Machine Learning Based Rain fall Prediction

Mr. V Naveen Kumar(Assistant professor),
Deverapally Nithin Kumar, Gudur Manoj Kumar, Marri Sai Laxman, Ramanchi Rushinivas,
Department of CSE,
MALLA REDDY INSTITUTE OF TECHNOLOGY AND SCIENCE, Telangana, Hyderabad.

ABSTRACT:

Precipitation forecasting is one of the most challenging and uncertain professions, having far-reaching implications for human society. It is possible to reduce risk and save money with timely and precise predictions. Using the present meteorological conditions in important Australian cities as input, this paper recounts a series of experiments that used state-of-the-art machine learning approaches to develop models that could anticipate the chance of rain tomorrow. This comparative study primarily looks at three things: inputs to the model, methodology for the model, and strategies for pre-processing. The results demonstrate the relative dependability of several machine learning algorithms for forecasting rainfall from meteorological data using different evaluation metrics.

INTRODUCTION

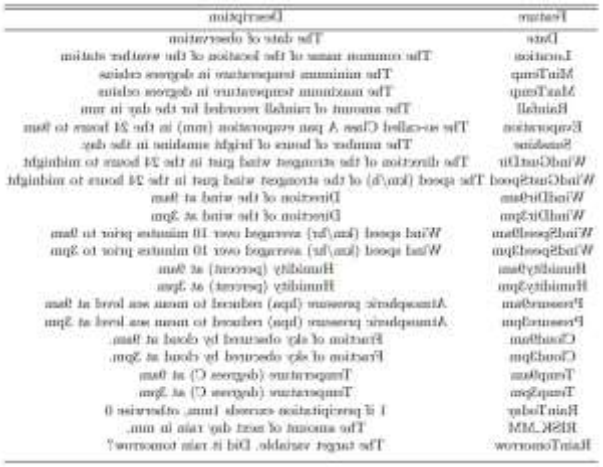
Governments, businesses, risk management organizations, and scientists are all paying close attention to the problem of rainfall forecast because of how serious it is. Many human activities are impacted by rainfall, which is a climatic element [1]. These include agricultural production, building, power generation, forestry, and tourism. Because rainfall is most strongly correlated with disastrous natural disasters including landslides, floods, mass movements, and avalanches, its forecast is crucial. For a long time, these occurrences have had an impact on society [2]. Being able to take precautions and lessen the impact of these natural disasters depends on accurate methods for predicting when and how much rain will fall [3].Several machine learning models and methodologies allowed us to create precise and timely forecasts, allowing us to eliminate this ambiguity. From data preparation to model implementation and evaluation, these publications attempt to cover it all in the machine learning life cycle. The stages involved in data preprocessing include filling in missing values, transforming features, encoding features, scaling features, and selecting features. Several models were put into action, including Decision Tree, Rule-based, Logistic Regression, K Nearest Neighbor, and Ensembles. For the purpose of assessment

As criteria for assessment, we used Accuracy, Precision, Recall, F-Score, and Area Under Curve. The weather data used to train our classifiers in our trials comes from a variety of weather stations across Australia. Following is the outline of the paper. Section 2 begins with a description of the data set that will be considered. While Section displays and discusses the tests and findings, Section presents the methodologies and approaches that were used.

CASESTUDY:

The data collection being discussed in this research includes daily weather observations from many weather stations across Australia. Rain-Tomorrow, which asks if it rained the day after, is the dependent variable of interest. Sure or No. Definitions are derived from http://www.bom.gov.au/climate/dwo/IDCJDW0000 . sham. You can get the dataset at https://www.kaggle.com/jsphyg/weather-dataset-rattle-package. There are 142k cases and 23 characteristics in the dataset. Here are the tires.

Table 1. Data set Description



METHODOLOGY

Data extraction and analysis, data pre-processing, model implementation, and model evaluation make up the overall architecture of this research. These steps are shown in Figure 1.

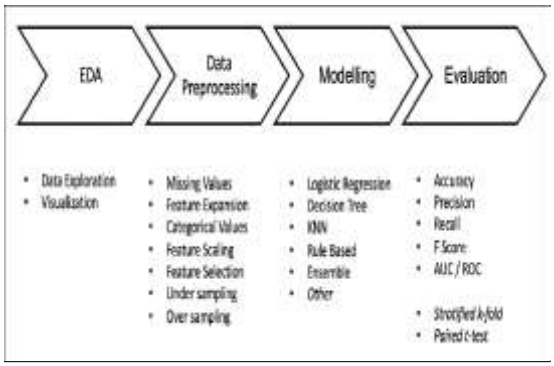


Fig.1.OverallArchitecture.

DATA EXPLORATION AND ANALYSIS:

Machine learning issues benefit from exploratory data analysis because it increases confidence in future outcomes' validity, correct interpretation, and applicability to target business settings [4]. Validating and checking for anomalies in raw data is essential for achieving this degree of assurance, since it ensures that the data set was obtained accurately. Additionally, EDA aids in discovering discoveries that academics and business stakeholders missed or didn't think were worth pursuing. Both the Univariate Visualization and the Pair-wise Correlation Matrix were used to do EDA. The former offers summary statistics for each field in the raw data set (figure 2), while the latter helps to understand the interactions between distinct fields (figure 3).

Table 2. Irrelevant Features

Feature	% of Null values
Sunshine	43%
Evaporation	48%
Cloud3pm	40%
Cloud9am	38%

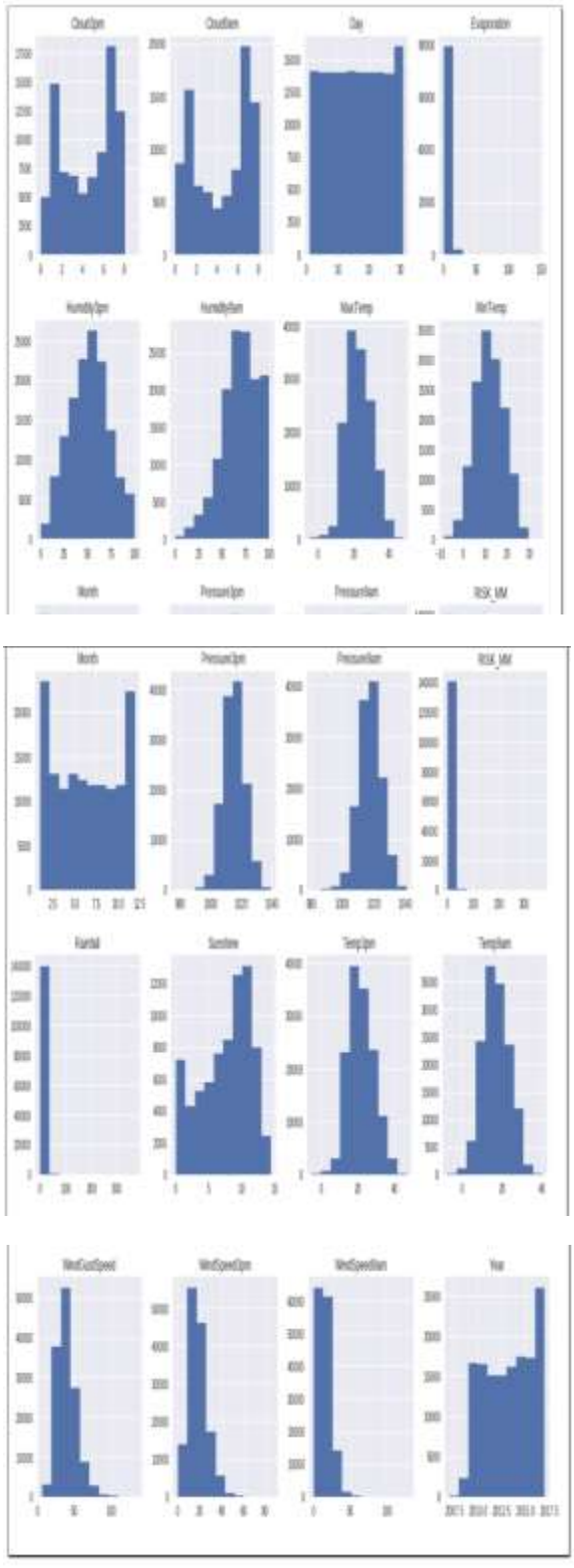


Fig. 2. Univariate Visualization.

We have other features with null values too which we will be imputing in our preprocessing steps. If we look at the distribution of our target variable, it is clear that we have a class imbalance problem with the number of positive instances - 110316 and number of negative instances – 31877.

PREDICTING RAINFALL USING MACHINE LEARNING TECHNIQUES:

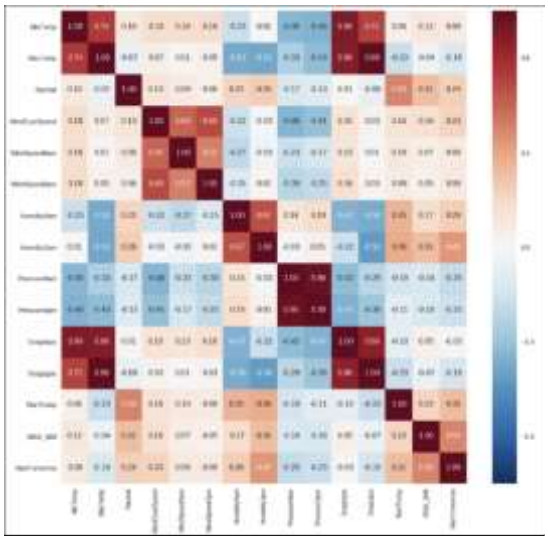


Fig.3. HeatMap The correlation matrix depicts that the features - Maxterms, Pressure9am, Pressure3pm,Temp3pmandTemp9amarenegativelycorrelatedwithtargetvariable.Hence,wecandropthisfeatureinourfeatureselectionsteplater.

DATAPREPROCESSING

Data preprocessing is a data mining technique that involves transforming rawdata into an understandable format. Real-world data is often incomplete, icon-sestet, and/or lacking in certain behaviors or trends, and is likely to containmanyerrors.Wehaveconductedthepreprocessingsteps.

Missing Values: As per our EDA step, we learned that we have few instanceswith null values. Hence, this becomes one of the important steps. To impute themissing values, we will group our instances based on the location and date andtherebyreplacethenullvaluesbytheirrespectivemeanvalues.

FeatureExpansion: Datefeaturecanbe expanded to Day, Month and Yearand then these newly created features can be further used for other preprocessingsteps.

Categorical Values: Categorical feature is one that has two or more categories,but there is no intrinsic ordering to the categories. We have a few categoricalfeatures-Industry,WindDir9am,WindDir3pmwith16uniquevalues.Nowit gets complicated for machines to understand texts and process them, ratherthan numbers, since the models are based on mathematical equations and call-collations. Therefore, we have to encode the categorical data. We tried two different techniques here.

DummyVariables:ADummyvariableisanartificialvariablecreatedtorepresent an attribute with two or more distinct categories/levels [5]. How-ever, as we have 16 unique values, our one feature will now get transformedto16newfeatureswhichinturnresultsin**a curseofdimensionality**.Foreachinstance,wewillhaveafeaturewith1valueandthe rest15featureswith0values.

Example: Categorical Encoding of feature - **windDir3pm** using DummyVariables

MinTemp	Rainfall	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm
13.4	0.6	W	44.0	W	WNW
7.4	0.0	WNW	44.0	NNW	WSW

Fig. 4.SampleInstance.

(WindDir3pm, S)	(WindDir3pm, SE)	(WindDir3pm, SSE)	(WindDir3pm, SSW)	(WindDir3pm, SW)	(WindDir3pm, W)	(WindDir3pm, WNW)	(WindDir3pm, WSW)
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1

Fig.5.DummyVariables.

FeatureHashing:

Featurehashingschemeisanotherusefulfeatureend-gingering scheme for dealing with large scale categorical features. In thisscheme, a hash function is typically used with the number of encoded fee-tires pre-set (as a vector of pre-defined length) such that the hashed valuesof the features are used as indices in this pre-defined vector and values areupdatedaccordingly[6].

Example:

CategoricalEncodingoffeature**windDir3pm**using FeatureHashing

MinTemp	Rainfall	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm
13.4	0.6	W	44.0	W	WNW
7.4	0.0	WNW	44.0	NNW	WSW

Fig.6.SampleInstance.

WindDir9am_0	WindDir9am_1	WindDir9am_2	WindDir9am_3	WindDir3pm_0
-1.0	0.0	0.0	0.0	-2.0
-1.0	0.0	0.0	-2.0	-3.0

Fig.7.FeatureHashing.

FEATURESCALING:

Features with widely varied magni-tides and ranges are included in our dataset. This is a problem, however, since the vast majority of ML algorithms base their calculations on the Eu-clean distance between any two pieces of data. The distance computations will give far more weight to features with large magnitudes than to features with smaller magnitudes. Raising the magnitudes of all characteristics to the same level will mitigate this impact. Scaling allows for this to be accomplished. The features were brought into the range of 0 to 1 using Scikit-Lear's min-max scalar [7].

...	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	RainToday	RainTomorrow
..	44.0	20.0	24.0	71.0	22.0	0.0	0
..	44.0	4.0	22.0	44.0	25.0	0.0	0

Fig.8.SampleInstancebeforescaling.

.	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	RainToday	RainTomorrow
_	0.289652	0.291765	0.259824	0.71	0.22	0.0	0.0
_	0.289652	0.029529	0.235294	0.44	0.25	0.0	0.0

Fig.9.SampleInstanceafterscaling.

FEATURE SELECTION

Selecting the attributes that have the most impact on our prediction variable or output may be done either automatically or manually; this process is called feature selection. When data contains irrelevant characteristics, the accuracy of the models might be negatively affected, leading to the model learning based on these features. The selection of features aids in reducing over fitting, improving accuracy, and decreasing training time. We achieved the same results by utilizing two techniques to complete this activity.

Choose without variation:

In order to determine which attributes are most strongly related to the output variable, statistical tests may be used. In order to choose a fixed number of features, the scikit-learn package offers the Select Best class, which may be used with a suite of various statistical tests. For this dataset, we utilized a chi-squared test to identify the five most useful features[8][9].

CORRELATIONMATRIXWITH HEATMAP:

The characteristics' relationships to one another and the goal variable are described via correlation. A positive correlation would indicate that the target variable's value grows as a function of the feature's value, while a negative correlation would indicate that the target variable's value falls as a function of the feature's value. We used the seaborn library to create a heatmap of associated characteristics (figure 3), which makes it simple to see which attributes are most connected to the target variable (figure 3).

HANDLING CLASS IMBALANCE

During the EDA phase, we discovered that our data set is very skewed. Due to our model's inability to learn as much about the minority class, biased findings are produced by imbalanced data. Our two tests were conducted using oversampled and under sampled data sets, respectively.

Sample Analysis:

To get rid of cases of the majority class, we utilized the Ambler random under sampler library [10]. To ensure minimal data loss, this removal is based on dis-trances (figure 10).

OVERSAMPLING:

We used Ambler's SMOTE technique to generate syn-thetic instances for minority class [10]. A subset of data is taken from theminority class as an example and then new synthetic similar instances arecreated.(Figure11

MODELS:

From several model families, we selected classifiers such as Linear classifier, Tree-based,

Distance-based, Rule-based, and Ensemble. Every one of

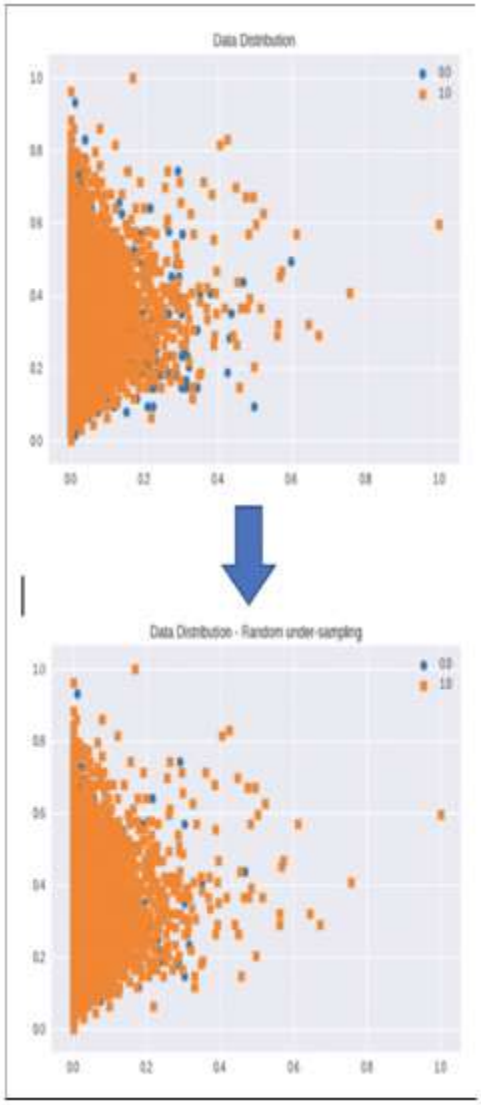


Fig.10. Under sampling.

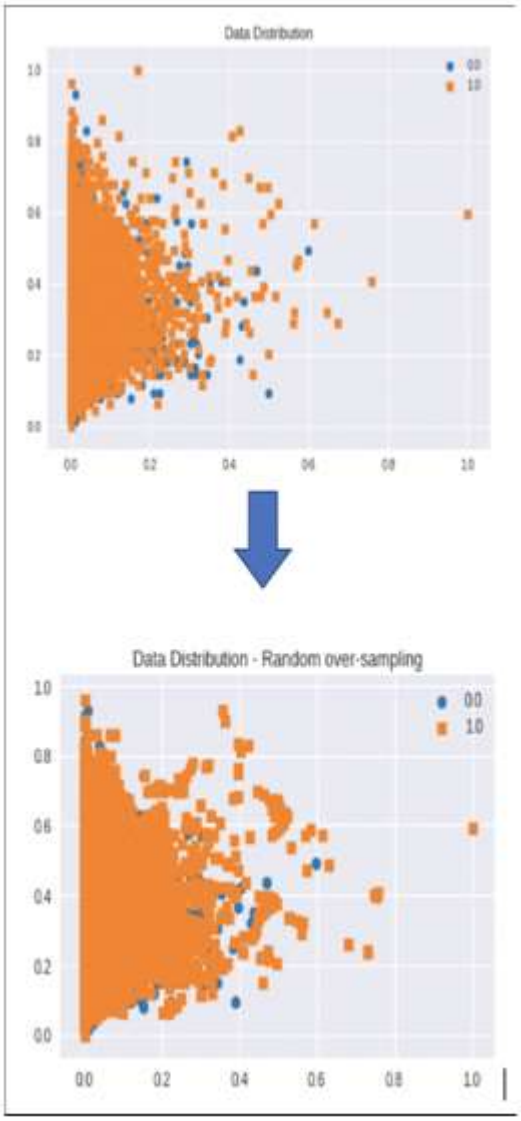


Fig.11.Oversampling.

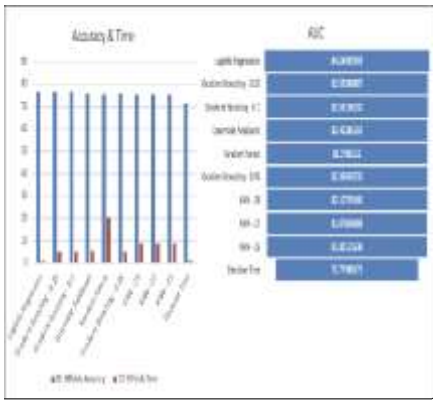
With the exception of the decision table, which was built using weak, all classifiers were developed using scikit-learn.The prediction model stop perform has been constructed using the following categorization algorithms:

EXPERIMENTS:

One classification approach that uses a number of independent variables to predict a binary result (1/0, Yes/No, True/False) is Logistic Regression. We utilize dummy variables to express binary or categorical outcomes. If the outcome variable is categorical and the dependent variable is the logarithm of the chances, then logistic regression is just a subset of linear regression. Fitting data to a logit function allows it to forecast the likelihood of an event occurring, in simple terms. Our issue is a binaryclassification, hence Logistic

Regression is the best match. Decision trees are well-suited for use in programming structures due to their inherent if-then-else nature. Problems involving the systematic checking of traits or features to establish a final category are also well-suited to them. The input and output variables may be either continuous or categorical, and it still works for both. Using the most important divergence in the input variables, this method divides the population or sample into two or more similar groups. Because our goal variable is a binary categorical variable, Decision Tree is well-suited to our situation because of its feature.

K-Nearest Neighbor is a technique for sluggish, non-parametric learning. A non-parametric statistic does not presume any particular distribution for the underlying data. Put simply, the dataset dictates the model's structure. In order to generate a model, lazy al-growth does not need any training data points. The testing phase used all training data. KNN is more effective when using fewer features rather than more features. As the number of characteristics grows, so does the amount of data needed. Overfitting is another issue that arises as the number of dimensions increases. Nevertheless, we have included feature selection to decrease dimensions, making KNN a promising contender for our issue. The setup of our model: After experimenting with n numbers between 3 and 30, we found that 25, 27, and 29 yielded the best results for the model. One convenient and condensed approach to describe complicated business logic is with a decision table. To better express the many parts that make up business logic, a decision table neatly divides them into rules, circumstances, and actions (decisions). the eleventh Weka was used to accomplish this. Random Forest is a method for guided ensemble learning. Ensemble learning involves combining several weak learners into a



single robust prediction. The Forest is a collection of decision trees that we have here.
To classify a new object based on attributes, each tree gives a classification, andwe say the tree votes on the subject. Out of all the trees in the forest, the one with the most votes is the one that the forest will pick as its categorization. Our model is set up with 100 weak learners and 4 maximum treedepths.AdaBoost uses a series of under-performing learners trained on various weighted datasets. It begins by making predictions based on the initial data set and assigns equal importance to each observation. It provides more weight to the inaccurately anticipated data if the first learner makes a poor prediction. Iterative processes always involve adding more learners until either accuracy or the number of models reaches a limit.
Here is our model set up: 50 weak learners. Boosting using Gradients Train a large number of models in a sequential fashion here. With the use of the Gradient Descent approach, each successive model progressively reduces the system-wide loss function (y = ax + b + e), where he is the error term. An improved estimate of the response variable is obtained by the learning approach via the sequential fitting of new models. The fundamental premise of this technique is to build new base learners that can be optimally correlated with the ensemble-relevant negative gradient of the loss function. Here is the setup of our model: 100 weak learners, learning rate range [0.05, 0.1, 0.25], maximum features = 2, maximum depth = 2.

EVALUATION

We used the following assessment measures [12] to assess the performance of our classifiers. When comparing the amount of input samples to the number of right predictions, we get the accuracy. If the quantity of samples from each class is equal, then it will work. Other measures will also be considered due to the fact that our data is skewed. Binomial classification problems make use of Area Under the Curve (AUC). A classifier's area under the curve (AUC) is the likelihood that it would assign a higher ranking to a positive example than a negative one, given two examples that are both randomly selected.

A classifier's accuracy may be measured by dividing the number of anticipated positive outcomes by the number of correct positive results. This ratio is called precision.

Calculating recall involves dividing the total number of relevant samples (all samples that should have been classified as positive) by the number of accurate positive findings.
The F1 score is calculated by summing

accuracy and memory. A value between zero and one is the F1 Score. It reveals our classifier's

precision (the number of cases it properly classifies) and robustness (the number of instances it does not miss). Extreme accuracy is achieved with high precision but poor recall, yet many difficult-to-classify occurrences are missed. The higher the F1 Score, the more effective our model is.

We get a matrix out of Confusion Matrix, which details the model's overall performance. This section primarily addresses the following: True Positives, where the actual output was also a yes; True Negatives, where the actual output was a no; False Positives, where the actual output was a yes; and False Negatives, where the actual output was a no.

Hierarchical k-fold We used a stratified k-fold strategy to train our models since our data is unbalanced. This method divides the data into k-folds with an equal number of positive and negative values. There would be less bias and better reliability with these measurements. We set k equal to 10.
We used paired t testing among the top three classifiers as a statistical tool to compare their performance.

EXPERIMENTSANDRESULTS

For all the experiments and development of classifiers, we used Python 3 andGoogle cola'sJupiter Notebook. We used libraries such as Skit Learn, Mat-plotid, Seaborn, Pandas, NumPy and Ambler. We used weak for implementingDecisionTable.
Wecarriedexperimentswithdifferentinputdata;one withtheoriginaldataset, then with the under sampled dataset and the last one with the oversaw-pled dataset. We splatted out dataset in ratio of 75:25 for training and testingpurpose.

EXPERIMENT1 ORIGINAL DATASET:

Post all the preprocessing steps (as men-toned above in the Methodology section), we ran all the implemented classifierseach one with the same input data (Shape: 92037 x 4). Figure 12 depicts twoconsidered metrics (10-skfold Accuracy and Area Under Curve) for all the class-sifters.Accuracy wise Gradient Boosting with a learning rate of 0.25 performed best,coveragewiseRandomForestandDecisionTreeperformedworst.

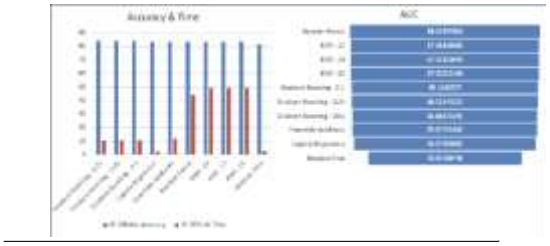


Fig.12.Experiment1.

Experiment2-Under sampled Dataset: Post all the preprocessing steps(as mentioned above in the Methodology section) including the under-samplingstep, we ran all the implemented classifiers each one with the same input data(Shape: 54274 x 4). Figure 13 depicts two considered metrics (10-skfold AccuracyandAreaUnderCurve)forallttheclassifiers.

CONCLUSIONANDFUTUREWORK

In this study, we investigated and used several preprocessing methods to understand how they affected our classifiers' overall performance. We also looked at how the input data affected the model predictions by comparing all the classifiers with various sets of data. There is no association between rainfall and the appropriate location and time in Australia, and the weather is quite unpredictable. Key characteristics were determined with the aid of patterns and linkages we discovered in the data. Look in the part that follows. We can use Deep Learning models like Multilayer Perceptron and Convolutional Neural Networks since our data is massive. The two types of models, deep learning and machine learning classifiers, should be compared.

REFERENCES

1. World Health Organization: Climate Change and Human Health: Risks and Responses. World Health Organization, January 2003
2. Alcantara-Ayala, I.: Geomorphology, natural hazards, vulnerability, and prevention of natural disasters in developing countries. Geomorphology 47(24), 107124 (2002)
3. Nicholls, N.: Atmospheric and climatic hazards: Improved monitoring and prediction for disaster mitigation. Natural Hazards 23(23), 137155 (2001)
4. [Online] Inda talabs, Exploratory Data Analysis: the best way to Start a Data Science Project. Available: https://medium.com/@InDataLabs/ why-start-a-data-science-project-with-exploratory-data-analysis-f90c0efcbe49
5. [Online] Pandas Documentation. Available: https://pandas.pydata.org/ pandas-docs/stable/reference/API/pandas. Get_dummies.html
6. [Online] Skit-Learn Documentation Available: https://scikit-

learn.org/ stable/modules/generated/sklearn.feature_extraction.FeatureHasher. html
7. [Online] Skit-Learn Documentation Available: https://scikit-learn.org/ stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
8. [Online] Skit Learn Documentation Available: https://scikit-learn.org/ stable/modules/generated/sklearn.feature_selection.SelectKBest.html Predicting Rainfall using Machine Learning Techniques 17
9. [Online] Raheel Shaikh, Feature Selection Techniques in Machine Learning with Python Available: https://towardsdatascience.com/ feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e
10. [Online] Imbalanced-learn Documentation Available: https://imbalanced-learn. readthedocs.io/end/stable/introduction.html
11. V. Varalakshmi and D. Ramya Chitra, Ripple Down Rule learner (RIDOR) Classsifter for IRIS Dataset. Issues, vol 1, p. 79-85.
12. [Online] Aditya Mishra, Metrics to Evaluate your Machine Learning Algorithm Available: https://towardsdatascience.c