APPLIED MACHINE LEARNING

Linear and Logistic Regression & Experimentations
Assignment#1

Professor : Muhammad Sabir

Aseem Mittal

| Net ID: axm179730

Used Student Performance dataset available for download at
https://archive.ics.uci.edu/ml/datasets/Student+Performance
Use only student-mat dataset.

Goal:

Implementation of the following two algorithms:

1. Linear regression
2. Logistic regression

Tasks:

1. Divide the dataset into train and test sets sampling randomly. Use only predictive attributes and the target variable (do not use non-predictive attributes). Also, do not use G1 and G2.

- Using the Pandas library, I have imported the data in Jupiter notebook and have converted the features with object datatypes to numeric/ int64 datatypes.
- I have then dropped the columns 'G1' and 'G2' as they were not required in our model.
- From ***sklearn.model_selection import train_test_split*** : I have divided the dataset into train and test sets sampling randomly by importing Scikit-learn's train_test_split library. I have split the train and test into 70/30 ratio. So the resulting X_train , X_test,y_train and y_test has shape as following:

```
cols = data.shape[1]
train, test = train_test_split(data, test_size=0.3,random_state=0)

X_train=train.iloc[:,0:cols-1]
y_train=train.iloc[:,cols-1:cols]

X_test=test.iloc[:,0:cols-1]
y_test=test.iloc[:,cols-1:cols]
```

```
In [12]: X_train.shape, X_test.shape, y_train.shape, y_test.shape
Out[12]: ((276, 31), (119, 31), (276, 1), (119, 1))
```

2. Use linear regression to predict the final grade (G3). Report and compare your train and test error/accuracy metrics. You can pick any metrics you like (e.g. mean squared error, mean absolute error, etc.)

- I have created functions for: **Linear regression, cost function, data preparation, gradient descent, and results for the model** to predict the final score (G3).
- The following are my findings, when run on all 30 features:

```
_____________________________________________________
Linear Regression | Running the algo on Train Data
Mean squared error = 13.4445
Mean Absolute error = 2.7894

_____________________________________________________
__
```

```
_____________________________________________________
Linear Regression | Running the algo on Test Data
Mean squared error = 25.3139
Mean Absolute error = 3.8819
_____________________________________________________
```

3. Convert this problem into a binary classification problem. The target variable should have two grade categories (Pass and Fail)

- Converted the Linear regression problem to *Logistic Regression (binary Classification)* problem for the grade categories to be Pass or Fail (1 or 0)
- I divided the data by using median as the baseline.

```
median = data['G3'].median()
data['G3'] = np.where(data['G3']>=median, 1, 0)
```

4. Implement logistic regression to carry out classification on this data set. Report accuracy/ error metrics for train and test sets.
   - Created cost function and gradient descent function for the Logistic Regression.
   - Used scipy.optimize.fmin_tnc to minimize the cost function of logistic regression
   - # *scipy.optimize.fmin_tnc* : Minimize a function with variables subject to bounds, using gradient information in a truncated Newton algorithm. This method wraps a C implementation of the algorithm.

```
result = opt.fmin_tnc(func=costFunction_LogisticRegression, x0=theta, fprime=gradientDescent_LogisticRegression, args=(X_train, y_train))

OptTheta = np.matrix(result[0])
```

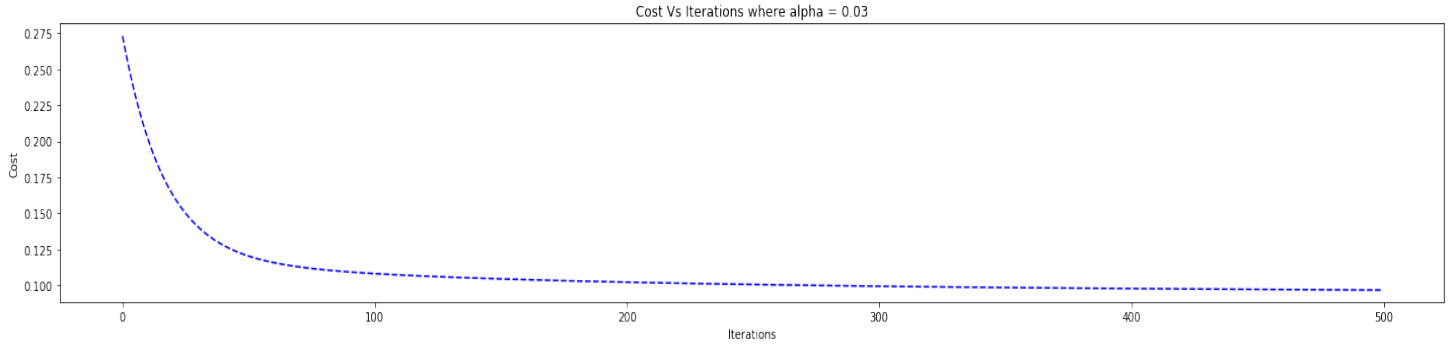
   - The finding of the Logstic regression model is as follows :-

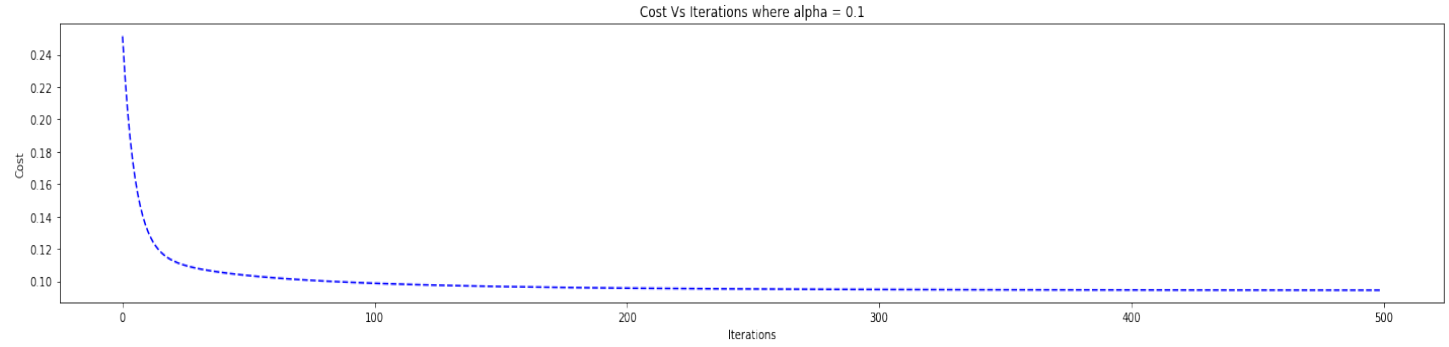
```
_____________________________________________________
Logistic Regression | Running the algo on Test set
Accuracy is 63.03 %
Confusion Matrix =
[[31 35]
 [ 9 44]]
Mean Absolute error = 0.3697
_____________________________________________________
_____________________________________________________
Logistic Regression | Running the algo on Train set
Accuracy is 70.29 %
Confusion Matrix =
[[ 68  52]
 [ 30 126]]
Mean Absolute error = 0.2971
_____________________________________________________
```

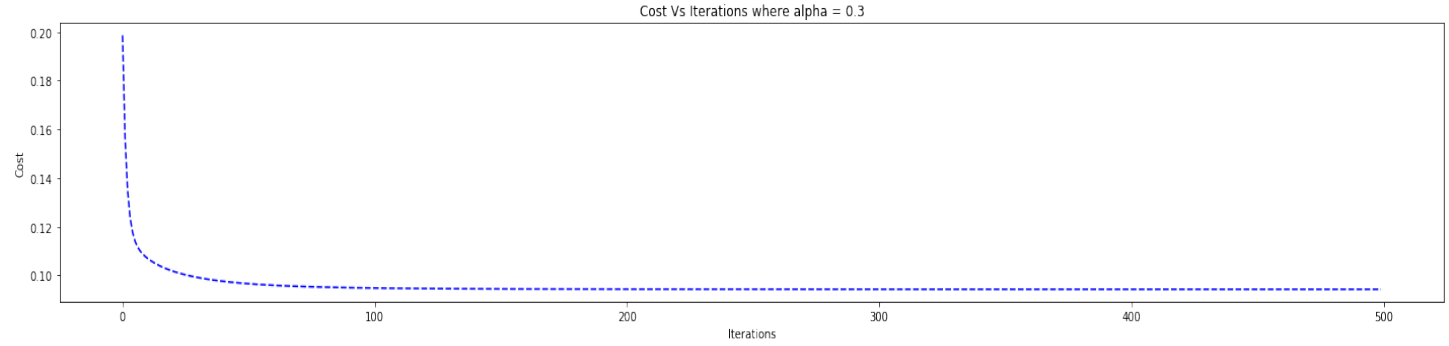
Experimentations:-

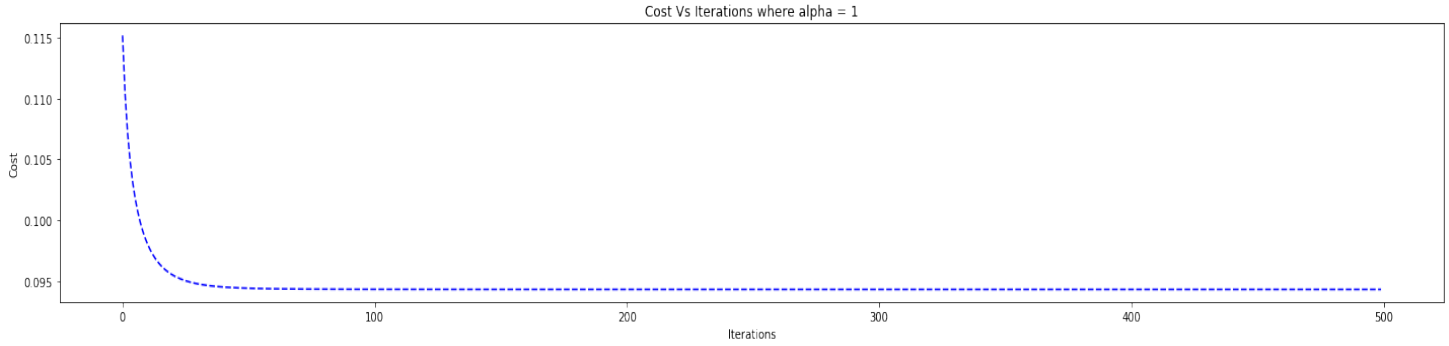
- ***Experiment 1*** **- Changing learning rate (alpha) :**

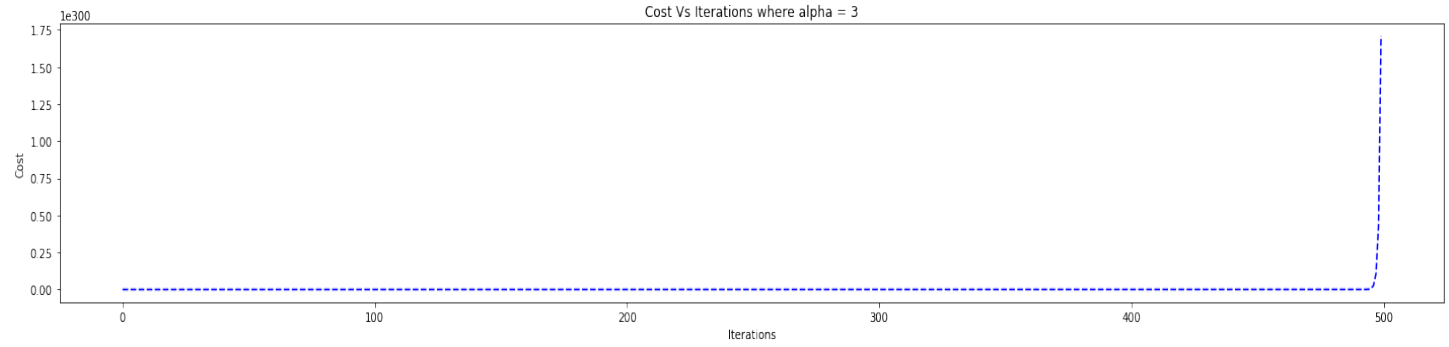
I have plotted cost vs Iterations using different learning rates:











As we can see from these plots, as alpha increases, the curve becomes steeper and able to reach lowest cost in less iterations but as alpha becomes greater than 1, cost goes to infinity because of too high alpha.

- ***Experiment 2 - Implement linear regression with 10 random features***

  I have used random 10 numbers using random sample function

```
X_train,X_test,y_train,y_test,theta =  data_preparation(data,random.sample(range(0,29),10))
```

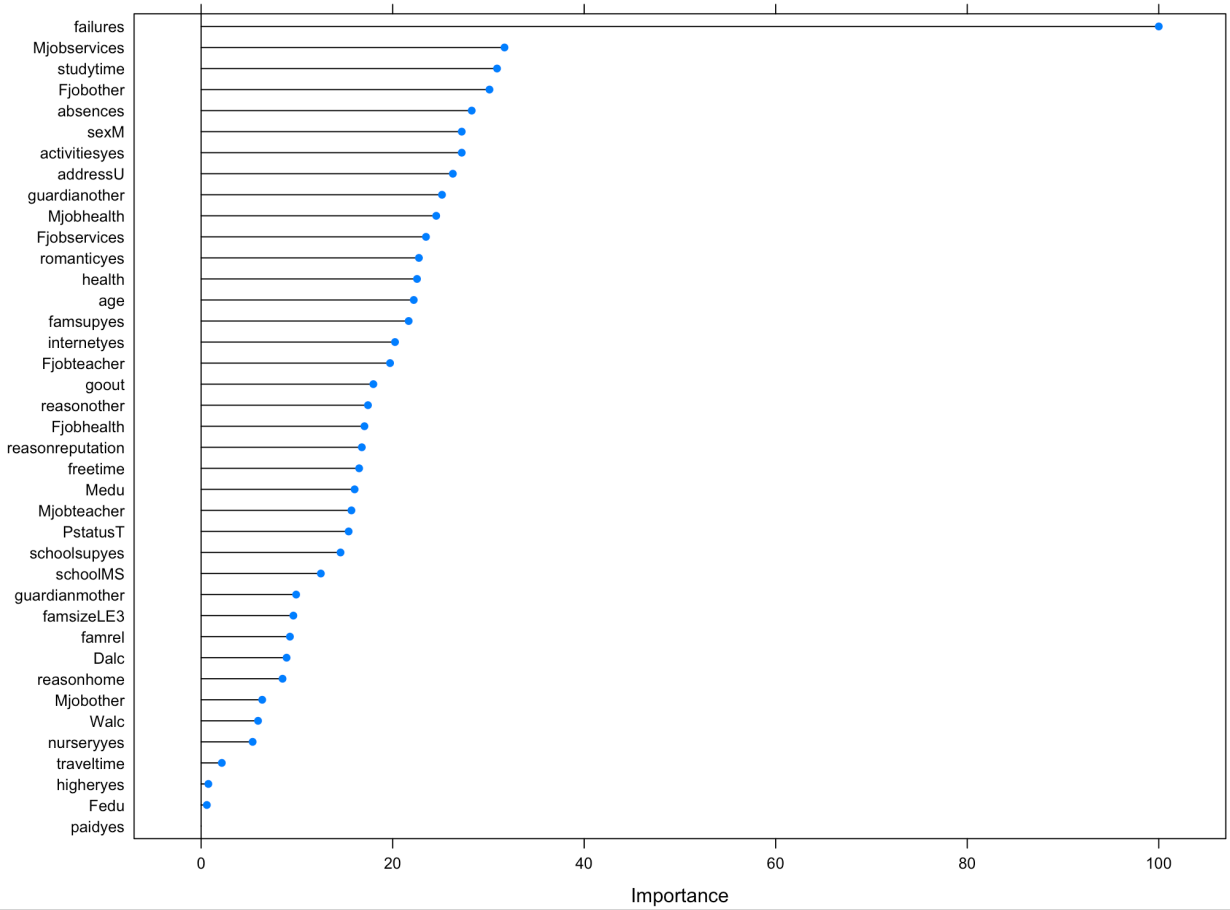
I have taken range from 0 to 29 because there were 30 features and index starts from 0; and the dependent variable (G3) will be added to the dataset as per the data preparation function.

The finding are as follows :

```
_____________________________________________________________
Linear Regression | Running the algo on Train data
Mean squared error = 0.2257
Mean Absolute error = 0.4541
_____________________________________________________________
_____________________________________________________________
Linear Regression | Running the algo on Test data
Mean squared error = 0.2606
Mean Absolute error = 0.4943
_____________________________________________________________
```

- ***Experiment 3 – Parameters that I think are best suited to predict the output***

  I feel below mentioned 10 features will be most helpful to predict the output better. These features were selected based variable importance function in R.




```
data1 = data[['failures', 'Mjob', 'Fjob', 'studytime', 'absences', 'sex', 'activities', 'address', 'guardian','romantic']]
```

Thus, we can see that failures have the maximum effect on the mindset of a child and his attitude towards studies. Similarly, jobs of mother and father effect the child's education which tells us how much time they can give to child. Studytime, absences and activities are also important features as they give insight about the how much effort a child puts in for stud and how serious is he for studies. Address was important as it tells us

if student was living in Urban area or Rural area, as this might affect the opportunities and exposures a student get to interact with. Guardian was important because it might affect the emotional state and support of the child. A child may feel more safe and happy with Mother or Father instead of Other and this can affect the studies and concentration of a child. Similarly, being in a romantic relationship may be distracting and might affect the final grade of a child.

absences - number of school absences
sex - student's sex
address - student's home address
Mjob - mother's job (nominal: "teacher", "health" care related, civil "services"
Fjob - father's job (nominal: "teacher", "health" care related, civil "services"
studytime - weekly study time
activities - extra-curricular activities
romantic - with a romantic relationship
guardian - student's guardian (nominal: "mother", "father" or "other")
failures - number of past class failures

The following are my finding for the linear regression for this model :

```
_________________________________________________________
Linear Regression | Running the algo on Train Data
Mean squared error = 0.2157
Mean Absolute error = 0.4345

_________________________________________________________
_________________________________________________________
Linear Regression | Running the algo on Test Data
Mean squared error = 0.2097
Mean Absolute error = 0.4291

_________________________________________________________
```

Here the MSE for Test Dataset was lower than the Train Dataset, which is good.

- What do you think matters the most for predicting the value and class of grades?

  I understand that it the explanatory power or correlation of the independent variable with dependent variable is important to understand. If there is any pattern between the two using the visualizations and exploiting those features. Also to check if the independent variables are correlated to each other and remove those.

  Every problem in Machine Learning cannot be solved with just one learning algorithm. We need to apply different learning algorithms depending on whether the relationship between the independent variables and the predictor variable is linear or non-linear. As we have seen in our analysis, Linear Regression and Classification did not have plausible accuracy for this particular dataset. There might be certain nonlinear relationships existing between the independent and the target variables in this dataset which might be better captured by learning algorithms like the neural network which are better suited to modeling non-linear relationships. Hence the most important thing that matters is the learning algorithm chosen.

- What other steps you could have taken with regards to modeling to get better results?

  I could have handled this problem would by first trying to reduce the dimensionality of the feature vectors by using Feature selection (forward selection or backward elimination) or PCA, and then use neural networks to model the predictions.