Prediction and Analysis of Rainfall Patterns

Maksimjeet Chowdhary (2020566)
maksimjeet20566@iiitd.ac.in

Shagun Mohta (2020468)
shagun20468@iiitd.ac.in

Abhey Kalia (2020420)
abhey20420@iiitd.ac.in

Chaudhary Digvijay Daniel Singh (2020559)
digvijay20559@iiitd.ac.in

1 Abstract

Through this paper, we attempt to perform rainfall prediction using local weather data (Australia) for a Machine Learning course project. Since the dataset has a target variable that is of 0-1 classification type we train different ML models on the dataset and compare the results and try to understand the advantages and shortcomings for each of the ones. GitHub Repo Link

2 Introduction

We chose Rainfall prediction as our project topic because rain is a critical component of the world's weather system.The average annual rainfall on the Shillong Plateau in the North-East can exceed 10,000 mm per year, while in Birdsville, Australia, it can be as low as 133 mm. Floods can be caused by irregular and heavy rain, which can destroy crops and damage property, while extreme rain scarcity can often cause crises such as droughts and famines. We believe that an accurate rainfall forecasting model could be a useful tool in mitigating the negative effects of erratic rainfall patterns, allowing people to take preventive measures early on!

3 Literature Survey

- In paper [1] The paper discusses the use of standard ML techniques in the rainfall prediction problem. Results obtained from nonlinear processes such as ANNs are compared.
- In paper [2] The paper tries to describe the result of an attempt to forecast expected rainfall in a state of India by using various MANN models
- In paper [3] The article discusses the use of fuzzy implementation. The result from different classifier models is collected and the majority is given as the output. This is useful because rainfall determination mimics fuzzy logic instead of binary logic in most cases.

4 Dataset

The dataset contains 10 years (between 2007 and 2017) of weather data with an output column for whether there will be rain on the next day, collected from numerous weather stations [2].The attributes available to us are: date of observation, wind speed and directions at 9 AM and 3 PM, minimum and maximum temperatures recorded and similar characteristics. Most of the columns had less than 10 percentage of null values and 4 columns had more than 40 percentage of the data missing (Evaporation, Sunshine, Cloud9am,Cloud3pm).

4.1 Correlation

Inferences from the correlation Heatmap:

1. As expected Temp and Pressure for a day remained more or less similar throughout the day as hinted by the positive correlation between Tempat9am and Tempat3pm and same for Pressures.

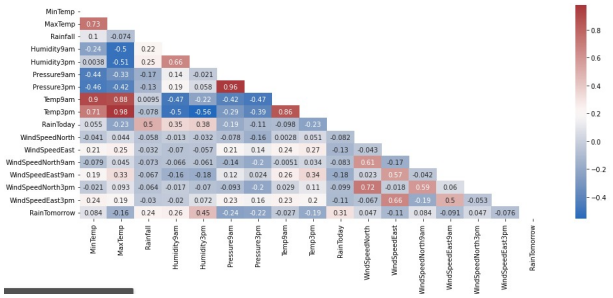

Figure 1: Correlation matrix

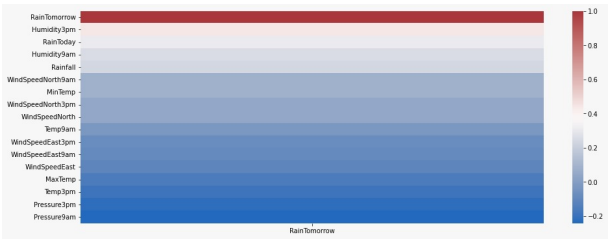

Figure 2: Correlation of each feature with the Target Variable

2. MaxTemp and Tempat3pm have high correlation because logically the day has highest temperature during and around afternoon as majority of the heat has been absorbed by the atmosphere by 3pm

3. MinTemp and Tempat9am are also largely positive correlated as lower MinTemp also implies lower temperature readings at 9am. Analogous logic applies for MaxTemp and Tempat3pm.

4.2 Preprocessing

First we handled null values in the dataset with various techniques. For attributes that had a small part of data missing (2-3 percent), the median values were inserted instead of missing data. For attributes that had around 8-10 percent of the data missing, KNN Imputing was applied which models the data into multiple dimensions, accurately predicting clusters of size k (a hyperparameter) and taking the average of the formed cluster for a specific attribute to be imputed in place of the missing value.

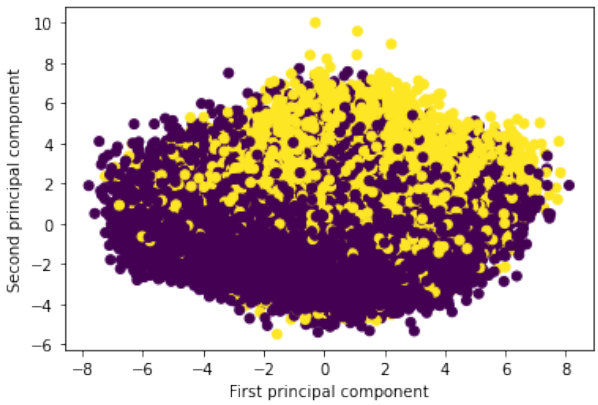

Figure 3: 2D Plot of the PCA reduced data

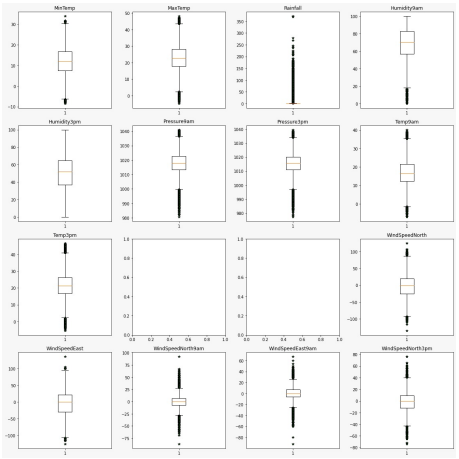

Figure 4: Boxplots of the Features

Attributes containing wind direction were categorical. In order to rid of a discrete distribution which would hamper the learning of the model we combined the direction and speed given for the wind into their vertical and horizontal components. Finally, we standardised the data.

4.3 PCA and Oversampling

We performed PCA on the data, getting the best 10 attributes to perform our classification and regression on. We also parallelly performed Oversampling on the base Data, along with PCA on the Oversampling Data. We tried our models on these four vairants of the Datasets.

5 Methodology

5.1 Classification

Various ML models were applied on the given dataset to maximize the accuracy of predicting rain for the next day. The original dataset was skewed towards the 0-value of the target-variable and hence weighted precision score was calculated rather than accuracy for most of the models evaluated.

1. **Gaussian Naive Bayes** model was trained to be the benchmark for the other models

2. **SVM Linear Kernel, SVM Poly Kernel, and SVM RBF Kernels** were trained with

3. **Logistic Regression** model was trained, and cross validated.

4. **Decision Trees** we tried for different depths, along with both trying Gini Index and Entropy for these depths.

5. **Random Forests** was trained with max depth ranging from 3 to 7, taking randomly half the data points and considering randomly half the features for each of the 100 trees.

6. **AdaBoost** model was trained with the resultant model from Decision Tree as the base model, and considering different n-estimators values

Algorithm	Normal	PCA	Sampling	Both
Linear Regression	0.295	0.274	0.34	0.32
Decision Tree Regressor	0.42	0.40	0.45	0.43
Random Forest Regressor	0.52	0.49	0.57	0.53

Table 1: R2 Scores of Regression Algorithms

Algorithm	Normal	PCA	Sampling	Both
Gaussian Naive Bayes	0.798	0.795	0.25	0.65
SVM Linear Kernel	0.70	0.766	0.824	-
SVM Poly Kernel	0.72	0.785	0.851	-
SVM RBF Kernel	0.74	0.766	0.844	-
Logistic Regression	0.786	0.702	0.798	-
Decision Tree	0.826	0.792	0.770	0.760
Random Forest	0.815	0.803	0.556	0.616
AdaBoost	0.830	0.781	0.798	0.798
xGBoost	0.743	0.737	0.799	0.652
KNNeighbours	0.784	0.711	0.479	0.624
Multi Layer Perceptron	0.732	0.720	0.499	0.652

Table 2: Precision Values of Classification Algorithms

7. **Multi Layer Perceptron** model was trained with the and considering different n-estimators values

8. **xGBoost** model was trained with objective binary classification with Logistic Activation.

9. **K Nearest Neighbours** model was trained with searching for values of K ranging from 1 to 9, for all the different versions of the dataset.

5.2 Regression

Parallelly, we turned this into a Regression Problem, with the aspirations that instead of a Binary outcome of whether it rains the day after, getting an idea of exactly how much it would rain can benefit the general public. Till now in this dataset we talked about Rain as atleast 1mm of Rainfall. If we were to talk of rain as Light, Medium, and Heavy, a better distribution would be below 2.5mm as Light Rainfall, below 7.5mm as Medium Rainfall, and above 7.5mm as Heavy Rainfall. We used three different Regression Models on our Data, and the models

used were Linear Regression, Decision Tree Regression, and Random Forest Regression.

1. **Linear Regression**
2. **Decision Tree Regressor**
3. **Random Forest Regressor**

6 Results and Analysis

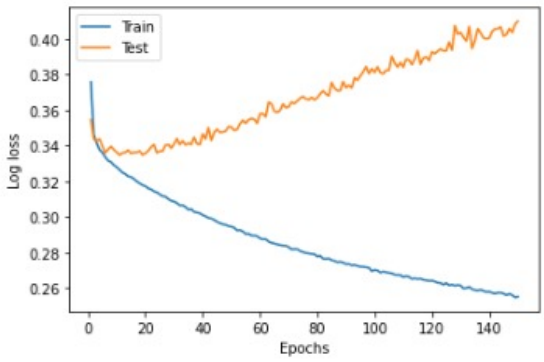

Figure 5: Multi Layer Perceptron Loss Curve

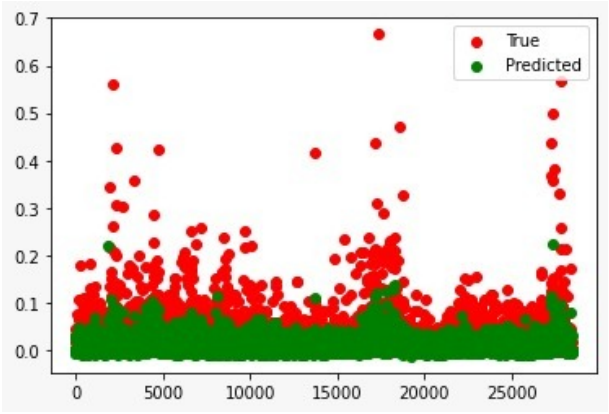

Figure 6: Linear Regression Result

The precision of the Gaussian Naive Bayes Model comes out to be equal to 79.8 percent(Table 1). This result

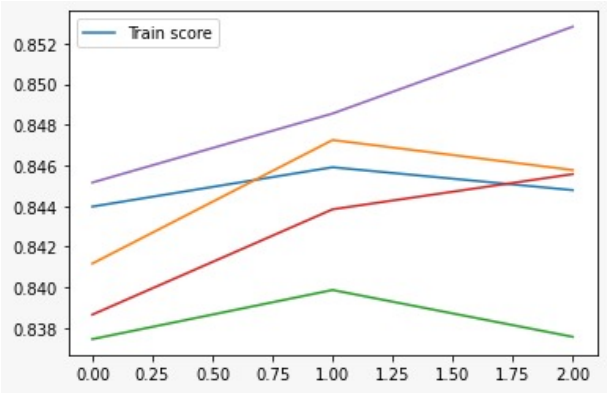

Figure 7: Random Forest Training Cross Validation

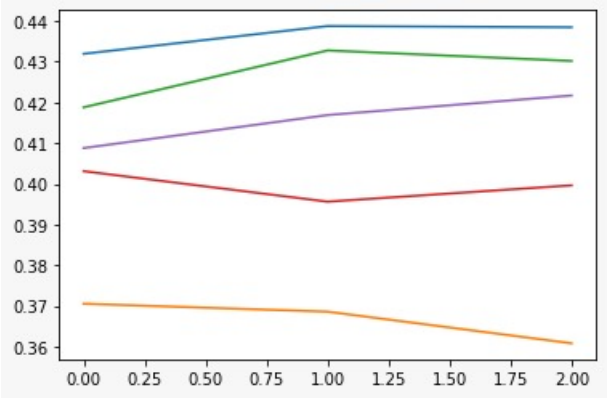

Figure 8: Random Forest Testing Cross Validation

can be expected to be a good threshold of the precision for the model. For further models applied we used this number as the threshold for consideration of the model in our final results. Any models that had precision less than 79 percent can be dropped. Despite the simplicity (some may say oversimplification), Naive Bayes gives a decent performance in many applications. We find the all the Linear Models, the Support Vector Machines and Logistic Regression returns results worse than that of Gaussian Naive Bayes. Whereas Non-Linear Models, the Decision Tree, Random Forests and the AdaBoost gave better results. For Decision Tree we found that the best Decision tree was of depth 8 with Gini Index, giving Precision of 82.6 percent. RF and AdaBoost models were also applied and the respective performance is recorded in the table given in Table 1. The highest performance recorded was for AdaBoost where the base model was the Decision Tree we found, having depth 8 with Gini Index, and at n-components 5. From Regression, we found that a Random Forest Regression model provided the best results whereas the Normal Linear Regression model provided the worst results of the three. We found that Oversampling provided poor results in our case, after performing PCA on the Oversampled Data, the results stayed the same. Only performing PCA on the Data gave somewhat better results but they were still relatively worse than training straight on the whole Data, whose downside is that it would cost more time.

7 Conclusion

We can conclude that Non-Linear models work better than Linear Models as the database is not Linearly Seperable. We can also infer based on our predictions that Australia's rainfall pattern is pretty irregular as evident from the fact that El-Nino winds are a possible cause for variability in rainfall and humidity and large parts of the country is arid and desert-like. In our EDA step, we discovered that our data set is highly imbalanced. To overcome this imbalanced, we tried to utilise oversampling. The models could not train properly on the Oversampled data as the models would train to always predict for a particular Class. We can overcome this with a better loss function that does not treat all the misclassifications equally. In the Regression Problem, we found that

7.1 Future Goals

We would like to try multiple loss functions and ratios and discover the best such that all the misclassifications would not get treated equally, rather it would help us with the Over Sampled data. This would help us take advantage of the Over Sampling we did that did not result positively for us.

7.2 Work Contribution

- **Abhey** EDA, Preprocessing, Random Forest, Multi Layer Perceptrons, Regression Models
- **Shagun** Literature Survey, EDA, PCA, Naive Bayes, Support Vectors Machines
- **Digvijay** Logistic Regression, Decision Tree, RandomForest, AdaBoost, xGBoost, Over-Sampling
- **Maksimjeet** Data Cleaning, Preprocessing, EDA, SVMs, Multi Layer Perceptrons, Regression Models

7.3 References

1. Machine Learning Techniques For Rainfall Prediction: A Review (ICIIECS)
2. Rainfall Prediction Model using Artificial Neural Network (ICSGRC)
3. Rainfall Prediction System Using Machine Learning Fusion for Smart Cities(MDPI)
4. Dataset