Correlating Operational Data with Sensor Diagnostics

Team SICK: Muhammad Kasim Patel, Sarthak Jagetia, Namir Fawaz
{kasimp93@bu.edu, sarthakj@bu.edu, namirfa@bu.edu}

1. Introduction:

In all the IoT hype one thing that is agreed upon by most pundits is that data is the value of the IoT. Hardware devices such as sensors digitize the physical world and generate all this valuable data. The data can be useful in lots of ways and can help determine the state of the system and how it can be improved. Thus, the objective of our project is to work with SICK in order to analyze their sensor data, find any correlation between operational data and sensor diagnostics data, and return meaningful analysis of the data.

2. Problem Formulation:

The problem was considered to be an open ended problem by taking the data and visualizing different parameters of the sensors and object. The goal was to find correlations between different parameters of the sensor and the object data, and also to look for anomalies in the data if any existed. A few examples were given by the project partner. Apart from these, there were other anomalies and relations which we found while analyzing the data and correlating different features of the dataset.

3. Data:

The raw data which was given to us had 42 different XML files of Operational Data and 8 TSV files of Sensor Data. The sensor data files were taken over a large period of time while the operational data was arranged according to the days. Each file had data which was collected during the whole 24 hours and also had timestamps to distinguish them.

There are two broad categories of Data available to us:

1. Operational data: Data collected by the sensors
    - Info Package box
    - Scan time
    - Dimensions
    - Angle of orientation
    - Gap info etc.

2. Diagnostics data: Data about the sensors themselves (sensor health)
    - Time
    - Temperature
    - Voltage
    - Operating Hours
    - Illumination etc.

3.1 Data Retrieval:

All of our data was provided to us in TSV and XML files. The Operational Data was organized into TSV format while the Sensor Diagnostics data was organized in XML format. The TSV files were rather easy to parse into a csv file using the pandas csv parser, because they were

already in the form of tables. On the other hand, parsing xml files took a major portion of our time. This is because the xml files were in the wrong format as they had multiple node elements with every element having it's own namespace.

This made it difficult for the iter-parser to parse the file and it encountered junk after the first node itself. After endless trials we later realized that adding a "fake root" element to every file in the xml format would solve the problem. The xml data was then retrieved into a dictionary and then into a csv file for easy read purposes.

3.2 Data Preprocessing:

Here, we focussed on cleaning the data after we converted the raw data to usable formats. We read the csv files into pandas dataframes for easy modifying/addition/deletion of columns. Pandas dataframe gave us the ability to drop those columns which are not at all useful for further computation. For eg: A column named 'MaxTemperatureDevice' in sensor diagnostics data was constant at 61 for all the entries in data. So we thought it is better to store that as a variable than a whole column in pandas which might increase the Space complexity along with computational intensity.

We dropped similar columns whose standard deviation was 0 and stored them as constant variables in our program. Furthermore data about sensors like sensor name and device id were not very useful in terms of correlation. The biggest challenge which we faced while combining the data was that the timestamps were not synced, and so in order to merge the sensor data and the operational data we had to go through the data and match the timestamps to make sure we were using the data during the same duration from the two types of data we were correlating.

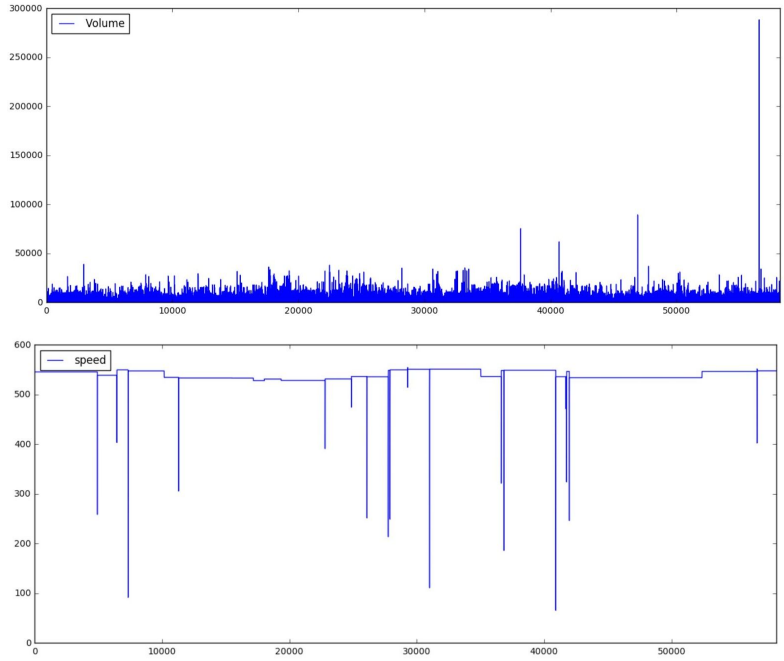
We normalized the data using the sklearn normalizer feature before using the data in any of our algorithm. We also used SVD before using the features in our algorithms. We had to split the timestamp into time and dates and convert them into the datetime format of python so that they could be useful in the time-series analysis and other analysis tools.

4. Data Analysis/Methodologies:

Initially we used different plots to analyze the data and using the data which had standard deviation greater than zero. Next, we tried to use the `.corr` method between pandas dataframe. We found that the correlation between column 'otl' is much higher than the 'volume' (calculated from ole*ohe*owi) equal to 0.65 which was much higher than other correlation values. In such manner we tried to find any correlation between data columns and then later decided to support that with the models discussed below.

4.1 Time-Series Analysis:

For Time-Series analysis we analyzed the different parts of the data and tried to find any correlation between them. One of them which we found worth highlighting was the relation between the speed and the volume of the boxes which were passing through the conveyor belt.



In the above plots we can see a clear relation between the volume of the boxes and the speed of the belt.

4.2 Linear Regression Models:

We developed 5 different models of Linear Regression with which to correlate the data. We decided it would be interesting to search for a correlation between the speed of the conveyor belt and the dimensions of the packages. The dimensions were kept in columns titled ole, owi, and ohe, which stand for object length, object width, and object height. We used the csv files to load data into Pandas Dataframes, and dropped all of the columns aside from the speed, ole, owi, and ohe. We then split this data into two separate data frames, one containing the speed and the other containing the rest of the data.

The first model we created attempted to find a linear correlation between the speed of the conveyor belt and the three dimensions of the package. Initially, we split the data with a 70:30 ratio in order to use data for testing and data for training. We used statsmodels.api in order to create the model, and loaded in the speed data as the “y” value. We loaded in the three dimensions of the package as the “x” value, and created the model. We will discuss the model’s results in *Section 8. Experimental Results.*

The second model we created attempted to find a linear correlation between the speed of the conveyor belt and the volume of the packages. The data was loaded into the Pandas Dataframes in a similar fashion, but this time we multiplied the ole, owi, and ohe values together in order to create a new volume column. Again, we used the statsmodel.api in order to build our model. We loaded the speed into the “y” value, and loaded the the volume into the “x” value. We will discuss the model’s results in *Section 8. Experimental Results.*

Finally the last three models attempted to find a linear correlation between the speed of the conveyor belt and any one of the package's dimensions. We loaded in the data very similarly to the previous models, with the 'y' data corresponding to the speed and and the 'x' data corresponding to one of the dimensions. Again, we will discuss the model's results in *Section 8. Experimental Results.*

4.3 K-Nearest Neighbours:

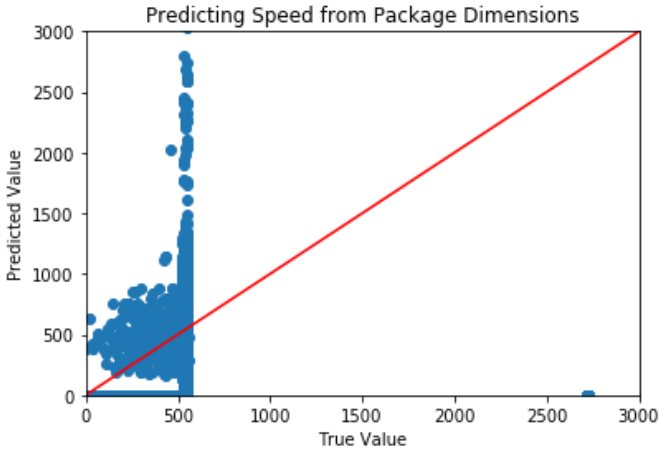
The aim to implement a K-nearest Neighbours algorithm was to find a relation between any variable say volume with the whole operational dataset. This was just to see how the volume relates with respect to all the variables in the data set. The accuracy for the algorithm for k=2 was 0.072

After which we tried to work with specific variables that can answer our questions about volume and speed specifically. We chose volume specifically because we believe that the size of the package might have an impact on the conveyor speed. I.e. if there are suddenly a lot of packages on the conveyor belt which are huge, the speed should show some changes in its value.

5. Experimental Results:

Linear Regression:

The first model we created attempted to find a linear correlation between the speed of the conveyor belt and the dimensions of the package. We found that this model had an R-squared value of 0.458, which was not very high. We used the last 30% of the data to test the model by allowing it to guess the speed based on the dimensions of the data, and plotted the predicted value against the true value, which is shown below:

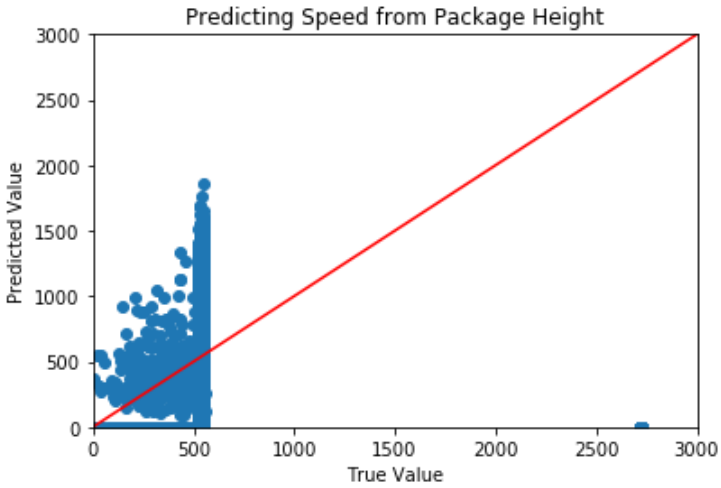


Clearly, the model did not do a very great job of predicting the speeds. In an ideal model, all of the points would lie on the red line because the True Values and Predicted Values would all be identical. This was to be expected because of our sub-par R-squared value, and because a the

Linear Model was bound to predict speeds that would be much higher than 550 ft/min even though the True Values rarely exceeded this number.

The second model we created attempted to find a correlation between the speed of the conveyor belt and the volume of the package. This model had an R-squared value of 0.178, which was very low. When we used the last 30% of the data to test the model, we found a plot very similar to the plot above, which is not worth showing.

The last few models had interesting results. The first model, which correlated the length with the speed, had an R-squared value of 0.405. The second model, which correlated the width with the speed, had an R-squared value of 0.448. The third model, which correlated the height with the speed, had an R-squared value of 0.414. When using the last 30% of the data to predict the speeds, all of the plots again looked very similar to the plot above. The most accurate looking plot was the third model's plot, because it capped the highest speed values much lower than the rest of the models did:



Even though this model had the lowest R-squared value, it seemed to guess the value of the speed fairly well, with less drastically high values than the other models.

6. Conclusions:

In the end, we were able to implement many of the tools we learned over the course of the semester in a real life project. Still, we did not find any interesting trends in the data. We attempted to use Linear Regression, K-Nearest Neighbors, and Time Series Analysis and found some distinct relations between data points such as package dimensions and conveyor belt speed using only time series analysis. In the case of our project, definitively finding no trends was equally as important as finding trends. By finding some trends between the data points, we are able to say with confidence that given our set of sample data, the dimensions of packages (volume) should in some ways affect the speed of the conveyor belt, but not the temperature.