Anomaly Detection of Twitter Tags Using Isolation Trees

Nathen Byford

Contents

1 Introduction

In data analysis one of the most prevalent issues among all areas of data collection is anomaly detection. Anomalies can cause numerous problems in a statistical analysis and effect the results in unwanted ways. Many modern data pipelines rely on data streamed at high rates to make near instantaneous decisions for business needs.

There are many possible causes for anomalies in data. Possible causes are: faulty sensors, poor data quality, external actors, and others. Identifying anomalies is the first step to identifying the cause of the anomaly. Knowing the cause of the anomaly can help determine if the anomalies should be included in the analysis of the data and the decision-making process.

1.1 Data

2 Methods

In this report we will compare the performance of four classes of anomaly detection algorithms. These classes are: linear regression, seasonal decomposition of time series by Loess, neural network, and isolation forest. Some of these models are post hoc and intended to be performed after the data is collected, others can be trained, and then the new data can be input into the model for decision-making. All models will be trained on the first half of the time series if necessary and all test measures are from the second half of the time series.

2.1 Linear regression model

Anomalies are similar to outliers, points that are not expected and further from the other data points. It's possible to use a common method of outlier detection with the leverage calculations for a simple linear regression. For the time series in this study the x variable is date/time and the y variable is the number of twitter tags. Then calculating the leverage measures of cooks distance, covariance ratio, and DF beta if a point has high leverage for any of these measures it's considered an anomaly.

In figure 1 the model is shown, the blue line is the simple linear regression, the green points are identified outliers using this method, and the red x is a true anomaly from the data set.

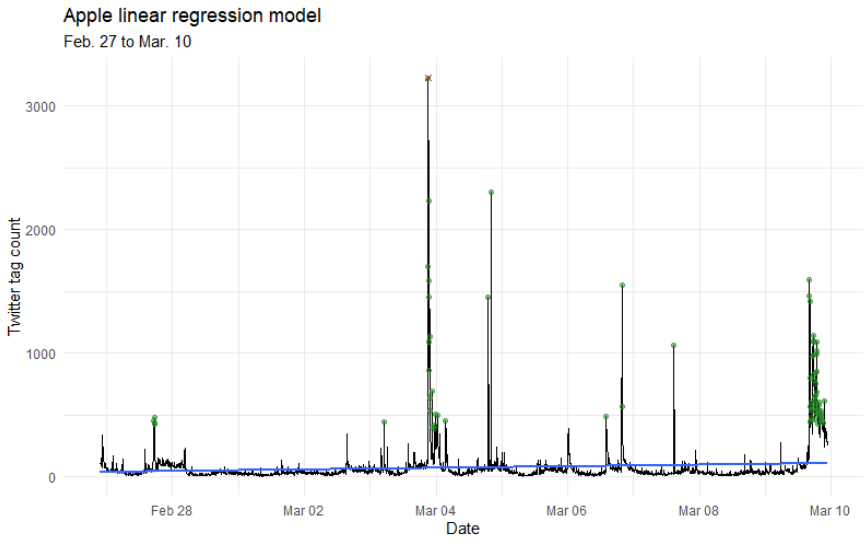

Figure 1: Example linear model

Thinking about this model, the linear regression slope and intercept are not of interest. This is counterintuitive from the typical construction of a linear regression model, here the primary interest is what

data points are leverage points. So there will not be any mention of the coefficients of the model here, only mention of the anomalies identified and the AUC value.

2.2 Seasonal decomposition of time series by Loess (STL)

A common method of anomaly detection with time series data is to utilize the seasonal decomposition of time series by Loess or STL. The STL is as its name suggests a decomposition of the time series by its trend component, seasonal component, and some residual remainder. This can be seen in figure 2. For anomaly detection we can place a bond on the remainder portion of the STL and determine that any remainder outside the these bounds is an anomaly. This would be an indication that the values observed are further than expected from the seasonal plus trend components of the time series.

Similar to the linear regression model, the STL model is post hoc and needs the full data, as the classification comes as a byproduct of the model fit to the complete data. The coefficients of the seasonal and trend portions are not if interested. The package that contains the function in R for STL anomaly detection is `timetk`.

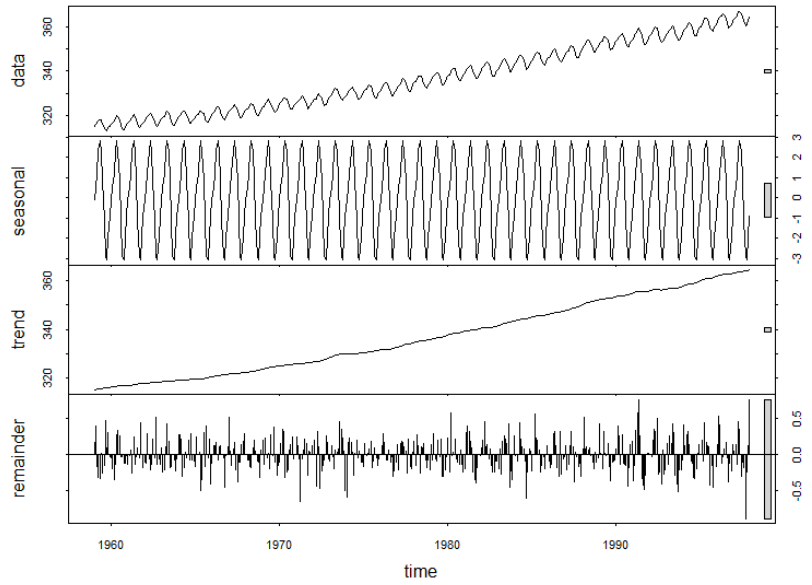

Figure 2: Example plot of STL

2.3 Artificial neural network

Neural networks try to replicate how the human brain thinks when classifying data. Neural networks can find connections that other methods don't see, but they are also more of a black box method where we don't learn much about what is important. Using the package `ANN2` in R for artificial neural networks for anomaly detection we can make a classification network based on stochastic gradient descent using a log loss function.

Using a neural network is good for prediction and classification, so that is why it can be useful for anomaly detection. Here the model will be trained in the `ANN2` package and utilizing a single hidden layer. One thing to note is that the time stamp is needed to be converted to a numeric as the function doesn't know how to use a date time for classification. This could take away the time structure of the data and cause some negative consequences. There are new methods being introduced for time series neural networks, but I was unable to get any to run.

2.4 Isolation forest

The last method is isolation forests, this method is similar to random forest and regression trees with a different goal and measure. Similar to trees we take the data and split it into partitions as seen in figure 3, the data will continue to be partitioned until all points are the only member of their own partition. The data are split randomly by variable and value, this is why an ensemble of these isolation trees is utilized. As each point gets its own partition we count the number of splits necessary to isolate the point, this is the isolation depth of the point for the tree. Using the isolation depth of each point for each tree in the ensemble, we take the average depth of each point and that is the "anomaly score".

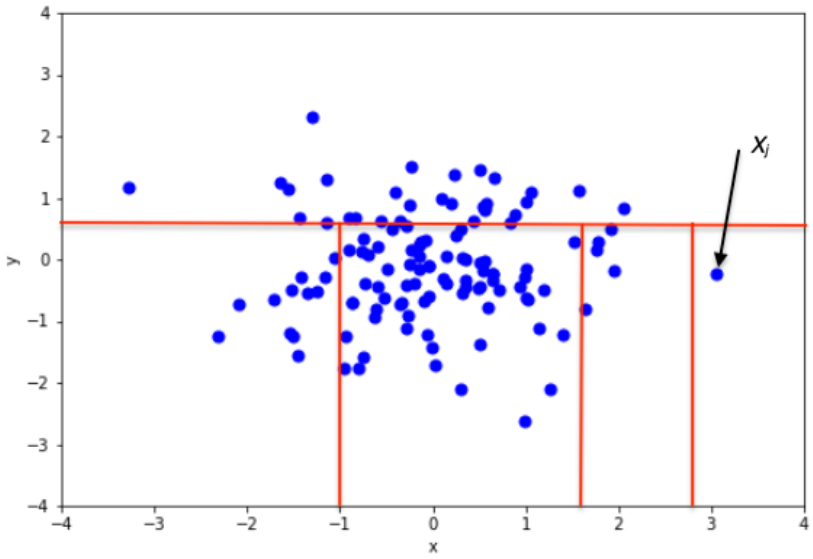

Figure 3: Example of data partition for isolation forest

3 Results

Table 1: AUC results of each model

Model	AAPL	AMZN	CRM	CVS	FB	GOOG	IBM	KO	PFE	UPS
lm	0.9962	0.9888	0.9814	**0.9899**	0.9945	0.4851	**0.9899**	0.9968	0.9790	0.4939
stl	0.9554	**0.9920**	**0.9892**	0.8667	0.9868	0.4839	0.9869	0.9843	**0.9869**	0.4817
SGD	**0.9994**	0.5000	0.5000	0.5000	**0.9999**	0.5000	0.5000	0.5000	0.5000	**0.5000**
IF	0.9967	0.5000	0.5000	0.7497	0.5000	0.4999	0.5000	**0.9991**	0.5000	0.4981
Density IF	0.9665	0.9859	0.9713	0.9611	0.9905	**0.9736**	0.9746	0.9912	0.9758	0.4888

4 Conclusion