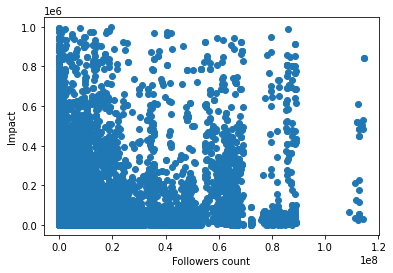
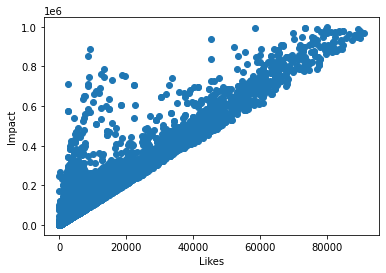
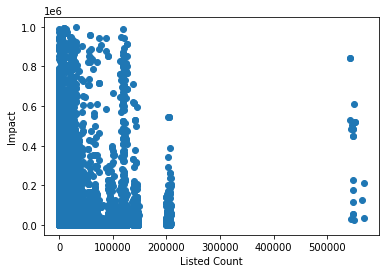
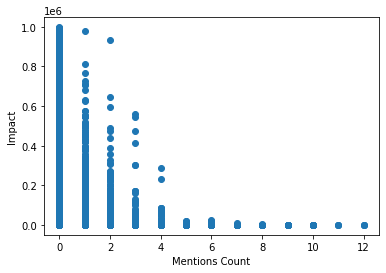
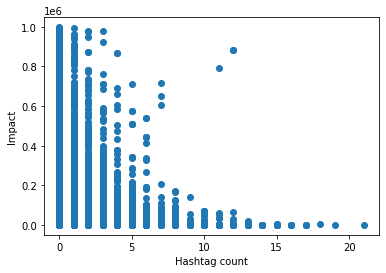
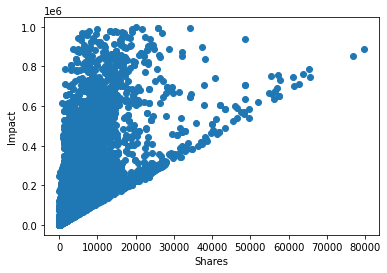
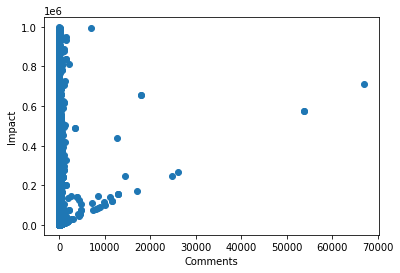
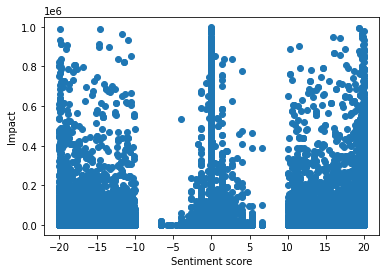
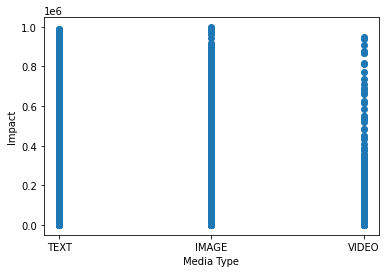
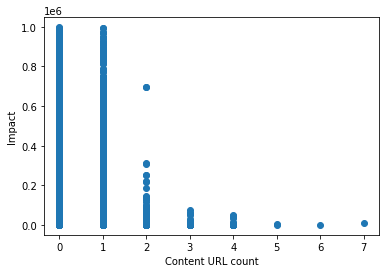
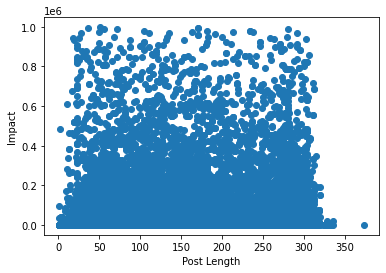
Data Analysis and Results

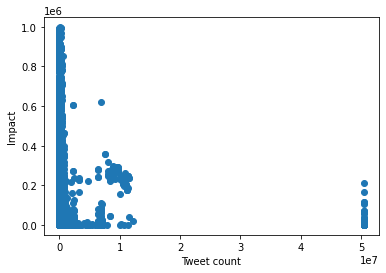
The Impact of a tweet is very crucial in today's world and can be used to analyze what makes a tweet go viral and how this new information can be used for marketing or measuring its effect and predicting what ripple effect a single tweet can generate.

I have used all the 3 learning algorithms and trained the model on it with more than .99 of r2 score which is measured for correlation although it does not suggest anything about future predictive performance*.*

The graphs of data also reveal interesting findings which can also be used for further learning and improving our algorithms.

**Graphs:**



These graphs show the relationship with the Impact on the y axis.

**Linear Regression Results:**

**Code:** <https://www.kaggle.com/firozk/twitter-impact-lin-reg>

I use Standard Scaler to make the values continuous and because the values vary a lot and are not categories.

After training the r2 value is .99 and it is a valuable indicator because more precise regression is one that has a relatively high R squared (close to 1) although it does not suggest anything about future predictive performance.

**Run Time 20.3 seconds**

**Neural Network:**

**Code:** <https://www.kaggle.com/firozk/twitter-impact-keras-nn>

In this I used Standard Scaler to make the values continuous and because the values vary a lot and are not categories.

For this I used a sequential model with 1 hidden layer and relu activation in all the layers as I wanted to make it fast and give only limited value.

loss='mean\_absolute\_error',

optimizer=tf.keras.optimizers.Adam(0.001),

metrics = "mean\_absolute\_error"

These are the parameters I used for training.

Loss and metrics are taken from keras module as they are the preferred ones for the regression model.

For optimizer I used Adam with 0.001 because Adam is more versatile, fast and widely used with .001 for through learning .

This gives me .52 as the loss and mean\_absolute\_error from metrics.

**Run Time 65.3 seconds**

**Decision Tree with Boosting:**

**Code:**<https://www.kaggle.com/firozk/decision-tree>

For this I first created the decision tree using scikit learn, which gives me an r2 value of .998 but after gradientBoostingRegressior the r2 value jumps to .999.

**Run Time : 33 seconds**

**Conclusion:**

**The Linear Regression model works the best as it gives the same highest r2 value in the lowest time of 20 seconds.**