How Much Did It Rain? II

Executive Summary

Problem description

In meteorology, Quantitative Precipitation Estimations (QPEs) is a technique to approximate the amount of precipitation that has fallen at a location or across a region. Currently, there are two types of measurements: radar and rain gauges. In this project, efforts have been devoted to fit the actual amount of rain reported at rain gauge given polarmetric radar measurements at a given location and time.

Major concerns and assumption

Rainfall is hard to predict because of the intrinsic uncertainties across time and space. Even for the meteorologist, the variance of the rainfall they predict can be up to 50%. This fact makes it a very intractable task for us to accurately estimate rainfall by using whatever data mining models.

The relationship between the predictors (polarimetric radar observations) and the response (rain gauge) is interfered by different factors and thus very difficult to determine. For example, radar observation is often contaminated by biological echoes whereas the rain gages are affected by siting, wind, splashing, etc.

The precipitation is itself a very complex process. During the raining process, different contents of precipitation (rain, snow, ice and mixes) may exist at different cloud layers which result in different radar output. However, when it comes to the ground surface, the rain gages all measure the same after one hour accumulation. This gap also hampers the development of data-based model for the proposed problem.

Rainfall in reality has extreme value. For example, the maximum rainfall ever recorded in the world is 330mm per hour. Considering the effects of wind or hail, it is assumed that rainfall values greater than 1,000mm are outliers and are removed in both training and test data (the maximum one is 33,000).

Summary of findings

Several data mining models are performed for this problem: multiple linear regression, neural network stochastic gradient boosting, eXtreme gradient boosting. The mean absolute error (MAE) is used for evaluating the performances of these models. The major findings are as follows:

1. All data mining models achieve very low R^2 value. This is explained by the huge uncertainties concealed in the problem. For MAE, the differences among all the models are trivial and it turns out that the neural network outperforms the others. It is noteworthy that the largest R^2 does not necessarily result in the lowest MAE.

2. In tree-based modeling, it can be found that the horizontal and vertical reflectivity output by polarized radar are most important variables in the model. This is consistent with that of the meteorology.

3. For all the models, it turns out that the models perform better for smaller rainfall than larger rainfall. For extreme rainfall, those data are perceived as outliers and may not be explained by the model we develop.

Recommendation

For the model we developed, it is recommended to be applied only to small and moderate rainfall. For large and extremely large rainfall, the model needs more consideration and extra information for improvement.

Due to the complexity of the problem itself, the recommendation for improving our model is to develop more feature engineering and explore possible models that can consider uncertainties. More information is needed to improve accuracy of the model. The radar observation is sometimes not reliable. The results may be improved if some filtering mechanism are applied to radar observations.

1 Project understanding

Rainfall is hard to measure and predict due to quantities of uncertainties across time and space. In meteorology, there are two types of measurements: radar and rain gauges. However, these two types of measurements never matched with each other because of the different sources of error. A radar is a spatial average of hydrometeors aloft (which is often contaminated by biological echoes) whereas a rain gauge is a point measurement of rain that fell into a small tube (are usually affected by siting, wind, splashing). Currently, radar observations which have more widespread coverage are "corrected" using nearby gauges and a single estimate of rainfall is provided to users who need to know how much it rained. This project tries to fit the actual amount of rain reported by the rain gauge given polarimetric radar measurements at a location and time [1]. The research of rain measurement prediction problem will help provide more accurate information to those who need to know how much it rained on a particular field, such as agricultural people. Moreover, it also boosts the further study of radar-based Quantitative Precipitation Estimates (QPEs).

The overall objective of this project is using the snapshots of polarimetric radar, see **Figure 1**, values to predict the accumulated hourly rain measured at gauges. The dual polarized radar is a technology recently developed by the U.S. National Weather Service in an effort to improve the accuracy of precipitation estimates. Different from the conventional Doppler radars, the polarimetric radars are able to provide higher quality data because they transmit radio wave pulses with both horizontal and vertical orientations. Dual pulses make it easier to infer the size and type of precipitation because rain drops become flatter as they increase in size, whereas ice crystals tend to be elongated vertically. In meteorology, the polarized radar observations are first used to clarify different types of precipitation and different rainfall rate relationships (as known as Z-R relationship) are developed separately for each type. Finally, the one hour accumulated rain is computed by combination of all rainfalls at different layers [2].

In this project, we attempt to use different data mining models to observe the relationship between accumulated hourly rain gage and the parameters of dual polarized radar. There are several challenges: (1) a great deal of uncertainties exist both spatially and temporally; (2) the parameters of dual polarized radar collected at a specific location and time are not directly linked with one hour accumulated rain gage but actually linked with rainfall rate; (3) Different types of errors attribute to the reliability of radar output and rain gage records. For example, radar is most affected by the biological echoes and obstacles while the rain gages are affected by wind, splashing, etc. Since the mechanism of precipitation is itself very complex, we cannot expect a clear relationship between predictors and response. Based on this fact, several data mining models are performed to find out the optimal model possible to solve the proposed problem: multiple linear regression, neural network, tree-based models.

Mean absolute error (MAE) is used as the metric for assessing the performance of data mining models in this project. The reason to use MAE is that some extreme values exist in the data (such as extraordinarily heavy rain) and using MAE is most useful when large errors are not of great interest or particularly undesirable [3]. It is noteworthy that a minor improvement on the MAE counts a lot for the final outcome since the problem itself contains a lot of uncertainties and the rainfall is hard to predict even for the meterologists.

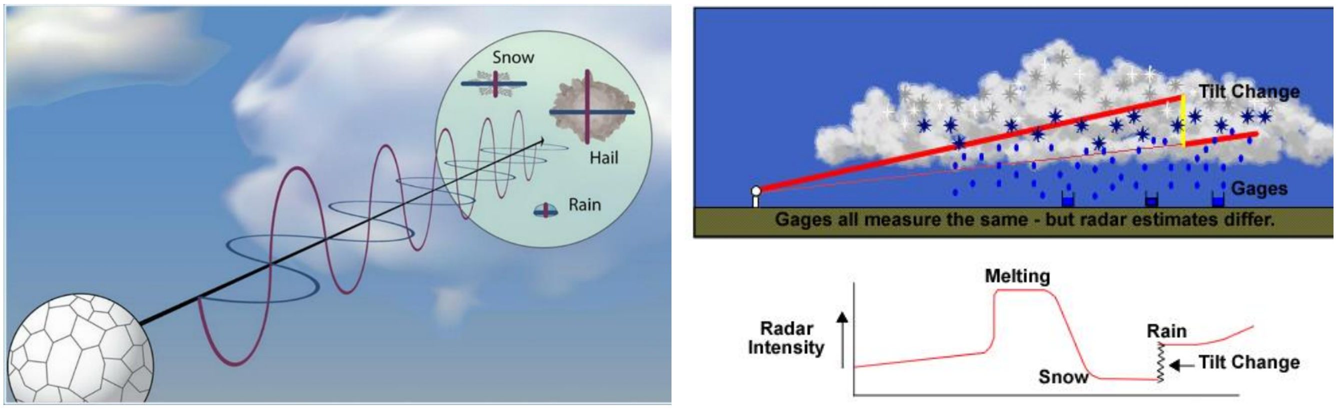

Figure 1: Dual polarized radar for estimating rainfall rate and accumulated hourly rain.

2 Data Understanding

2.1 Data description

In this project, we use the training data provided by *Kaggle* as our whole data set. The dataset consists of NEXRAD data (US National Weather Service's weather radar network, which provides the polarimetric radar data) and MADIS data (a US government service, which provides the rain gauge data) collected on 20 days between April and August 2014.

The dataset consists of 13,765,201 observations of 24 variables. These variables can be divided into three parts: general information, radar output and expected hourly rainfall reported by rain gauge. A specific description of the variables is listed in **Table 1**.

Table 1: Data set attributes.

factor	attribute	Descriptions
General Information	id	unique id of hour at a gauge
	minutes_past	minutes past the top of the hour that the radar observations were carried out
	Radardist_km	Distance of gauge from the radar
Observation from Radar	Ref	Radar reflectivity in km
	Ref_5x5_10th	10th percentile of reflectivity values in 5x5 neighborhood around the gauge
	Ref_5x5_50th	50th percentile of reflectivity values in 5x5 neighborhood around the gauge
	Ref_5x5_90th	90th percentile of reflectivity values in 5x5 neighborhood around the gauge
	RefComposite	Maximum reflectivity in the vertical column above gauge In dBZ.
	RefComposite_5x5_10th	10th percentile of RefComposite values
	RefComposite_5x5_50th	50th percentile of RefComposite values
	RefComposite_5x5_90th	90th percentile of RefComposite values
	RhoHV	Correlation coefficient
	RhoHV_5x5_10th	10th percentile of RhoHV values
	RhoHV_5x5_50th	50th percentile of RhoHV values
	RhoHV_5x5_90th	90th percentile of RhoHV values
	Zdr	Differential reflectivity
	Zdr_5x5_10th	10th percentile of Zdr values
	Zdr_5x5_50th	50th percentile of Zdr values

	Zdr_5x5_90th	90th percentile of Zdr values
	Kdp	Specific differential phase
	Kdp_5x5_10th	10th percentile of Kdp values
	Kdp_5x5_50th	50th percentile of Kdp values
	Kdp_5x5_90th	90th percentile of Kdp values
Response	Expected	Actual gauge observation in mm at the end of the hour

2.2 Data Visualization

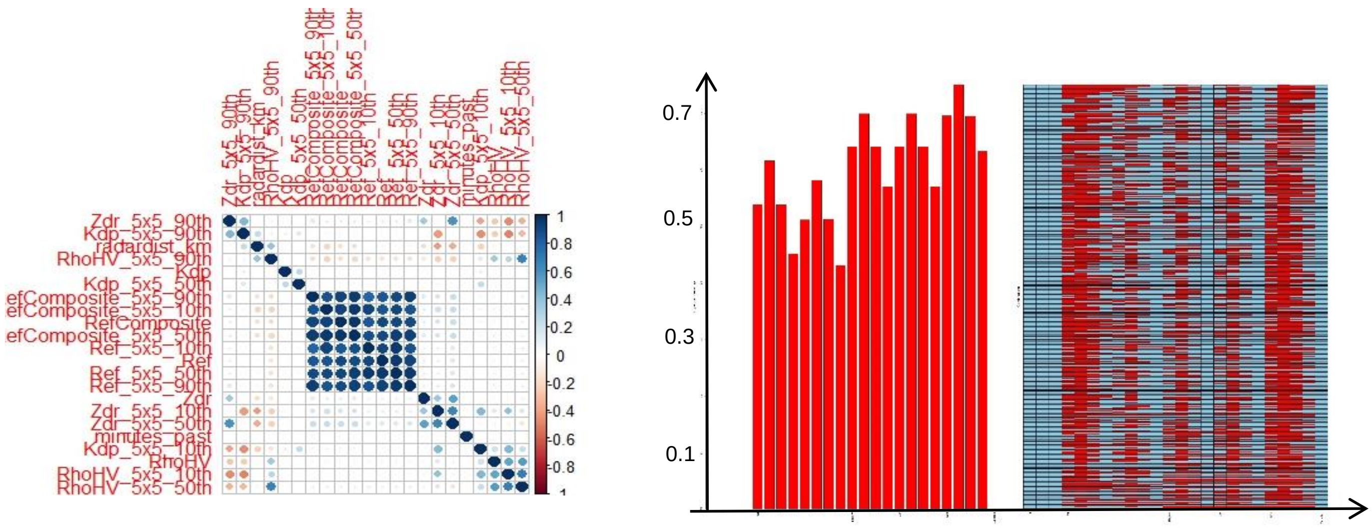

Figure 2: Left side: Correlation of Variables; Right side: Missing values.

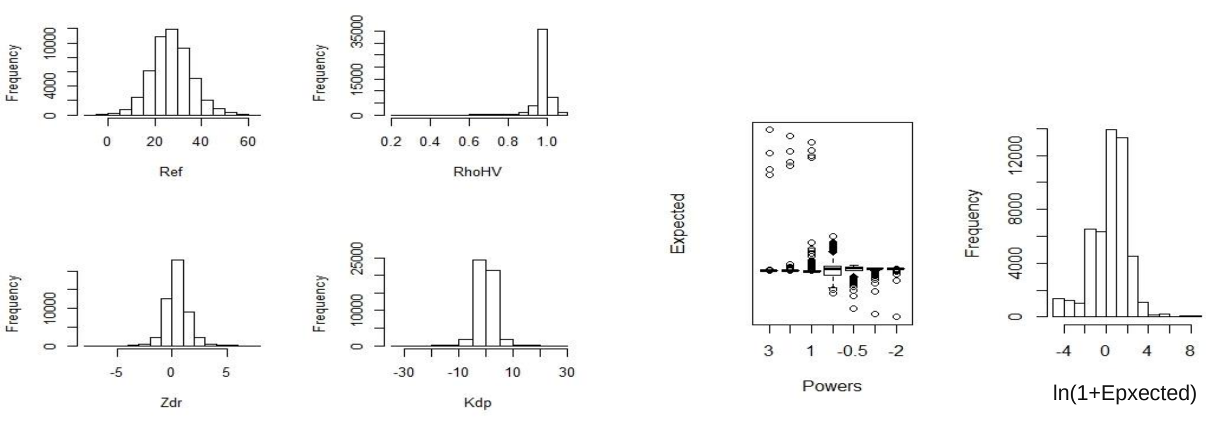

Figure 3: Left side: distribution of key variables (Ref, rhoHV, Zdr, Kdp); Right side: transformation of response.

Among all the 24 variables, there are five key variables outputted by the dual polarized radar: Ref and RefComposite, RhoHV, Zdr, Kdp. Each consists of three additional observation variables: 10th, 50th, 90th percentile values. From **Figure 2** Left, we found that these additional variables are highly correlated with their related key variables. Moreover, the reflectivity from horizontal and vertical radar waves (i.e., Ref and RefComposite) are also highly correlated.

The missing values for each variable are displayed in **Figure 2** Right. Overall, more than 50% of the data is missing.

For distributions of variables, it is found that most of the predictors follow or approximately follow a normal distribution while the response is highly skewed. The distributions of four key radar output variables are displayed in **Figure 3** Left.

3. Data Preparation

3.1 Pre-processing

(1) Missing values
The total number of samples in the data set (13,765,201) is extraordinarily larger than the number of predictors (24). Moreover, in most cases for those samples containing missing data, the number of missing variables is more than 15 out of 24. This indicates that those samples can be non-informative and useless since most observations within that sample are missing. Based on this fact, the simple method of listwise deletion is used for missing values in this data set. After listwise deletion, the sample size condensed to 2,769,088.

(2) Outliers
The maximum value of response (Expected rainfall) in the training set is more than 33000mm, which is obviously unrealistic for one hour rain gage. The mean values of expected value is 108.62. Based on the perspective of meteorologist, the accumulated one hour rainfall greater than 1000 millimeter is impossible [4], so we only use the cases within1000mm. This results in a sample size of 2,763,588.

(3) Transformation
As is described above, the distribution of response is highly skewed. After investigation, we finally use natural logarithm function ln(1+x) to transform the response variable (see **Figure 3** Right).

3.2 Feature Engineering

(1) Data aggregation based on time
As can be seen in **Figure 4** Left, for each ID the radar has many snapshots (observations at specific time) during one hour, but gauges only record the cumulative rainfall at the end of one hour for that ID. So there is a need to aggregate the radar snapshots to be consistent with gauges. For this purpose we do simple and weighted time averaging for each observation with same ID as follows:

$$Value_{ID} = \frac{1}{n}\sum_{i=1}^{n} Value_{ID,i}$$

$$Value_{ID} = \frac{1}{t}\sum_{i=1}^{n} Duration_{ID,i} * Value_{ID,i}$$

(2) Modeling based on distance
As shown in **Figure 4** Right, in the cloud the temperature decreases from lower layers to the higher layers. Thus generally there is snow in the higher layers and rain in the lower layers. The radar records observations at higher layers as the distance between radar and rain gauge increases. Thus radar is more

likely to capture rain for closer gauges and capture snow for gauges with longer distance. Based on this, we categorize our models manually into 16 sub-models based on the distance ranging from 0 to 15 Km.

(3) Marshall-Palmer Parameter

From an empirical study of the distribution of raindrop sizes, J.S. Marshall and W.M. Palmer concluded that such distributions could be adequately represented as negative exponential functions. They gave empirical formulas for the parameter of the distributions as a function of the rainfall intensity [5]. Often when Marshall and Palmer's results are presented the parameter is considered to be determined by the rainfall intensity. The causality however runs the other direction. The intensity of rainfall is a function of the negative exponential parameter and reasonable estimates of the relationship can be derived a priori that approximate the empirical relationship found by Marshall and Palmer.

Here we use the Marshall-Palmer function [5] to extract the MP values from Reflectivity values in the dataset.

$$MP = \left(\frac{(10^{ref/10})}{200}\right)^{0.625}$$

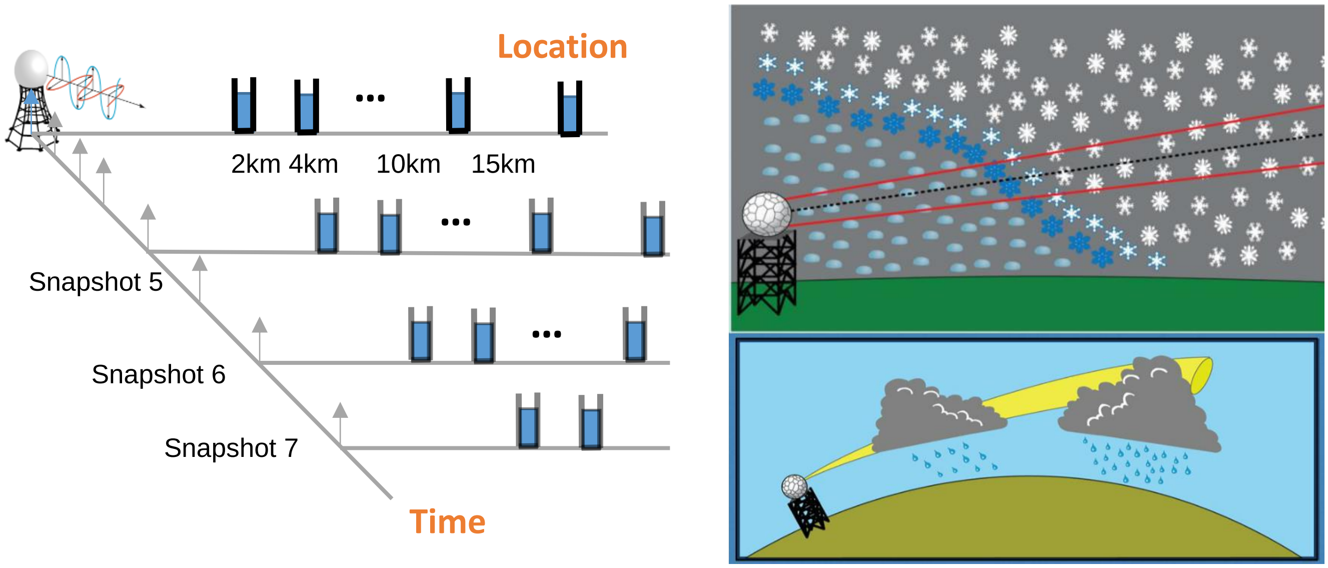

Figure 4: Left side: Data aggregation based on time; Right side: Modeling based on distance.

3.3 Data splitting

Since the "Kaggle.com" does not provide test solution, we use the training data provided as our whole dataset. The original dataset has already been randomly shuffled, thus a partition method has been adopted to split the dataset into training data and test data. After splitting, the training data is 2,000,000 while the test data is 763,588.

4. Modeling

4.1 Multiple linear regression

(1) Ordinary Least Squares

After data pre-processing, the Ordinary Least Squares (OLR) is performed at first. 10-fold cross validation is used for the training set. Considering that some predictors are highly correlated (key variables with their related 10th, 50th, 90th percentile values), we only use the key variables together with the general information as predictors.

The result of ordinary linear regression is displayed in **Table 2**. In this table, it is found that Kdp and Zdr are not significant and therefore not included in the linear regression model.

Figure 5 displays the observed values versus the predicted values, as well as the residuals versus the predicted values. The predicted values do not fit the observed value very well and the residuals do not follow a random pattern along the predicted value. This indicates that the OLS does perform well and some nonlinear behaviors may also exist.

Table 2: Coefficients of ordinary linear regression model.

Coefficients	Estimate	Std. Erro	r t value	Pr(>|t|)	
(Intercept)	-1.16E+00	8.08E-02	-14.342	<2e-16	***
radarDist	3.04E-02	7.24E-04	42.019	<2e-16	***
Ref	2.01E-02	1.04E-03	19.252	<2e-16	***
RefComposite	1.42E-02	9.99E-04	14.186	<2e-16	***
RhoHV	1.05E+00	8.11E-02	12.898	<2e-16	***
Zdr	2.73E-03	3.79E-03	0.72	0.471	
Kdp	-1.55E-03	1.42E-03	-1.091	0.275	
RainMP	4.22E-04	9.09E-06	46.421	<2e-16	***

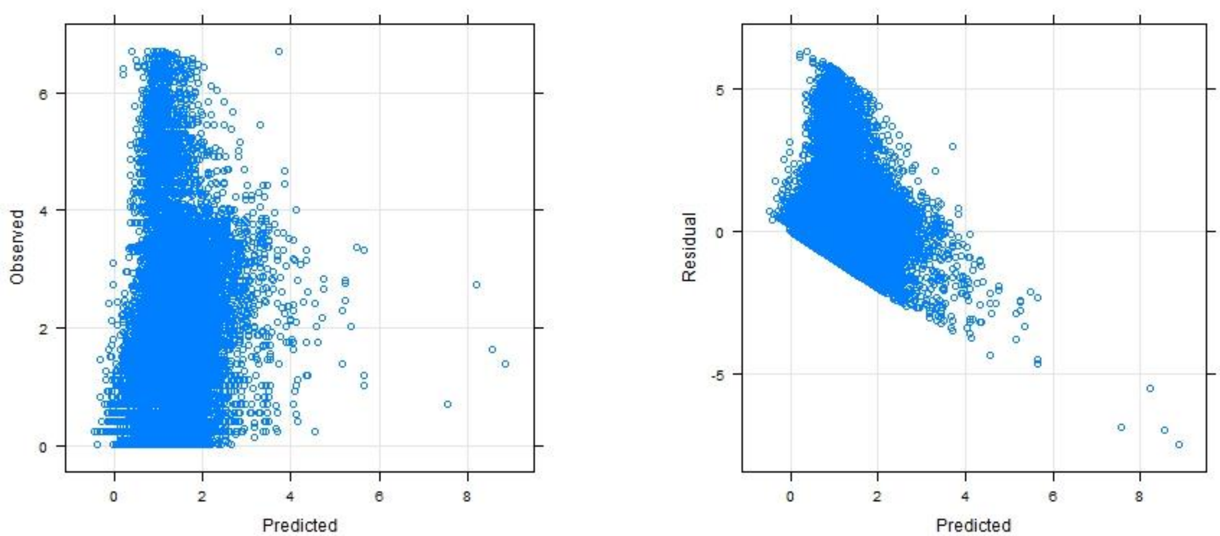

Figure 5: Left side: Observed versus predicted values for ordinary linear regression; Right side: Residuals versus predicted values for the ordinary linear regression (Graphs are in logarithmic scale).

(2) Partial least squares
To consider the multicollinearity and also reduce the dimension, the partial least squares method is carried out. Also, the 10-fold cross validation is used for the training set. The result turns out that only 3 components are needed as shown in **Figure 6** Left.

4.2 Neural network

The neural network is a nonlinear regression technique which uses the original predictors to construct an intermediary set of unobserved variables (named hidden units) and then uses these unobserved variables to model the outcome. 5-fold cross validation is used and two parameters are tuned. It turns out that the number of units in the hidden layer equals 18 and parameter for weight decay equals 0.01 (see **Figure 6** Right). The predicted values versus observed values for both training and test data are displayed in **Figure 7**.

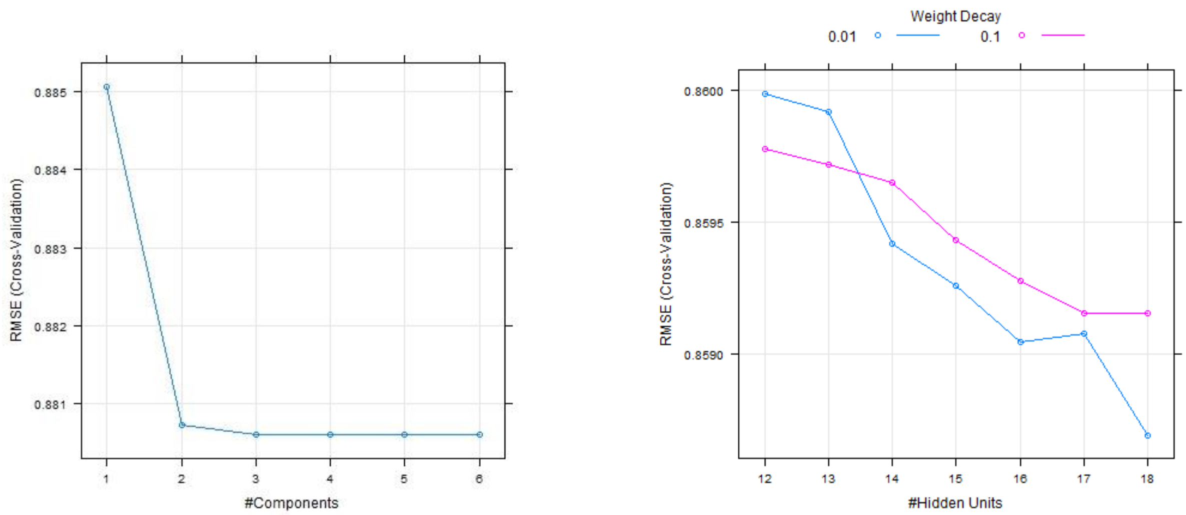

Figure 6: Left side: Tuning parameter for the PLS; Right side: Tuning parameter for the neural network.

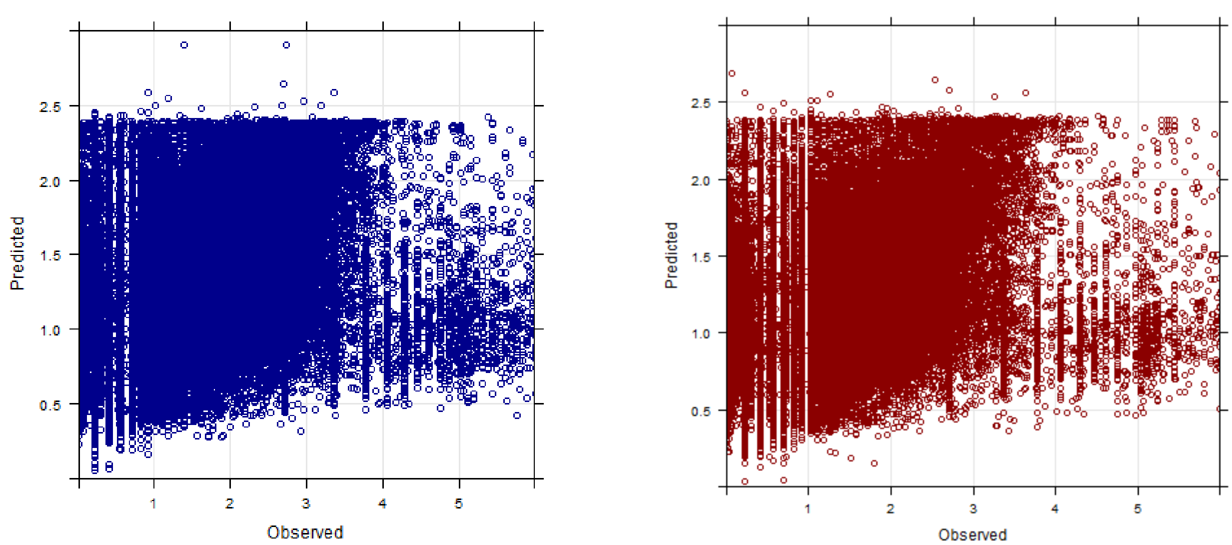

Figure 7: Left side: Predicted versus observed for training data; Right side: Predicted versus observed for test data (Graphs are in logarithmic scale).

4.3 Stochastic Gradient Boosting

Stochastic Gradient Boosting is a boosting regression tree with a little modification to prevent overfitting. This model uses RMSE as a metric. This model has four tuning parameters, minimum number of observations in leaves, learning rate, complexity of the tree and number of trees. For tuning the parameters we keep the learning rate equal to 0.1 and the minimum number of observation in each node equal to 20. Also, we tried three different complexity of 2, 3 and 4 and different number of trees ranging from 100 to 4000. Using 5-fold cross validation the model finds the optimal parameters with lowest RMSE as shown in **Figure 8**. The predicted versus observed values by this model is displayed in **Figure 9** Left and Right for training and test data respectively.

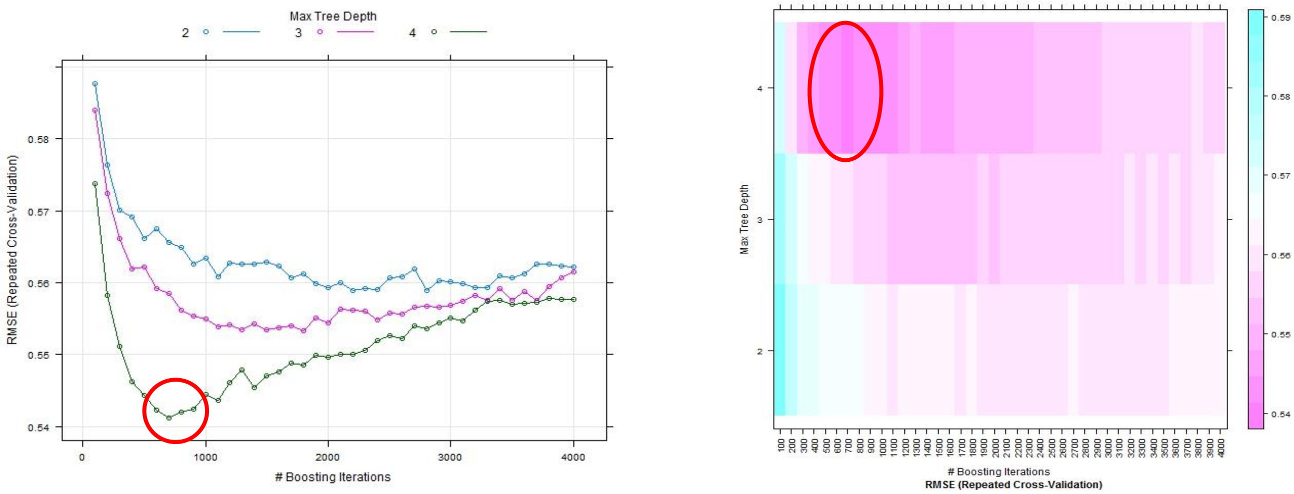

Figure 8: Tuning parameters for the stochastic gradient boosting.

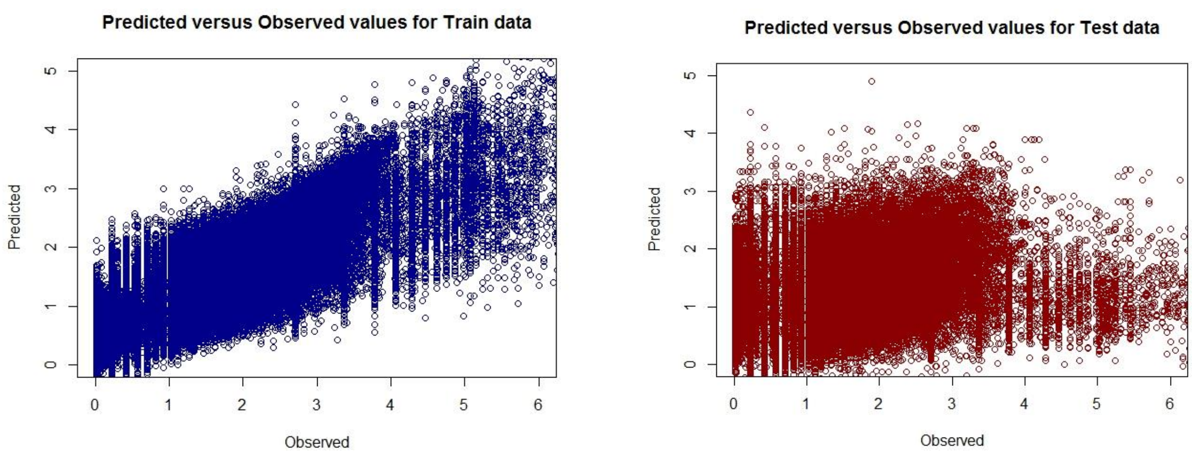

Figure 9: Left side: Predicted versus observed value for training data; Right side: Predicted versus observed value for test data (Graphs are in logarithmic scale).

4.4 eXtreme Gradient Boosting (XGBoost)

XGBoost is a technique to optimize the boosted tree algorithms. The goal of this function is to push the extreme of the computation limits of machines to provide a scalable, portable and accurate for large scale tree boosting. In this method, there are several tuning parameters such as learning rate, gamma, maximum depth, minimum sum of instance weight, subsample ratio, number of max iterations and watchlist (See **Figure 10** Left).

XGBoost provides a watchlist parameter which can be used to watch the training process and calculate the real time MAE values in validation set. Thus, here the training data for each distance is divided further into training and validation subset. The proportion of training and validation subset is 80% and 20% of the whole training data. During the training process, the function will supervise both the MAE for training subset as well as for validation. Due to the larger numbers of parameters in XGBoost, the parameters are tuned manually one by one, i.e., tuning one parameter each time to get the minimum MAE given all other tuning parameters constant.

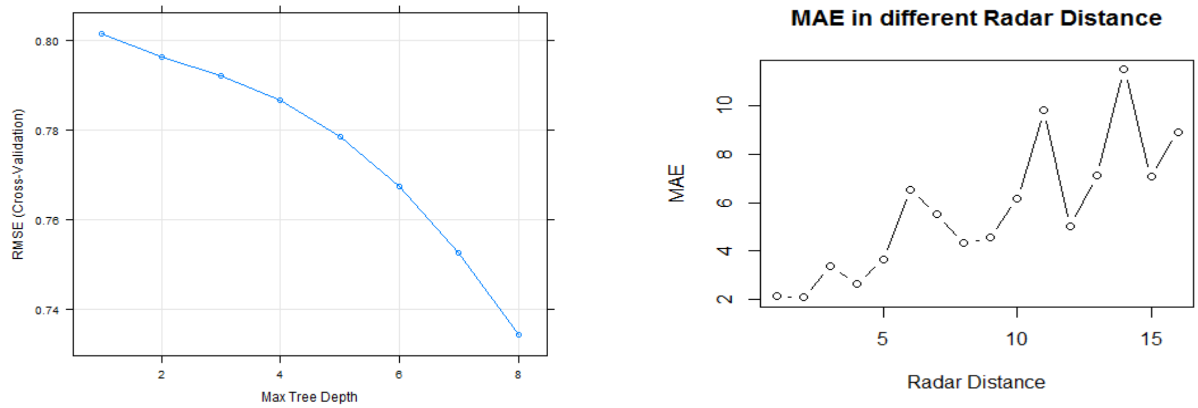

Figure 10: Left side: Tuning parameter for XGBoost; Right side: MAE for different radar distances.

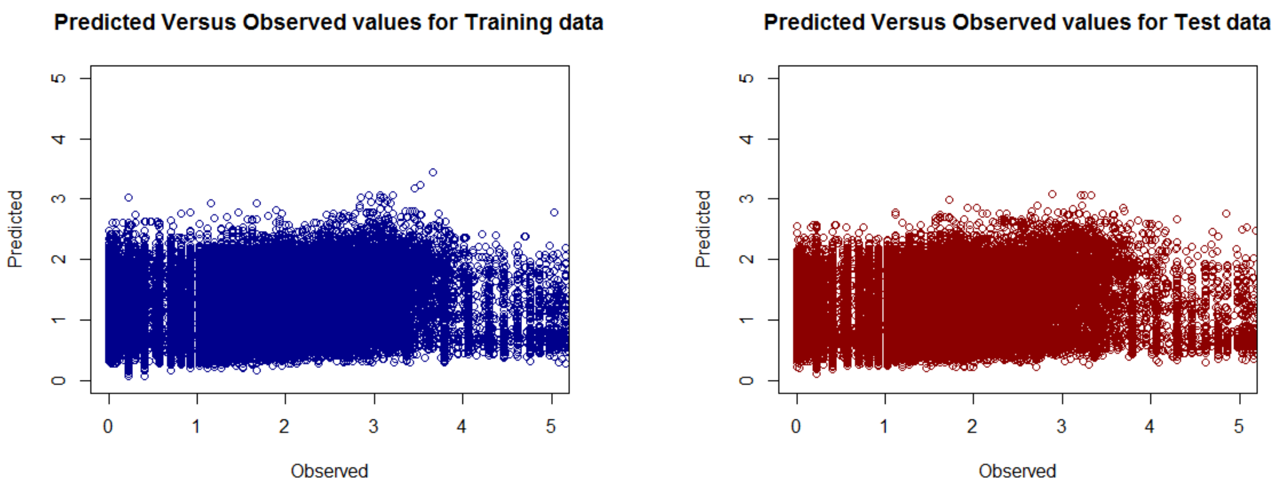

Figure 11: Left side: observed versus predicted for training data; Right side: observed versus predicted for test data (Graphs are in logarithmic scale).

Comparing the predicted with the observed in **Figure 11**, we find that the model cannot predict the extreme values either in training or test data. But for the test values less than 2.5, the XGboost model can predict them with higher accuracy. It should be noted that there is a clear cut in the predicted values in the plots above, this is because rainfall values greater than 200 mm/h is removed intentionally in the XGBoost model while the rainfall in test data is all preserved. The truncate in 200mm can be explained in this way: In realty, the probability of having a rainfall more than 200mm/h can be seen as an extreme event. It is not wise to include such extreme value in the model. This assumption is verified by test MAE. After removing the extreme values in training set, the test MAE is greatly reduced. MAE for different radar distances is displayed in **Figure 10** Right. As the distance increases MAE increases. This is due to the fact that reliability of the radar output is reduced when the distance between radar and rain gauges increases.

5 Results and discussion

In the modeling part, we have thoroughly discussed the modeling techniques, cross validation strategy and hyper-parameter tuning in each model. In this part, the results of five models are summarized (see **Table 3**) and discussed as follows:

Generally, relatively complex models such as Neural network and XGBoost can yield better test scores (6.08 and 6.161 respectively) than simple models such as OLS and PLS (6.334 and 6.322 respectively) in terms of MAE.

For tree-based models (Stochastic Gradient Boosting and XGBoost), the cross validation errors are much smaller than the other models while the test errors do not outperform the other models. It suggests a sign of overfitting in the boosting trees models.

Another issue to be noted is the nuance relationship between MAE and R^2. For OLS, PLS and Neural Network, smaller MAE always relates to larger R^2 (both training and test). However, for tree-based models, it shows an opposite relation. This seems weird since it does reflect the nature of these two metrics. Actually it is because the R^2 amplifies the extreme values by squaring the errors while the MAE measures the mean absolute errors. It is possible to yield smaller R^2 and MAE at the same time if the model concentrates more on predicting small values while emphasizing less on extreme values.

In Part **3.2** feather engineering, the distance is introduced as an important feature. In this part, such idea is supported by the modeling results. In **Figure 10** Left, it can be seen that there is a clear relationship between test MAE and radar distance (the fluctuating points are due to extreme values in the test data). When the distance becomes larger, the radar tends to capture different contents of precipitation (rain, ice or snow). Moreover, as the distance increases, the signal tends to be contaminated more by other non-meteorology matters thus relatively more "noise". It should be noted that the modeling based on distance is only employed in tree-based models. For the first three models, there is no sign of improvements, which is quite interesting.

From **Figure 5, 7, 9** and **11**, it can be found that models predict better for smaller rainfall than larger rainfall. This is due to the following reasons: on the one hand, since our objective is to obtain the smallest MAE and the majority of the actual rainfall values are less than 10 mm (the definition of heavy rain varies from 7.6 to 40mm per hour from different organizations, but for sites in the US, rainfall greater than10mm per hour can be treated as extreme ones [6]), the models developed tend to focus more on the smaller observations. On the other hand, it is believed that many factors are related to extreme rainfalls and the mechanism is quite different from that of small and medium ones. It is suggested that future research should develop models specifically for extreme rainfalls by building new features reflecting their unique nature.

Table 3: Summary of the model errors.

Model	**Training**	**Test**

	R squared	MAE	R squared	MAE
Linear regression	**0.145**	**5.725**	**0.145**	**6.334**
Partial least squares	**0.145**	**5.711**	**0.136**	**6.322**
Neural network	**0.192**	**5.460**	**0.180**	**6.080**
Boosting Trees	**0.463**	**4.136**	**0.323**	**6.299**
Xgboost	**0.252**	**3.771**	**0.223**	**6.161**

6 Conclusions

In meteorology, there are two types of measurements: radar and rain gauges. However, these two types of measurements never matched with each other because of the different sources of error. The overall objective of this project is using the snapshots of polarimetric radar observations to predict the accumulated hourly rain measured at rain gauges. Since the rain is itself a very complex process, we used different data mining models to estimate and explain the relationship between these two types of precipitation measurements, and thus establishing a new data-based methodology for estimating hourly rainfall based on radar observations.

The data mining models performed include linear regression, partial least squares and nonlinear models such as neural network, Stochastic Gradient Boosting and XGBoost. As explained above, R^2 is not a good metric when large errors are particularly undesirable because R^2 tends to penalize the extreme error while MAE measures the mean absolute error. So, MAE is used as the main metric for evaluating the models.

There are several findings from our research:

i) Neural network outperforms other models in terms of MAE value.

ii) There is a nuance relationship between MAE and R^2. The smaller MAE does not necessarily result in larger R^2. In this particular project, the MAE is more useful because in some cases the R^2 tends to penalize extreme error (for extreme rainfall) which is undesirable.

iii) Feature Engineering is important for model improvement. This project develops three major features based on the mechanism of precipitation in meteorology. It is found that some models gain considerable improvements by including these features while the other models do not.

iv) Ref, RefComposite (i.e., reflectivity from horizontal and vertical radar waves) are found to be the most important factors related to rainfall.

v) Models are inclined to predict better outcomes for rainfall ranging from small to medium values. However, it underestimates large values (actual rainfall greater than 10mm per hour).

In summary, rainfall is hard to predict, especially for extreme values. With the help of state-of-the-art Polarimetric Radar, we can extract more complex features from the data set and build more robust models to better reflect the nature of rainfall. However, rain is such a complex process with a lot of uncertain factors involved that merely the radar observation is not enough to precisely estimate and predict rainfall. It would be desirable if more measures such as surface temperature and air pressure could be incorporated in future models. Human beings still have a long way to go to precisely predict the rainfall. Hopefully, we are in the right way with the help of “Bid Data” analysis.

References

[1] https://www.kaggle.com/c/how-much-did-it-rain-ii
[2] http://www.nws.noaa.gov/com/weatherreadynation/news/130425_dualpol.html#.VnCTdvlViko
[3] Willmott, Cort J., and Kenji Matsuura. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance." Climate research 30.1 (2005): 79.

[4]https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwjW5_eJyd_JAhVM8WMKHRyiDhQQFggvMAI&url=http%3A%2F%2Fwww.erh.noaa.gov%2Fbgm%2Fresearch%2FERFFW%2Fpresentations%2Fjune_03_2010%2Fplenary%2Fjunker_wes_keynote_extreme_rainfall_2010.ppt&usg=AFQjCNG-M1QAqFWfAgAZQnxCBunQYZyFOg&sig2=WAWWVo6wsL91Ipt23K11_w&bvm=bv.110151844,d.cGc

[5] J. S. Marshall and W. M. Palmer, "The Distribution of Raindrops with Size," Journal of Meteorology, Vol. 5, No. 4, 1948, pp. 165-166.

[6] Llasat, María-Carmen. "An objective classification of rainfall events on the basis of their convective features: application to rainfall intensity in the northeast of Spain." International Journal of Climatology 21.11 (2001): 1385-1400.

Appendix

Parameters for the XGBoost model

booster	*which booster to use. Set "booster" ="gbtree";*
eta	*Shrinkage. Control the learning rate: scale the contribution of each tree by a factor of 0 < eta < 1 when it is added to the current approximation. Set "eta" =0.03;*
gamma	*minimum loss reduction required to make a further partition on a leaf node of the tree. the larger, the more conservative the algorithm will be. Set "gamma" = 0.1;*
max_depth	*max depth if a tree. Set "max_depth" = 2;*
min_child_weight	*minimum sum of instance weight(hessian) needed in a child. In linear regression mode, this simply corresponds to minimum number of instances needed to be in each node. Set "min_child_weight " = 4;*
colsample_bytree	*subsample ratio of columns when constructing each tree. Set "colsample_ bytree " = 0.55;*
subsample	*subsample ratio of the training instance. Set "subsample " =0.7;*
nrounds	*the number of max iterations)=1000.*
eval_metric	*evaluation metrics for validation data. Here user defined function MAE.eval.f is used to calculate the evaluation metric. The user defined function MAE.eval.f can be found in Appendix A.*
watchlist	*Watchlist is used to specify validation set monitoring during training. Set "watchlist" = list(validation_data, training_data),*
early.stop.round	*If set to an integer k, training with a validation set will stop if the performance keeps getting worse consecutively for k rounds. Set "early.stop.round" = 100;*
maximize	*maximize=TRUE means the larger the evaluation score the better. Set "maximize " = FLASE;*