CS 5304 Homework 2 Report

Conor Cunningham
Rory Connolly

March 8, 2019

1 Evaluating and Processing the Data

After getting a general sense of the data, we worked towards dropping fields which we felt were insignificant. First, we made note of any fields that had only one or two values (weather station ID, latitude, longitude, etc.) and removed them. We were left with one column identifying the date, another identifying the weather station, and the remaining containing integers and floats representing the different weather values. Next, we worked towards understanding the distribution of data and identifying any linear correlations. In agreement with the homework outline, we replaced any NaN values found in the solar radiation column with 0's. Then we calculated the Pearson correlation between precipitation values and the other fields (see Fig. 1). With this initial evaluation, we found a slight correlation between gust, dew, and temperature. After identifying the slight correlation, we split the data into two dataframes: one containing values for *Sao Goncalo* and the other *Vitoria*. Fig. 2, illustrates the precipitation each month at each weather station. In viewing the missing values for each data-set, we found that the *Sao Goncalo* was missing less data in the percipitation feature than *Vitoria*. As a result, we opted to train and test on *Sao Goncalo* for the rest of the report.

The next step was addressing outliers. To visualize the data, we plotted a box-plot for the *Sao Goncalo* data-set as shown in Fig. 3. The range in *smin* and *smax* values were alarming, so we plotted both fields together as shown in Fig. 4. To further visualize the fields, we plotted the standard distribution of the pressure data in Fig. 5. We opted to remove the outliers by thresholding the Z-score at 3 and replacing outliers with the mean value. The *gbrd* field also looked to have outliers, and we chose to threshold it as well.

Fig. 6 shows a box-plot of the *Sao Goncalo* data-set after thresholding, which now looks more acceptable. Similarly, Fig. 7 shows the improved distribution of the pressure fields post-threshold. Fig. 8 shows that there are still outliers between the *smin* and *smax* values. Since these results were outliers possibly to due faulty monitoring systems, we opted to drop the *stp*, *smin*, and *smax* columns from the data-set. After dropping the fields, the linear correlation between fields in the *Sao Goncalo* data-set became more pronounced, as shown in Fig. 9.

Next, we began to build our features for training our models. Using the correlations shown in Fig. 9, we chose to drop the *temp*, *wdsp*, *wdct*, and *gbrd* columns from our data-set in an effort to limit the number of features and discard uncorrelated data. After dropping the fields, the remain data-set was normalized. With the features defined, we began processing the data. We wanted to process the data such that each entry of our training data covered a 3 hour window, and the label for each entry would be the amount of rainfall predicted to occur in the next hour. To process this, we iterated over the *Sao Goncalo* data-set with a window of size 4. For each iteration, we took the weighted average of each feature from the 3 oldest entries in the window, where the features of the most recent entries were weighted higher than the features of the older entries. Next, we looked at the most recent entry in the window. If this entry had a value greater than 0 for precipitation, then we labelled the weighted average with that value. In this way, each entry represented a 3 hour period and the label served as a prediction for precipitation (yes/no/amount) in the next hour.

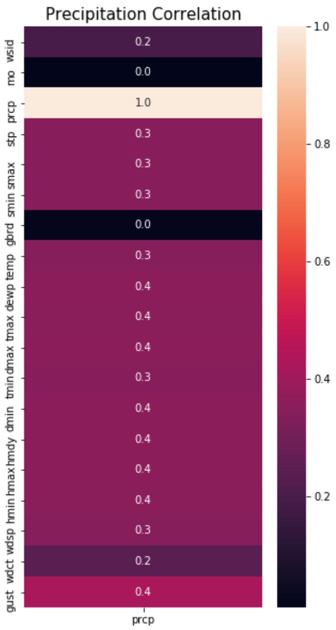


Figure 1: Correlation between 'prcp' and other fields in uncleaned data.

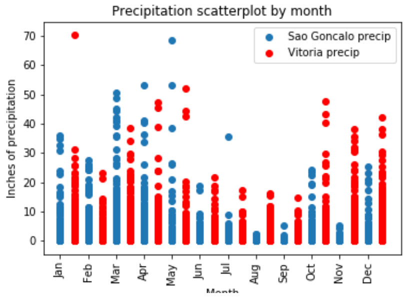


Figure 2: Distribution of rainfall per month per station.

For our linear regression model, we could use this data as is. For the logistic regression model, if the entry label determined there was precipitation in the next hour, the label was converted to a 1 for classification.

2 The Model Pipeline

With our training data defined, we defined a processing pipeline and began fitting and training our models. We first fit a logistic regression model with the training data. Given a feature vector that represented a 3 hour period (see the explanation above), the model would output a 1 if it predicted it would rain in the next hour, otherwise it would output 0. When cross-validating this model, we consistently obtained an accuracy of about **92%** on the *Sao Goncalo* data-set. To further validate, we split the data-set into training and testing

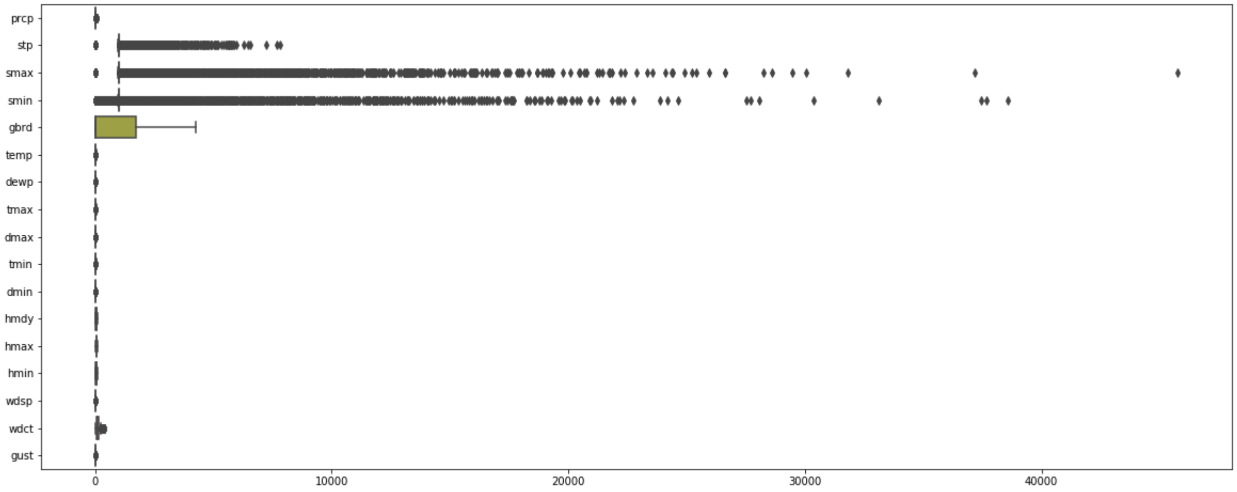


Figure 3: Box-plot of the Sao Goncalo data-set.

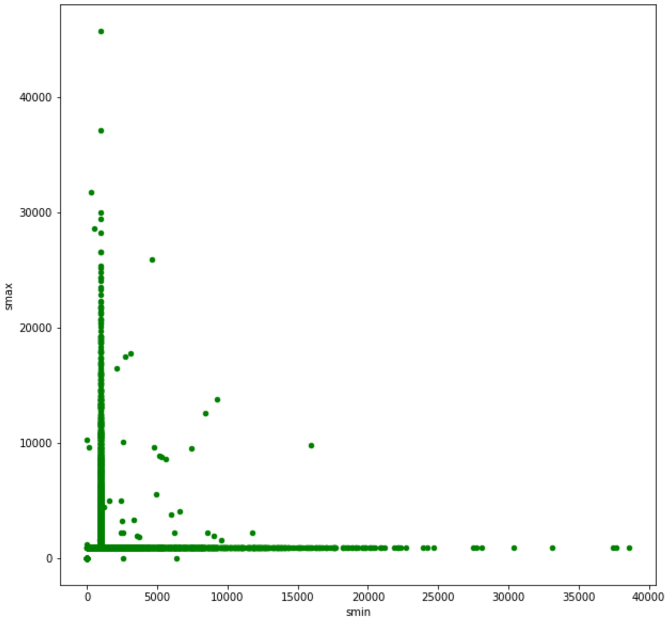


Figure 4: Correlation between smin, smax in Sao Goncalo data-set.

and trained the model again. The model performed with an accuracy of **92.47%** and had a precision-recall score of **0.449**. The ROC curve for the model is shown in Fig. 10. The AUC is **0.55**. Although the accuracy

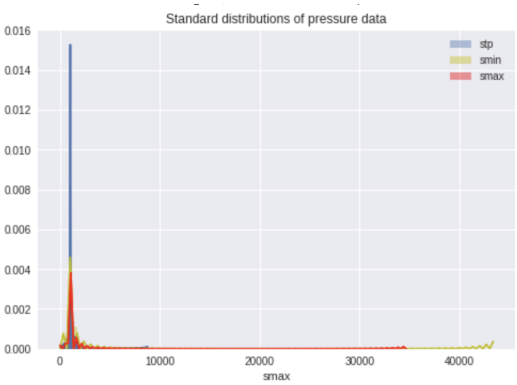


Figure 5: Distribution of smin, smax in Sao Goncalo data-set.

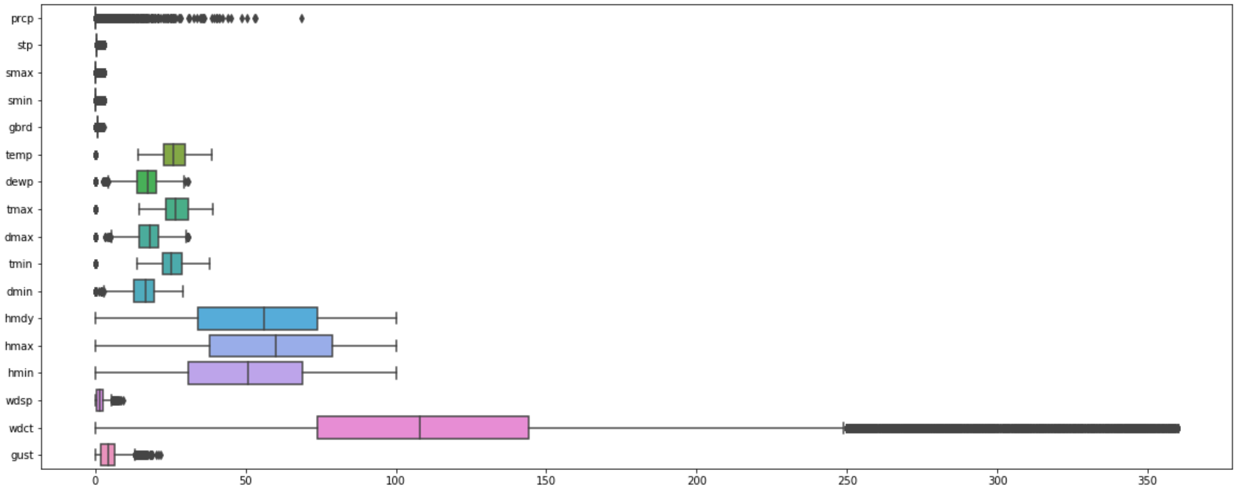
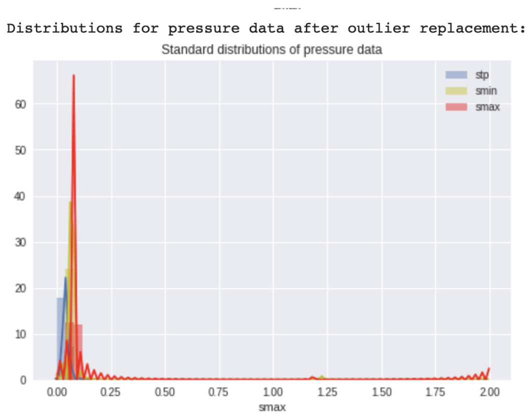


Figure 6: Box-plot of the Sao Goncalo data-set after thresholding.

is high, the model does only slightly better than a random predictor.

Next we trained a linear regression model on the *Sao Goncalo* training data. After testing (and without help from the logistic regression model), it reported an R-squared value of **0.19**, a Mean Absolute Error of **0.2459** (how big an error we can expect from the forecast on average), a Mean Squared Error of **1.126** (a measure of the quality of the model), and a Root Mean Square Error of **1.061** (a measure of how spread out residuals are). Fig. 10 shows the relation between the linear regression's predictions and the true values. Although there does seem to be some matching between the predictions and the truths for smaller values, overall the model is not terribly effective by itself.

With the above results, we designed the prediction pipeline. First, both the logistic and linear regression models are trained on a set of training data. The logistic regression model takes a data sample that represents 3 hours of weather data and predicts if there will be rain in the next hour. Similarly, the linear regression model takes a data sample that represents 3 hours of weather data and predicts the amount of rainfall in



Distributions for pressure data after outlier replacement:

Figure 7: Distribution of smin, smax in Sao Goncalo data-set after thresholding.

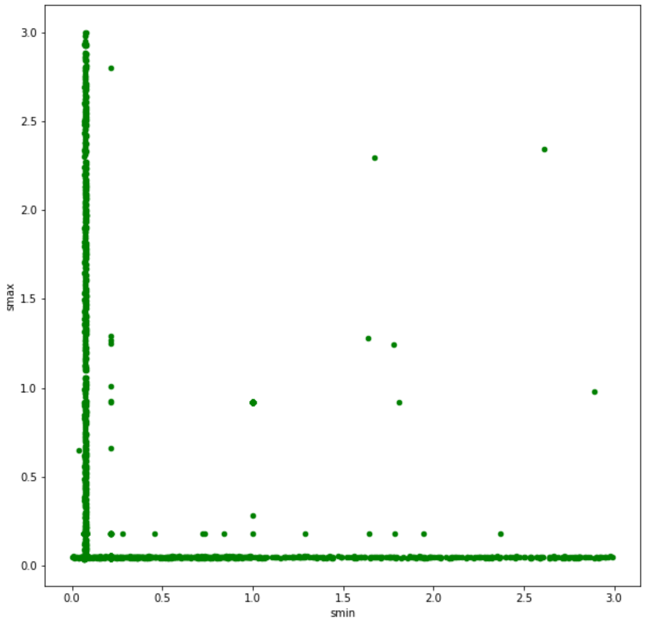


Figure 8: Correlation between smin, smax in Sao Goncalo data-set after thresholding.

the next hour. In this way, both models are tested on training data and output a list of predictions. We iterate through the linear model's prediction list. At each point, we check if the logistic model predicted if there should be rain in the next hour. If there should be rain in the next hour, the linear model's prediction is left alone. Alternatively, if the logistic model predicted there should **not** be rain in the next hour, the linear model's prediction is set to 0. In this way, to overcome the disparity in the gap between rain and no rain, the two models work to complement one another.

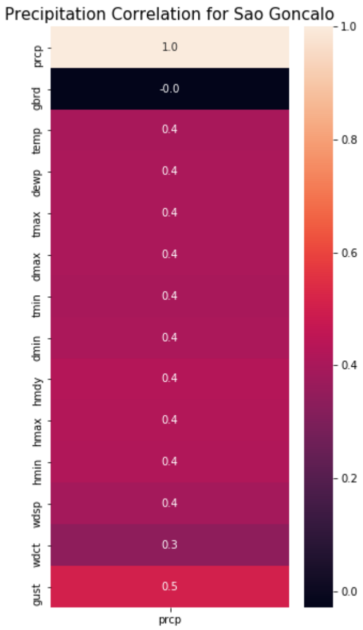


Figure 9: Correlation between 'prcp' and other fields in cleaned Sao Goncalo data-set.

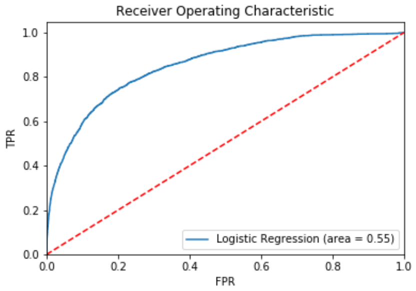


Figure 10: ROC curve for the logistic regression model.

3 Results

On the pipeline, we tested the clean *Sao Goncalo* data, the unclean *Sao Goncalo* data, and the unclean *Vitoria* data.

For the clean *Sao Goncalo* data, the pipeline reported an R-squared value of **0.153** (an improvement over the linear model), a Mean Absolute Error of **0.13** (a 54% improvement), a Mean Squared Error of **1.118** (a higher error than the linear model alone), and a Root Mean Square Error of **1.09**. Fig. 12, Fig. 13, and Fig. 14 show the relation between the pipeline's predictions and the true values. The table below shows the values over all the data-sets.

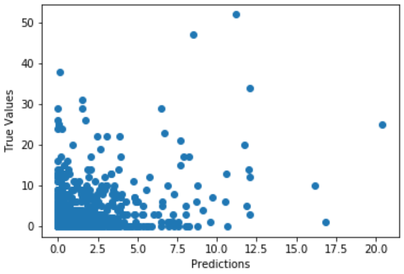


Figure 11: Predictions vs. truths for the linear regression model.

Station	R-squared	MAE	MSE	RMSE
Sao Goncalo (Clean)	0.153	0.13	1.118	1.09
Sao Goncalo (Dirty)	0.058	0.07	1.01	1.005
Vitoria (Dirty)	0.215	0.13	1.10	1.04

Table 1: Pipeline results

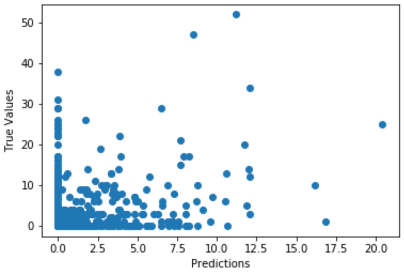


Figure 12: Predictions vs. truths for the model pipeline on clean San Goncalo data.

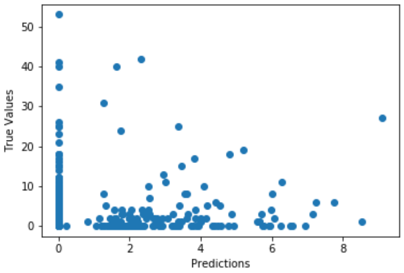


Figure 13: Predictions vs. truths for the model pipeline on dirty San Goncalo data.

4 Conclusion

Since we are dealing with weather, and the effects of weather are continuous over time intervals, we believe there is a provable pattern with which one can make predictions. Yet, there was a lack of strong linear correlations in the given data, combined with missing values and erroneous air pressure fields. As non-

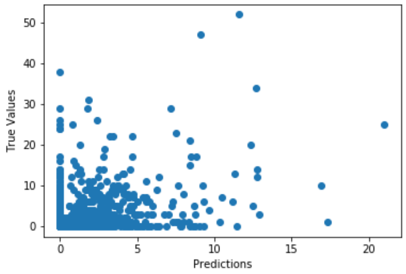


Figure 14: Predictions vs. truths for the model pipeline on dirty Vitoria data.

experts in the weather data field, we felt that dealing with known erroneous data was too difficult to parse and introduced unwanted noise. To make predictions with higher accuracy, we found it necessary clean up the input data. In cleaning the data, thresholding to remove outliers, and dropping columns to limit the feature space, the linear correlations between different data fields increased. As correlations became more pronounced, we leveraged the relationship between the fields to better develop more effective models.

Our final results show a slight improvement on the linear regression model when using our pipeline, but the predictions are still not as accurate as we would like. Similarly, the quality of predictions does not change much depending on the data-set. Going forward, we may need to rethink our data pre-processing. In terms of predictions, we believe some of the errors are due to the fact that it could be raining in the following hour, but it might not be raining exactly when the weather station measures the weather conditions. Similarly, there is the possibility that rain is in the process of stopping early on in the next hour and the accumulation is almost 0. By weighting the 3 hour time interval based on how recent the data values are, we hoped to overcome some of these issues. In tackling this problem again, we would try a Ridge Regression model over the linear regression model to take advantage of the multi-collinearity of weather patterns.