of the attributes during prediction like its done in Multiple Linear Regression and K-Nearest Neighbour algorithms.

From the above example we see that we used brute force to solve our problem. Now let's check out how we can even faster and a little better solve the same problem using Multiple linear regression.

```
# Read all data from csv
all_data = load_data('data/train.csv')
# Format the data to only have numbers no strings in it
formated_data = format_data(all_data)
# Drop the grades (Grade) column from data set
X = formated_data.drop('Grade', axis=1)
```

Very importan think is to split the dataset to training data and test data, we did it using the built in sklearn method, like this:

```
# Split the data in trainig and test sets
X_train, X_test, Y_train, Y_test = train_test_split(
    X, formated_data.Grade, test_size=0.10, random_state=5)
```

After this we create our model for linear regression and then we fit the data using the fit method from sklearn library:

```
# Create a LinearRegression model
lm = LinearRegression()
# Fitting a linear model
lm.fit(X_train, Y_train)
```

From the fit method we get the intercept and coefficients for our equation. Next step is to predict the grades for our students. After that we calculate the predictive error with RMSE – Root mean square error.

```
print("RMSE for multi linear regression ", math.sqrt(mean_squared_error(Y_test,lm.predict(X_test))))
```

Now, if we plot the data got from this functions we get the following graph Figure 1:

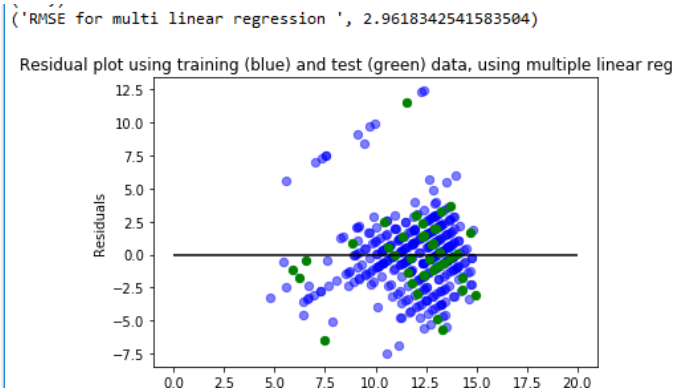

Figure 1 Residual plot for multiple linear regression

Residual plots are a good way to visualize the errors in your data. If you have done a good job then your data should be randomly scattered around line zero. From the picture we can see that our RMSE is aroung 2.96.

Now we do the similar steps but we create the model as KNeighborsRegressor. When we plot the data got from KNN we get this diagram Figure 2

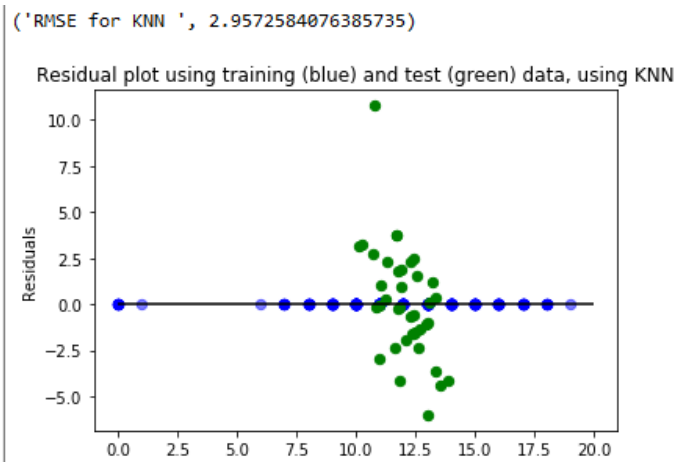

Figure 2 Residual plot for KNN

From the picture we can see that RMSE is lower then the previous two, which makes K-nearest neighbor better method to use.

We would like to mention that we used the cross_val_score algorithm from sklearn, to derminate the best K for our KNN algorithm, see the following picture Figure 3.

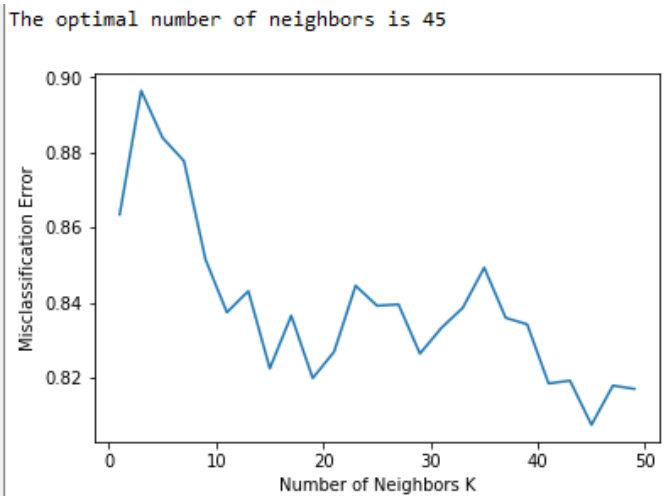

Figure 3 Graph for the best K

Here we see that the algorithm found the best K (45) for our problem.

VI. Conclusion

In this particular case, the accuracy of prediction didn't change drastically when using algorithms that predict the student's final grade based on all attributes, coompared to predicting solely on the highest impact attribute, the number of student's class failures. The RMSE achieved using Simple Linear regression was calculated at approximately 2.997. Incorporating all of the attributes, the error was reduced just slightly 2.961 using Multiple Linear Regression. Best results were achieved using the K-Nearest Neighbour algorithm, although still not significantly improved, at 2.957 RMSE.

VII. References

[1] http://statweb.stanford.edu/~susan/courses/s60/split/node60.html

[2] http://www.statisticssolutions.com/what-is-linear-regression/

[3] http://www.investopedia.com/terms/m/mlr.asp

[4] https://www.udacity.com/course/intro-to-machine-learning--ud120

[5] http://www.saedsayad.com/k_nearest_neighbors_reg.htm