ECEN-689 Applied Information Science

Rainfall Prediction

Report of Project-2

By

Harish Chigurupati

Ranjith Tamil Selvan

Shabarish Kumar Rajendra Prasad

Introduction

The goal of this project is to predict the possibility of rain in any of the 41 given cities in Australia, given the weather conditions for that particular day. Prediction of precipitation has numerous applications and scopes. This project is emphasized on studying various statistical meteorological and oceanographic data such as temperature, humidity, pressure, wind-speed, thereby identifying valuable indicators and consequently designing a viable model. Moreover, the performance of the different prediction techniques are evaluated.

Rainfall prediction is important to government departments, disaster management organisations, sea and air travel, agriculture and farming, insurance companies. Rainfall also is important to other living organisms - plants and animals. The data used in this project is obtained kaggle dataset here. This dataset consists of meteorological data from across 41 cities in Australia for over 8 years, i.e. from 2009 to 2017. It consists of 24 columns and 145,000 rows. But still this database has many missing values for some of the important features we are planning to use in this project. While using this data in this project, we would clean up the records with missing values.

Methodology

We are going to approach this problem in two different ways. First, we use the complete data available in the dataset and try to find an optimal model to fit the data. Next we would try to extract more features and use feature crossing techniques to check whether it would improve the performance of the designed models. Since the problem at hand is categorical, the models that we plan to use include logistic regression, SGD classifier, KNN classifier, SVM classifier, decision tree classifier, random forest classifier and neural networks.

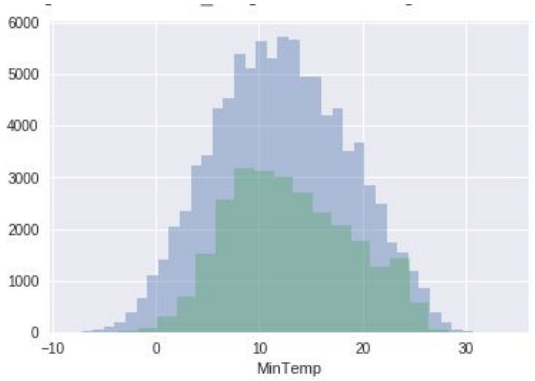
Without any feature manipulation

Features

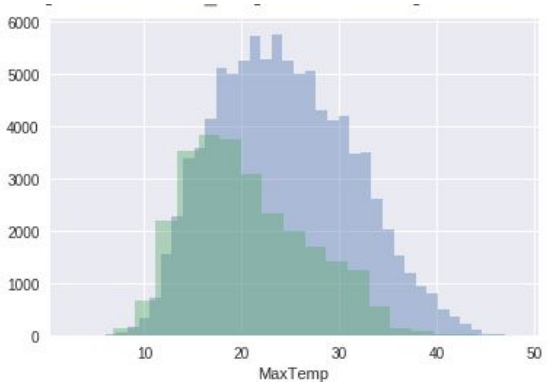
The entire dataset consists of 24 columns. However for our project we are only interested in the following 12 features for training with the 'RainToday' column as the target. The features we consider we include:

1. Location
2. Minimum Temperature
3. Maximum Temperature
4. Wind Gust Speed
5. Wind Speed at 9.00AM
6. Wind Speed at 3.00PM
7. Humidity at 9.00AM
8. Humidity at 3.00PM
9. Pressure at 9.00AM
10. Pressure at 3.00PM
11. Temperature at 9.00AM
12. Temperature at 3.00PM

All these features have little to no correlation with the the target variable. For example we have visualized an histogram of the minimum and maximum temperatures versus the chances of rainfall, which can be found below,



Minimum Temperature vs No. of occurences



Maximum Temperature vs No. of occurences

As we can see from the histogram representation of the the temperature values vs. no. of occurrences, the histogram of the positive samples overlap with those of the negative samples. This is the same case with mostly all the features in the dataset. But still we will try fitting models on these feature and find how they perform on the feature set they have. We believe that the performance of these features are not all that efficient.

Logistic Regression

While fitting the logistic regression model on the feature set we are able to achieve an accuracy of about about 83.51% with all the hyperparameters tuned to their maximum efficiency.

SGD classifier

The SGD classifier is able to achieve an accuracy of 83.05% on the same dataset. This value is similar to that obtained from logistic regression model seen earlier, but still we believe other models can perform better on the same dataset.

KNN classifier

The KNN classifier is able to achieve an accuracy of 82.99% on the dataset with 5-NN. Increasing the number of neighbours result seem to generate better performances. 15-NN gets an accuracy of 83.94%, 31-NN was able to achieve 84.01% and 55-NN ,an accuracy of 84.23%. But the value of accuracy is found to be saturated around 85% and it is hard to tune the value of N to achieve an accuracy higher than 85%.

SVM Classifier

With ‘Sigmoid’ kernel SVM classifier on the dataset, we are able to achieve an accuracy of 77.95%. With ‘RBF’ (Radial basis function) kernel, we are able to achieve around 83.4%

Random Forest Classifier

The random forest classifier was able to achieve an accuracy of 85.11% with the number of estimators equal to 100. While tuning the number of estimators uptill 200 it was able to achieve an accuracy of 85.44% and upon tuning the number of estimators up, the accuracy value is found to saturate around 85.49%.

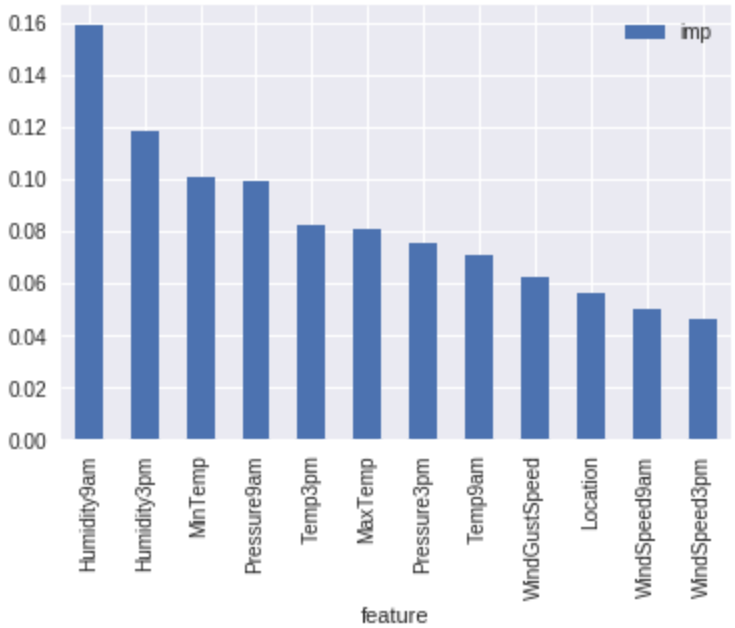
Neural Networks

We designed a neural network with two hidden layers, with the number of node in each hidden layer being 12 and 5 respectively and relu as the activation function. This network was able to achieve an accuracy of 88.32% after training using adam optimizer for around a 1000 epochs, with all the hyper parameters tuned.

Feature Manipulations

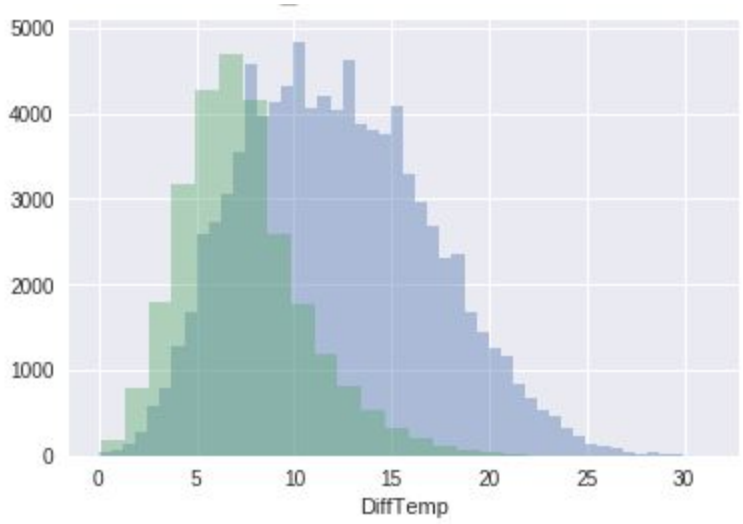
As we are not able to achieve even around 90% accuracy with available data with just the available features, we are interested in creating new features that might help us achieve a better accuracy on the dataset.

To understand what are the features that are most important in our dataset, we shall use the random forest classifier to find the contribution of each feature towards the classification. The bar graph below shows the contribution of each feature towards the classification using random forest classifier:



Now that we know how important each feature is, we shall try adding more features and see whether we can add more important features.

First we tried to create temperature difference feature from the minimum and maximum temperature of the day. The histogram representation of the temperature difference is shown below:



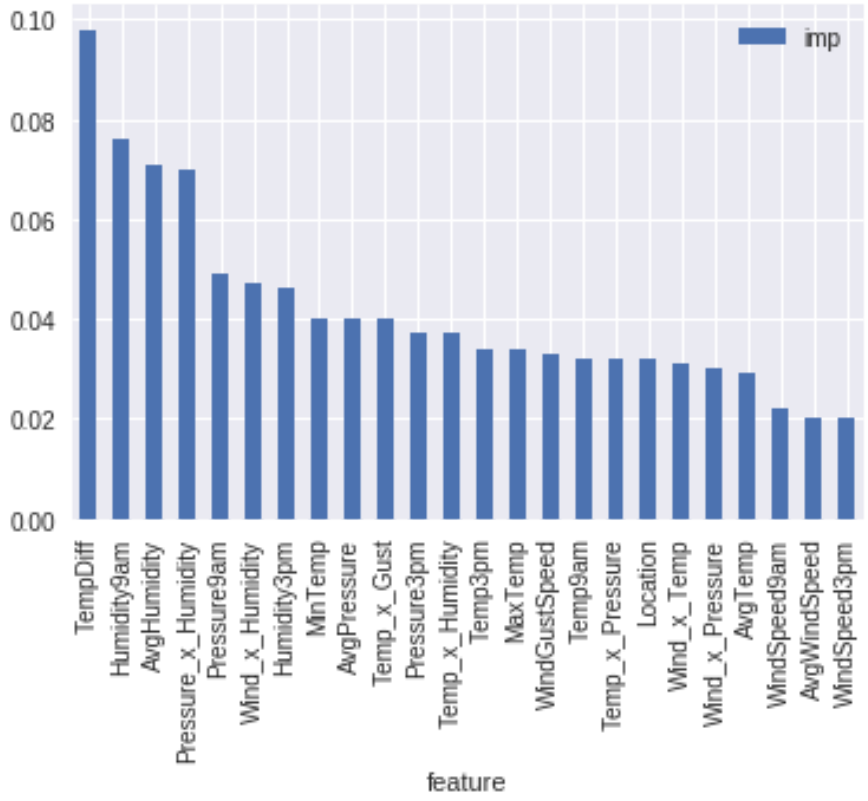
As we can see, this feature has a little better way of differentiating between the rainy days and the days without rain. So we tried to extract more cross features like these. Below are the list of the new features we have introduced into the dataset and the description about the feature:

1. Difference between the maximum and minimum temperature of the day.
2. Average wind speed between wind speed at 9.00am and the wind speed at 3.00pm
3. Average pressure between the pressure at 9.00am and the pressure at 3.00pm
4. Average humidity between the humidity at 9.00am and the humidity at 3.00pm
5. Average temperature between the temperature at 9.00am and the temperature at 3.00pm
6. The average temperature of the day multiplied by the average wind speed of the day.
7. The average pressure of the day multiplied by the average humidity of the day.
8. The average temperature of the day multiplied by the average humidity of the day.

9. The average temperature of the day multiplied by the average pressure of the day.
10. The average pressure of the day multiplied by the average wind speed of the day.
11. The average humidity of the day multiplied by the average wind speed of the day.
12. The difference in temperature multiplied by the wind gust speed of the day.

Now our dataset contains 24 features in total. In addition to the newly added features, we are also interested in performing PCA on these features.

First we shall find whether the features we added to the dataset are important while classifying the data. We shall use the same random forest classifier to find the importance of each of the 24 feature during classification. The bar graph below shows the contribution of each feature during classification:



We can see that most of the features we added have become more important that the features that were already existing in the dataset. A few of the new features have also turned out to be less important, so we shall use the feature extraction techniques to extract only the features that discriminate between the classes.

The goal of doing PCA over the dataset is to extract around 90% of the variance explained in the dataset. After fitting our training data in a PCA model, the variance explained by different number of components is listed below:

- 1 - component = 37.83%
- 2 - components = 59.28%
- 3 - components = 74.67%
- 4 - components = 82.54%
- 5 - components = 87.79%

So we extract only the first 5 components as they explain around 88% of the variance among the 24 features.

We have trained each of the six previously mentioned models on the new feature set that we have extracted and the results of the accuracy obtained on the test set after training are listed below:

- Logistic Regression - 90.41%
- SGD Classifier - 89.28%
- KNN Classifier (15-NN) - 89.12%
- SVM classifier (rbf) - 91.36%
- Random Forest Classifier - 93.23%
- Neural networks - 93.84%

We can see that all the models perform much better than they did without any feature manipulations. Neural networks give the best performance of around 94% accuracy on the test set.

Results and discussion

Performance Overview

For performance of each of the models, we have considered three attributes, namely - F1-score, Precision and Recall score values.

The F1 score (Balanced F-score / F-measure) is the weighted average of precision and recall, the maximum and minimum values being 1 and 0 respectively. The relative contribution of precision and recall to the F1 score are equal.

The formula for the F1 score is:

F1 = 2 * (precision * recall) / (precision + recall)

The precision is the ratio tp / (tp + fp) , where tp is the number of true positives and fp the number of false positives. The precision denotes the ability of the classifier to avoid false-positives. **The recall** is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall denotes the ability of the classifier to find all the positive samples.

We have calculated the F1-score, precision and recall values of each of the classifier and listed them below:

Without Feature manipulations:

Classifier	F1-Score (weighted)	Precision (weighted)	Recall (weighted)
Logistic Regression	0.84	0.83	0.86
SGD Classifier	0.84	0.83	0.86
KNN Classifier	0.84	0.83	0.86
SVM Classifier	0.83	0.83	0.86
Random Forest Classifier	0.85	0.84	0.87
Neural Network	0.88	0.87	0.89

After Feature manipulations:

Classifier	F1-Score (weighted))	Precision (weighted)	Recall (weighted)
Logistic Regression	0.89	0.88	0.90
SGD Classifier	0.88	0.87	0.88
KNN Classifier	0.89	0.87	0.90
SVM Classifier	0.91	0.90	0.93
Random Forest Classifier	0.93	0.91	0.95
Neural Network	0.94	0.92	0.95

Conclusion

This paper investigates the different machine learning techniques that could be used to forecast rain for a given city, given some meteorological and oceanographic features. During the process, appropriate data cleaning and preprocessing techniques were performed. Different models - Logistic Regression, SGD Classifier, KNN Classifier, SVM Classifier, Random Forest Classifier, Neural Networks have been designed and their performance attributes have been observed.

Based on this study :

1. Neural Networks gives best accuracy and performance as compared to the other classifiers such as SGD, KNN, SVM and Random Forest.
2. PCA has been effective during the pre-processing stage. The cross features and the PCA that we included have been instrumental in increasing the accuracy performance of each of the models on the dataset.

References

The following are some of the references which have influenced this project:

1. Dataset : https://www.kaggle.com/jsphyg/weather-dataset-rattle-package
2. İmren Dinç,1 Madhav Sigdel,1 Semih Dinç,1 Madhu S. Sigdel,1 Marc L. Pusey,2 andRamazan S. Aygün1 Evaluation of Normalization and PCA on the Performance of Classifiers for Protein Crystallization Images
3. P. Goswami and Srividya, "A novel Neural Network design for long range prediction of rainfall pattern," Current Sci.(Bangalore), vol. 70,no. 6, pp. 447-457, 1996.
4. Kotsiantis S.B, Kanellopoulos D and Pintelas P.E (2006). Data pre-processing for supervised learning. International Journal of Computer Science, Vol 1 No.2, 111-117.
5. Minns, A. W and Hall, M. J. (1996). Artificial neural networks as rainfall-runoff models. Hydrol. Sci. J.,41, 3 , 399-417
6. Wei-ChiangHong Rainfall forecasting by technological machine learning models https://doi.org/10.1016/j.amc.2007.10.046
7. Hong, W. (2008). Rainfall forecasting by technological machine learning models, Applied Mathematics and Computation 200(1): 41–57.
8. https://skymind.ai/wiki/neural-network
9. Scikit Learn : Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
10. API design for machine learning software: experiences from the scikit-learn project, Buitinck *et al.*, 2013.
11. Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org