Next Day Rain Prediction in Australia

Hema Sri Kambhampati
Maters in Computer Science
University of Ottawa, Ottawa,
Canada
hkamb097@uottawa.ca

***Abstract*— Traditionally, weather forecasting has been done by using physical model of atmosphere to calculate perturbations. But this approach has provided unstable perturbations which lead to inaccurate predictions of atmosphere. This paper will delve into an application to accomplish accurate weather forecast for large period as machine learning approaches are more vigorous to perturbations. This paper elaborates how a machine Learning model was constructed from various aspects including data preprocessing, normalization, sampling, feature selection and finally evaluating machine learning algorithms like Linear model (Naive Bayes), Distance base model(KNN), Rule based model(scope rules, dummy), Tree based model(Decision Tree), Ensemble learning(Random Forest). Additionally, this paper addresses the binary classification problem to predict whether its is going to rain tomorrow in Australia based on the weather data collected from metrological department of Australia for 10 years.**

Keywords—Machine Learning model, Naive Bayes, KNN, scope rules, Dummy, Decision Tree, Random Forest, Binary classification.

I. INTRODUCTION

Humans are trying to predict climate based on prehistoric weather. There are good reasons to know when to plant crops, when to farm, and when to prepare for a drought and flood in a country like Australia. Since agriculture depends on rainfall, floods caused by heavy rains often lead to large crop losses. Rainfall is a very complex phenomenon, which depends on various climatic, oceanic and geographic parameters. The relationship between these parameters and rainfall is unstable. This fact coupled with changing behavior will reduce the availability of existing weather forecasts to customers.

Australia Bureau of Meteorology for the period November 2007 to June 2017. It is mainly focused on the development of models for rainfall prediction in various cities of Australia (Weather Stations). Rainfall prediction is very important for the Australia economy and day to day life. When scarcity or heavy weather changes, rainfall greatly affects rural and urban life. In this paper humidity, temperatures & wind parameters are used as predictors. Few classification algorithms have been used for model building, training and prediction. Those are Naive Bayes Classification, Rule based classification, Decision Tree Classification, KNN classification and Random Forest Classification. The developed and trained model can predict rainfall in advance for next day of a given area. Accuracy percentage varies from algorithm to algorithm.

This paper is an effort to predict rainfall within Australia. It takes the strategy of applying machine learning models to weather data gathered in Australia. As part of this work, this application was written using python and scikit-learn to demonstrate rainfall prediction using multiple machine learning models.

II. DATASET AND FEATURES

The Dataset was obtained from Kaggle, which was reduced from weather dataset from metrological department of Australia which consists of 142,193 daily weather observations from 49 weather stations across Australia over the period November 2007 to June 2017. The following are 24 original variables in this dataset.

Basic Features:

1. **Date**-The date of observation,
2. **Location**-The common name of the location of the weather station,
3. **MinTemp**-The minimum temperature in degrees Celsius,
4. **MaxTemp**-The maximum temperature in degrees Celsius,
5. **Rainfall**-The amount of rainfall recorded for the day in mm,
6. **Evaporation**-The so-called Class A pan evaporation (mm) in the 24 hours to 9am,
7. **Sunshine**-The number of hours of bright sunshine in the day,
8. **WindGustDir**-The direction of the strongest wind gust in the 24 hours to midnight,
9. **WindGustSpeed**-The speed (km/h) of the strongest wind gust in the 24 hours to midnight,
10. **WindDir9am**-Direction of the wind at 9am,
11. **WindDir3pm**-Direction of the wind at 3pm,
12. **WindSpeed9am**-Wind speed (km/hr) averaged over 10 minutes prior to 9am,
13. **WindSpeed3pm**-Wind speed (km/hr) averaged over 10 minutes prior to 3pm,
14. **Humidity9am**-Humidity (percent) at 9am,
15. **Humidity3pm**-Humidity (percent) at 3pm,
16. **Pressure9am**-Atmospheric pressure (hpa) reduced to mean sea level at 9am,
17. **Pressure3pm**-Atmospheric pressure (hpa) reduced to mean sea level at 3pm',
18. **Cloud9am**-Fraction of sky obscured by cloud at 9am.
19. **Cloud3pm**-Fraction of sky obscured by cloud
20. **Temp9am**-Temperature (degrees C) at 9am,
21. **Temp3pm**-Temperature (degrees C) at 3pm,
22. **RainToday**-Boolean: 1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0,
23. ***RISK_MM***-The amount of rain. A kind of measure of the "risk",
24. **RainTomorrow**-The target variable. Did it rain tomorrow?

III. DATA ANALYSIS

Data analysis is used to find valuable data and data trends. It deals mainly with descriptive or inferential statistical-probability distributions. Data analysis involves cleaning, changing, exploring and modeling your data to determine and support your data. Important uses of data analysis include hypothesis testing and machine learning [1].

While data analysis helps you in providing probability of your hypothesis to happen, machine learning is a method of data analysis used to design a model to learn the trends, findings and dependence between attributes and target variable without programming explicitly [1].

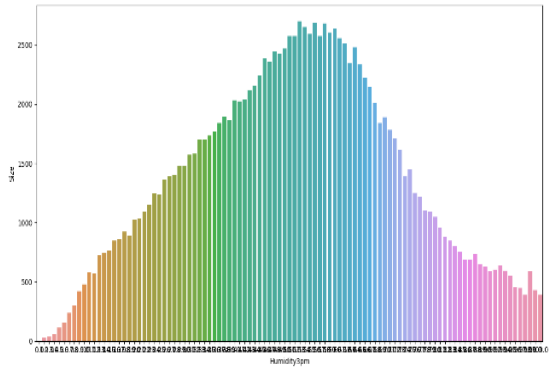

Figure 1 Visualization of data in feature Humidity at 3pm

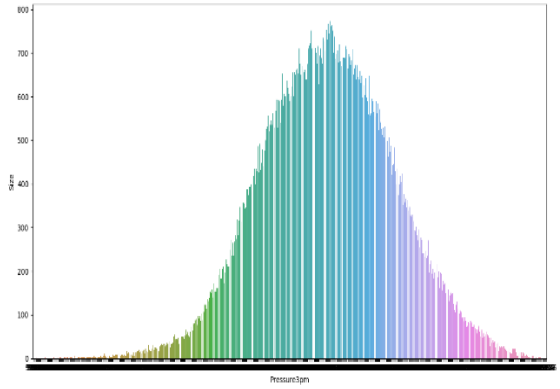

Figure 2 Visualization of data in feature pressure at 3pm

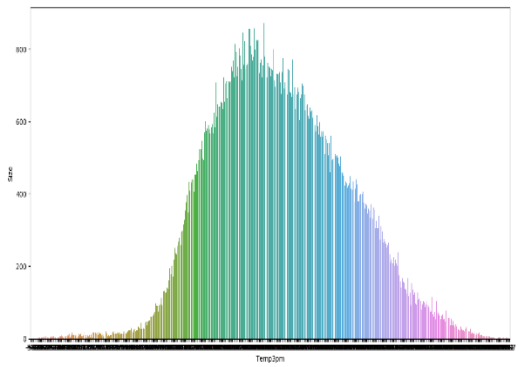

Figure 3 Visualization of data in feature Temperature at 3pm

A. Scan for missing values

The data set is almost complete, which introduces various biases in your analysis. One of the simplest approaches is to just delete the data points with missing values but deletion will bias the outcome of the model. Another approach is imputation which will replace the missing values with the mean /median of the respective feature's data. Generally missing values in the dataset are null, Nan, or NA value. In figure 1 The dataset mentioned in the paper has Na values in Windspeed, Humidity and Pressure at 9am and 3pm.Additionally it had NaN values in feature like WindGustDir, WindDir at 9am and 3pm.Missing values in each feature with numerical data are replaced with the mean of the values in each location.

For example, consider the feature WindSpeed9am in *figure 4* has categorical data so NA values so the whole row was deleted to delete the missing values in the dataset.

In *figure 4* "WindSpeed" had numerical data with missing values "NA" so they are replaced with mean of the values of "WindSpeed" in the specific city in Australia. Missing values like 'NaN' in Features with categorical data are replaced with '0'. This approach of replacing the missing values reduced the bias in the dataset as the values in each feature vary according to the location(city).

G	H	I	J	K	L
Sunshine	WindGustI	WindGustS	WindDir9a	WindDir3p	WindSpeed
NA	W	44	W	WNW	20
NA	WNW	44	NNW	WSW	4
NA	WSW	46	W	WSW	19
NA	NE	24	SE	E	11
NA	W	41	ENE	NW	7
NA	WNW	56	W	W	19
NA	W	50	SW	W	20
NA	W	35	SSE	W	6
NA	NNW	80	SE	NW	7
NA	W	28	S	SSE	15
NA	N	30	SSE	ESE	17
NA	NNE	31	NE	ENE	15
NA	W	61	NNW	NNW	28
NA	SW	44	W	SSW	24
NA	WNW	50	NA	WNW	NA
NA	ENE	22	SSW	E	11

Figure 4 Missing values in the weather Dataset

B. Class label Distribution

Data imbalance usually reflects an unequal distribution of classes within a dataset.The output label "*RainTomorrow*" of dataset was inequal distribution of minority and majority classes

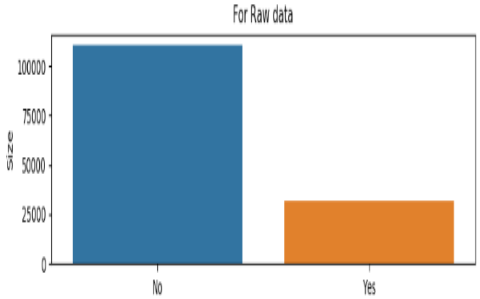

Figure 5 Class Label Distribution for Raw Data

In the above *figure 5* the output label "*Rain Tomorrow*" was unequally distributed where majority class with label "NO" had a greater number of values compared to minority class with label "Yes".

C. Finding Skew and kurtosis

Skewness

The normality test is used to determine whether the data set is designed to be a normal distribution. The distribution of many statistical functions must be common or close to normal. There are graphical and statistical methods for estimating generality. The skewness is a measure of the odds of the probability distribution of a random variable with respect to its mean. In other words, the distortion tells you the sum and direction of the curve (the deviation from the horizontal symmetry). Curved values can be positive or negative: if stored in a CSV format. This is a huge advantage because it greatly reduces the time needed to crawl and clean up data [2].

If the skewness is less than -1 or greater than 1, the distribution is highly skewed. If the distortion is between -1 and -0.5 or between 0.5 and 1, the distribution is moderately skewed. If the distortion is between 00.5 and 0.5, the distribution is approximately symmetric.

```
Skewness
MinTemp          0.062522
MaxTemp          0.319188
Rainfall         9.726668
WindGustSpeed    0.881228
WindSpeed9am     0.747503
WindSpeed3pm     0.647625
Humidity9am     -0.474635
Humidity3pm      0.017224
Pressure9am     -0.099957
Pressure3pm     -0.049693
Temp9am          0.158011
Temp3pm          0.339794
RainToday        1.344466
RainTomorrow     1.359111
dtype: float64
```

Figure 6: Skewness in Dataset

Kurtosis

Kurtosis is definitely the tail of the distribution, not kurtosis or flatness. It is used to describe the extreme value of one tail relative to another. This is a measure of outliers in distribution.

Most kurtosis in the data set indicates a large tail or outlier in the data. If kurtosis is high, we should investigate why there are so many outliers. It shows a lot of things, could be wrong data entry or anything else.

Low kurtosis in the data set indicates that the tail of the data is shallow or that there are no outliers. If our kurtosis is low (great, unreliable), then we also need to research and trim a bad result dataset. [2].

```
kurtosis
MinTemp          -0.550994
MaxTemp          -0.499746
Rainfall         165.123383
WindGustSpeed     1.419357
WindSpeed9am      0.987627
WindSpeed3pm      0.760902
Humidity9am      -0.032211
Humidity3pm      -0.520247
Pressure9am       0.247189
Pressure3pm       0.152611
Temp9am          -0.510324
Temp3pm          -0.414468
RainToday        -0.192414
RainTomorrow     -0.152821
dtype: float64
```

Figure 7 Kurtosis in Dataset

D. Correlation Analysis

Correlation analysis is a widely used technique for identifying interesting relationships in data. These relationships help us to identify the characteristics of the target category. Correlation analysis is a widely used statistical metric through which various studies can effectively identify interesting collinear relationships between different features of the dataset. This article uses correlation analysis to identify the symptoms that can severely affect the severity of depression and the emotional state [3].

Correlation matrix:

It is obtained by plotting all the features against each other. I have plotted heat map for this for all the features and analyzed to see the relation between each of them. We even obtained correlation of each feature to the predicted value [3].

Heatmap:

Each attribute (variable) is listed on two axes and their relationship to other variables is shown in color. When the color is dark in both directions (red or blue), this means that the correlation between these variables is high and therefore should not be used in the same model in pairs. Those who want to use this housing data should consider variables that are highly correlated with prices, but less correlated with each other. (This data is compressed to show variables that are highly correlated with each other.) [4]

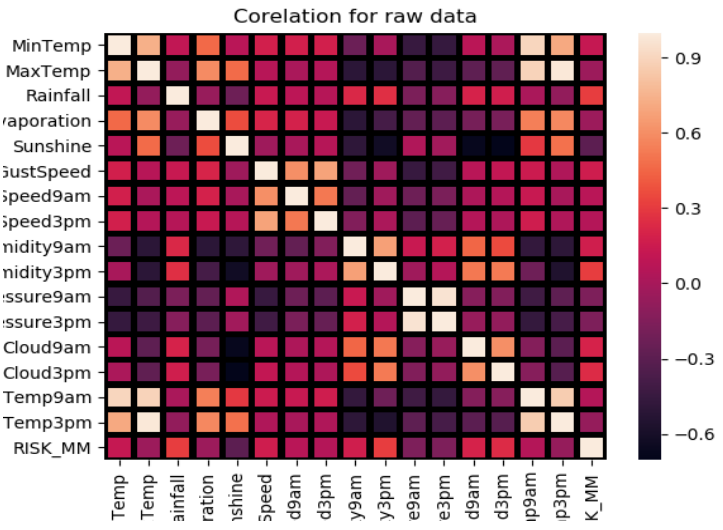

Figure 8 Correlation Matrix for Raw Data (Heatmap)

From figure 8 shows that the features “Max_temp” and the “Temp3pm” are positively corelated. Others features the figure *8* also depicts that there is no high correlation between the features and there is no high correlation with the predicted variable. So, this makes that we cannot remove any features and each feature is important for building the model [5].

IV. DATA PREPROCESSING

Data preprocessing is the process of preparing raw data and making it suitable for machine learning models. This is the first and crucial step in creating a machine learning model. When creating a machine learning project, you can’t always look at clean and formatted data. Also, when performing any operation on data, it must be cleaned and formatted. So, for this purpose I am using data preprocessing work [6].

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model [6].

A. One-hot encoding

One-hot encoding takes properties with numeric values and encodes these values into binary array. The length of these arrays is the maximum in numeric categories. The result of the "one hot encode" attribute is n binary attributes that represent the values in the original attribute. This allows machine learning algorithms to take advantage of information in category values without being confused by convention [7].

Machine learning algorithms provided by libraries such as scikit learn have difficulty processing properties with string values. To fix this problem, we perform a pre-processing step that converts string valued attributes into numerical representations. Scikit learn has a LabelEncoder that does just that: turns string values in an attribute into numerical values [7]. After running the data above through the scikit learn LabelEncoder, our table will look something like this:

==========Dataset After onehot encoding ==========

	MinTemp	MaxTemp	Rainfall	WindGustSpeed	...	x2_SW	x2_W	x2_WNW	x2_WSW
0	13.4	22.9	0.6	44.0	...	0.0	0.0	1.0	0.0
1	7.4	25.1	0.0	44.0	...	0.0	0.0	0.0	1.0
2	12.9	25.7	0.0	46.0	...	0.0	0.0	0.0	1.0
3	9.2	28.0	0.0	24.0	...	0.0	0.0	0.0	0.0
4	17.5	32.3	1.0	41.0	...	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
142188	3.5	21.8	0.0	31.0	...	0.0	0.0	0.0	0.0
142189	2.8	23.4	0.0	31.0	...	0.0	0.0	0.0	0.0
142190	3.6	25.3	0.0	22.0	...	0.0	0.0	0.0	0.0
142191	5.4	26.9	0.0	37.0	...	0.0	0.0	1.0	0.0
142192	7.8	27.0	0.0	28.0	...	0.0	0.0	0.0	0.0

Figure 9 Dataset after One hot Encoding

Figure 9 illustrates how the categorical features are converted to numerical features. Feature like “WindGustDir” is replaced with the new columns x2_SW, x2_W, x2_WNW, x2_WSW.Similarly all the Categorical featured are converted into numeric features by adding new columns to the dataset for each unique value for each categorical feature.

B. Normalization

Many machine learning algorithms attempt to find data trends by comparing the characteristics of the data. However, when problems arise, the functionality is very different. When data appears to be compressed, we know there is a problem the data looks squished like that, we know we have a problem [8].

When normalization is performed, the size of the value is scaled to a fairly small value. This is important for many neural networks and k nearest neighbor algorithms.

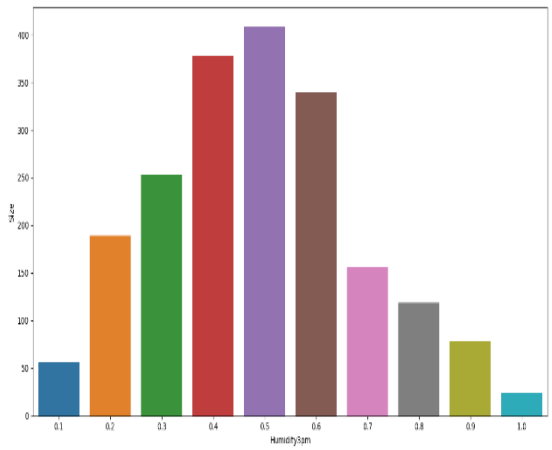

Figure 10 Visualization of normalized data for feature Humidity at 3pm

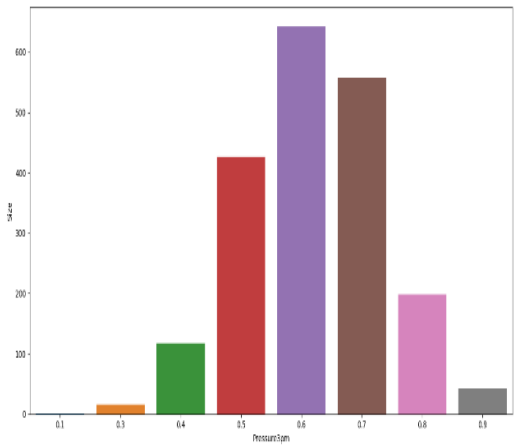

Figure 11 Visualization of normalized data for feature Pressure at 3pm

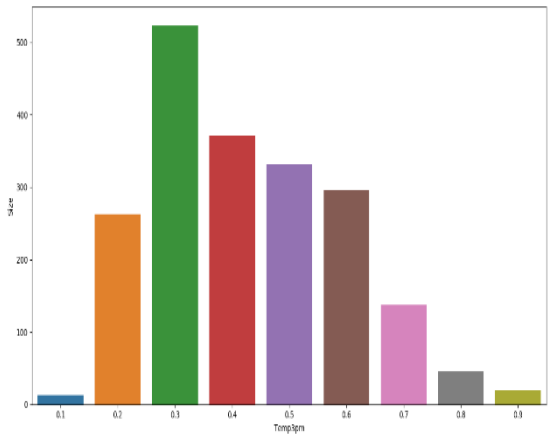

Figure 12 Visualization of normalized data for feature Temperature at 3pm

Min-Max Normalization:

Min-max normalization is one of the most common methods for normalizing data. For each attribute, the minimum value of the attribute is converted to 0, the maximum value is converted to 1, and each other value is converted between 0 and 1 as a decimal. [8].

$$\frac{value - min}{max - min}$$

MinMaxScaler() function of Scikit learn was used in the project . This automatically takes the min and max value for the values of the feature in the data set and performs the normalisation.

==========Dataset After normalization==========

	MinTemp	MaxTemp	Rainfall	WindGustSpeed	...	x2_SW	x2_W	x2_WNW	x2_WSW
0	0.513064	0.446154	0.001632	0.294574	...	0.0	0.0	1.0	0.0
1	0.370546	0.494505	0.000000	0.294574	...	0.0	0.0	0.0	1.0
2	0.501188	0.507692	0.000000	0.310078	...	0.0	0.0	0.0	1.0
3	0.413302	0.558242	0.000000	0.139535	...	0.0	0.0	0.0	0.0
4	0.610451	0.652747	0.002720	0.271318	...	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
1995	0.546318	0.367033	0.009249	0.325581	...	0.0	0.0	0.0	1.0
1996	0.368171	0.340659	0.008705	0.271318	...	0.0	0.0	0.0	0.0
1997	0.263658	0.371429	0.000544	0.124031	...	0.0	0.0	0.0	0.0
1998	0.277910	0.367033	0.000544	0.255814	...	0.0	0.0	1.0	0.0
1999	0.315914	0.389011	0.000000	0.170543	...	0.0	0.0	0.0	0.0

Figure 13 Dataset after normalization

Figure 14 explains how the numerical values are normalized to range 0,1. For example “WindGustSpeed” the original value at row 0 in figure 9 had value 44.0 which was normalized to 0.294574 in above figure.

C. Dimensionality Reduction

In machine learning problems, each training scenario usually consists of ten thousand tasks. This can be a problem because it makes our training too slow and can be overused. This problem is often called the curse of dimensionality [9].

Due to the problems associated with dimensional curse, it is necessary to drastically reduce the number of features / dimensions to help improve the performance of the model and find the best solution for our machine learning model. Fortunately, in most real-life problems, we can usually reduce the size of our training set without losing much difference in our data [10].

In the project the features “Evaporation", "Sunshine", "Cloud3pm", "Cloud9am” were deleted as they have all “NA” values in dataset. The feature “RISK_MM” was also removed from dataset as it might leak answers to the model at time of prediction and make our model less efficient for rain prediction.

D. Cross Validation

Cross-validation is a statistical method for assessing the skills of machine learning models. It is commonly used in applied machine learning to compare and select models for a given predictive modeling problem because it is easy to understand, easy to implement, and resultant estimations are less biased than other methods [11].

Cross validation is mainly used in applied machine learning to assess the skills of machine learning models based on invisible data. That is, a limited model is used to predict how the model will normally perform when used to make predictions on unused data during model training. This is a popular method because it is easy to understand and leads to less bias or optimism about model skills compared to other methods (such as general training / test partitions). [11].

In similar way the project used *K- fold Cross Validation* and *Test and Train split* which will be explained further in the paper.

E. Sampling

In many real-world classification tasks, such as churn prediction and fraud detection, we often encounter class imbalance problems, which means that one class number is significantly higher than another. Class imbalance problems pose a huge challenge to standard classification learning algorithms. They incorrectly classify a small number of instances in most cases in an unbalanced data set. To solve the problem of learning from unbalanced data sets, one of the many solutions is the "resampling" method, which attempts to solve the problem by reverting the data and serves as the preprocessing step. Some re-amp design techniques [12].

Oversampling:

Oversampling can be defined as adding more copies of certain classes. If you cannot handle large amounts of data, oversampling may be a good option. I use Scikit-Learn's Resampling Module to randomly copy samples from minority groups [12].

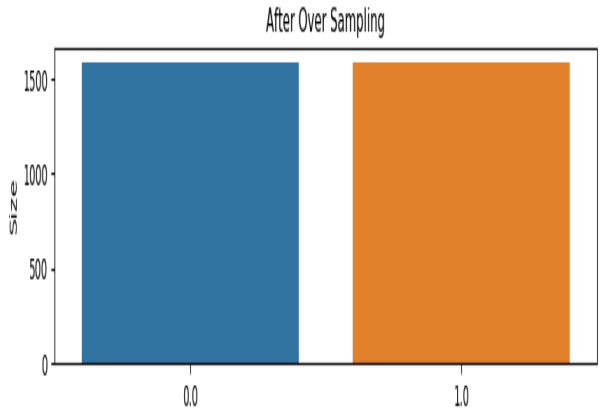

Figure 14 Class Label distribution after oversampling

Figure 14 illustrates the labels in majority class "No" are replaced with nearby minority class labels "No".

Under sampling:

Under sampling removes some of the considerations from most categories. When you have a lot of data, under sampling can be a good option — consider millions of rows. The disadvantage is that we are deleting valuable information. This leads to a low and generalizability of the test set. I use Skikit-Learn's Resampling Module to randomly copy samples from minority groups [12].

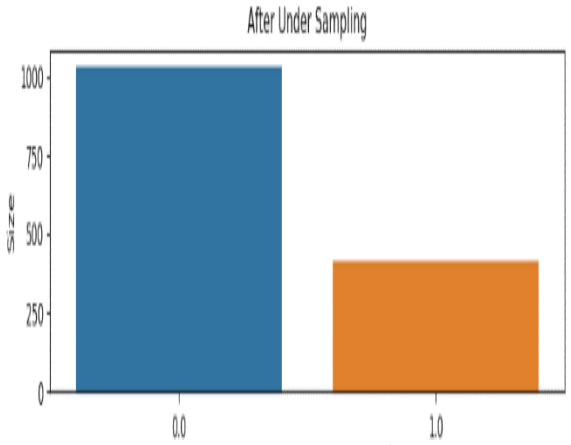

Figure 15 Class Label Distribution after Under Sampling

Figure 15 illustrates the labels in majority class "No" are reduced without changing any labels in minority class labels "No".

Balance sampling:

It is the hybrid sampling technique which is the combination of under sampling and over sampling.

Figure 16 illustrates the oversampling and under sampling are performed simultaneously on minority class "Yes" and majority class "yes" respectively.

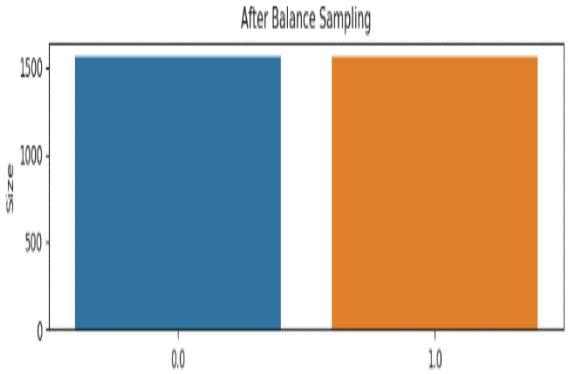

Figure 16 Class label distribution after balance sampling

F. Feature Selection

Feature selection refers to the process of obtaining a subset from the original feature set according to some feature selection criteria. The feature selection criteria select the relevant attributes of the data set. It has been instrumental in reducing the level of data processing, eliminating unnecessary and irrelevant features. Feature selection can pre-process the technology learning algorithm. Good feature selection results improve learning accuracy, reduce learning time and facilitate learning outcomes. [13].

The performance of the feature selection method is usually evaluated by the machine learning model. The commonly used machine learning models includes Naive Bayes, KNN, Decision Tree, Random forest and dummy classifier. A good feature selection method should have high learning accuracy but less computational overhead (time complexity and space complexity). Although there have been solid reviews on feature selection.

The below figure 17 shows the list of features that are considered as important features among all other features in the dataset.

```
After Feature Selection Results
Feature Selection Algorithm Data
Index(['MinTemp', 'MaxTemp', 'Rainfall', 'WindGustSpeed', 'WindSpeed9am',
       'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am',
       'Pressure3pm', 'Temp9am', 'Temp3pm', 'RainToday', 'RainTomorrow',
       'x0_E', 'x0_ENE', 'x0_ESE', 'x0_N', 'x0_NE', 'x0_NNE', 'x0_NNW',
       'x0_NW', 'x0_S', 'x0_SE', 'x0_SSE', 'x0_SSW', 'x0_SW', 'x0_W', 'x0_WNW',
       'x0_WSW', 'x1_E', 'x1_ENE', 'x1_ESE', 'x1_N', 'x1_NE', 'x1_NNE',
       'x1_NNW', 'x1_NW', 'x1_S', 'x1_SE', 'x1_SSE', 'x1_SSW', 'x1_SW', 'x1_W',
       'x1_WNW', 'x1_WSW', 'x2_E', 'x2_ENE', 'x2_ESE', 'x2_N', 'x2_NE',
       'x2_NNE', 'x2_NNW', 'x2_NW', 'x2_S', 'x2_SE', 'x2_SSE', 'x2_SSW',
       'x2_SW', 'x2_W', 'x2_WNW', 'x2_WSW'],
      dtype='object')
```

Figure 17 List of features after performing Feature selection

V. MODEL CONSTRUCTION

After the preprocessing of data and dividing it into test and train splits we have built different models using different kinds of algorithms. All the classification algorithms that are linear based,

tree based, rule based, probability based, and ensemble models are evaluated, and results are discussed.

A. *Naive Bayes Classifier*

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem [14].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naïve [14].

They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real-life cases, the predictors are dependent, this hinders the performance of the classifier.

After applying Naïve Bayes classifier to the model following results are obtained as shown in figure 18.

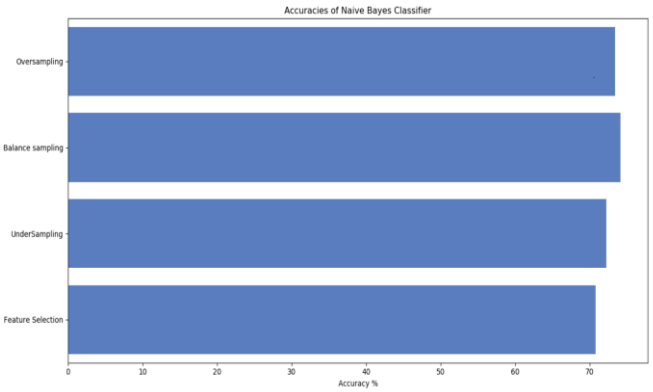

Figure 18 Accuracies for Naive Bayes Classifier

B. *K- Nearest Neighbor Classifier*

The rationale behind the nearest-neighbor classification is to find pre-determined "k" training models that must be close to new models of distance. The labels of the new models are defined from these neighbors. k the nearest neighbor classification has a fixed user-defined constant for the number of neighbors [15].

This algorithm can be summarized as:
A positive integer k is specified, along with a new sample. Model then selects k entries in dataset which are closest to new sample [16]. Most common classification class out of these k is found.

This project used *randomized_search()* to get the best value for K.

We classify this new instance into the majority class out of those K instances chosen KNN does not learn any model. Makes predictions just in time by calculating similarity between sample and training instances.

After applying KNN classifier 10 neighbors to the model following results are obtained as shown in figure 19.

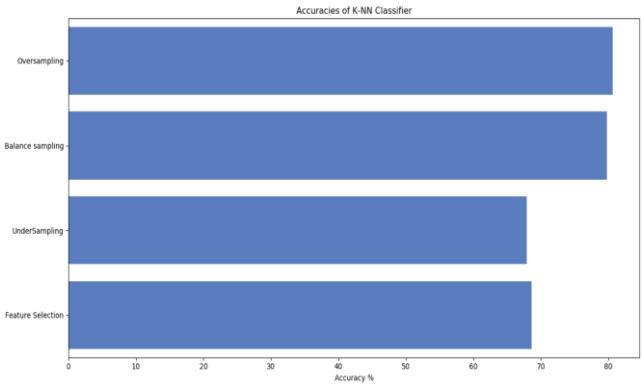

Figure 19: Accuracies for KNN classifier

C. *Decision Tree Classifier*

In decision analysis, decision trees can be used to visualize decisions and decisions making. In data mining, decision trees interpret data, but not decisions. Instead, the resulting classification tree can be used as input for decision making. A tree structure is constructed that breaks down the data set into smaller subsets, which ultimately leads to predictions. There are decision nodes to partition the data, and leaf nodes give predictions [24].

The root node replaces the splits data based on the most efficient feature. It has 2 measures, Gini impurities (criteria that reduce the likelihood of misclassification) and immorality (a measure of impurity) [25].

Entropy

$$Entropy = -\sum_j p_j \log_2 p_j$$

Gini Index

$$Gini = 1 - \sum_j p_j^2$$

After applying Decision Tree classifier to the model following results are obtained as shown in figure 20

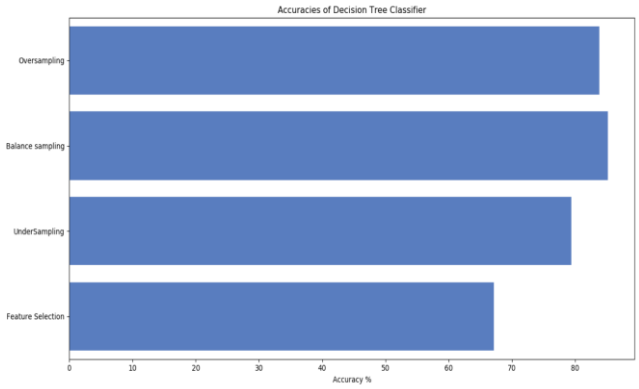

Figure 20 Accuracies of Decision Tree classifier

D. Dummy Classifier

Dummy classification is a classification that produces no insight into data, but only uses general rules to classify a given data. The behavior of the classifier is completely independent of the training data because the trends in the training data are completely ignored and a strategy is used to predict course labels [17].

It can only be used as a general benchmark for other classifiers, that is, any other classification should work well in a given dataset. This is especially useful for determining unbalanced data sets. Based on this philosophy, any analysis method for classification problems should be better than the random estimation method. [17].

After applying Dummy classifier to the model following results are obtained as shown in figure 21.

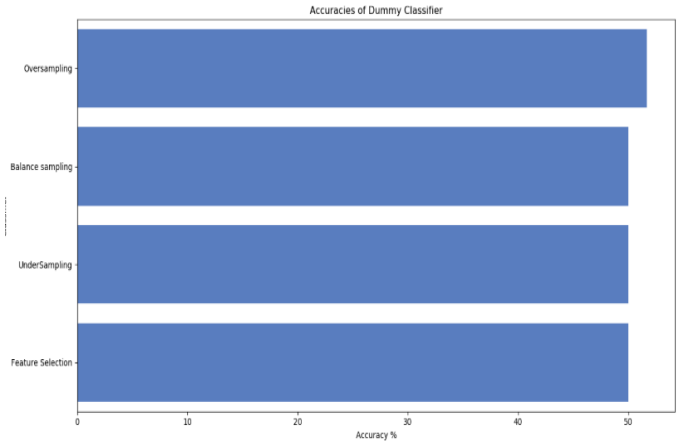

Figure 21 Accuracies of Dummy Classifier

E. Skope Rules

Skope-rules is a Python machine learning module built on top of scikit-learn and distributed under the 3-Clause BSD license. Skope-rules aims at learning logical, interpretable rules for "scoping" a target class, i.e. detecting with high precision instances of this class [18].

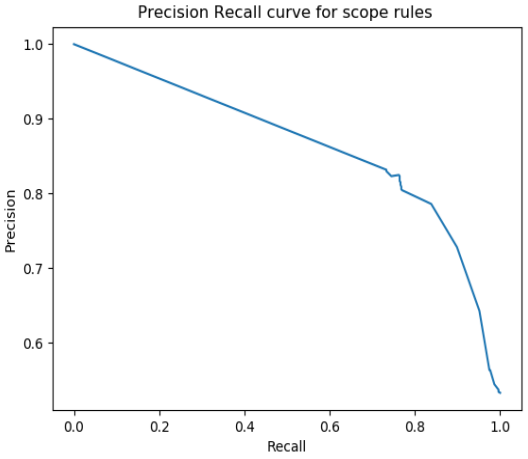

Figure 22 Graph for Precision and Recall using Skope Rules for Oversampled Data

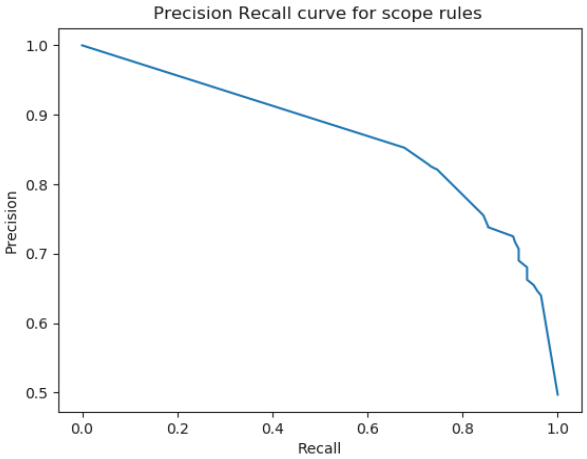

Figure 23 Graph for Precision and Recall using Skope Rules for Balance Sampled Data

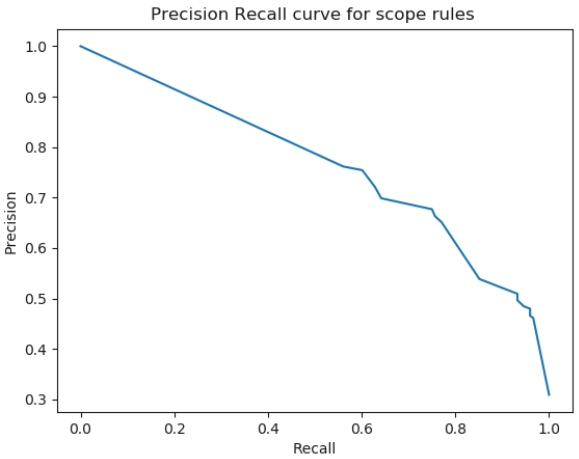

Figure 24 Graph for Precision and Recall using Skope Rules for Under Sampled Data

F. Random Forest Classifier

The random forest algorithm randomly selects observations and features to construct multiple decision trees, and then averages the results.

Deep Decision can cause tree overfitting. Random forests prevent overfilling by creating random subsets of properties and using these

subsets to build smaller trees. This will then merge the subtrees. Note that this does not work every time and it can also slow calculations depending on the number of random forest-built trees [19].

After applying Ensemble classifier to the model following results are obtained as shown in figure 25.

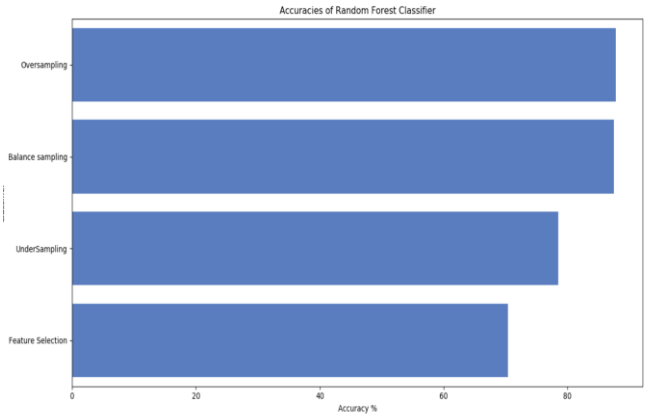

Figure 25: Accuracies for Random Forest Classifier

VI. EVALUATION TECHNIQUES

A. *K- fold Cross Validation*

The most commonly used cross-validation method is to divide the data into k times and k times. Each time the k-1 fold is used to build the model, the remaining folds are tested, and the process is repeated 10 times. With nearly 40,000 examples in our dataset, building the model and validating it with K-folding takes a long time [20].

Cross validation is a re-evaluation process used to evaluate machine learning models on limited data models.

This process has only one parameter called k, which represents the number of groups in which a given data sample is divided. Therefore, this process is often called k-fold cross-validation. When a specific value of k is chosen, it can be used instead of k in the model, for example, k = 10 times cross-validation [20].

	naive	random	decision	dummy	knn
0	0.771144	0.905473	0.905473	0.616915	0.915423
1	0.786070	0.835821	0.825871	0.731343	0.835821
2	0.845771	0.845771	0.835821	0.666667	0.786070
3	0.805000	0.880000	0.880000	0.715000	0.885000
4	0.765000	0.860000	0.865000	0.640000	0.865000
5	0.775000	0.855000	0.850000	0.735000	0.850000
6	0.770000	0.830000	0.835000	0.630000	0.840000
7	0.788945	0.814070	0.839196	0.678392	0.814070
8	0.793970	0.814070	0.819095	0.683417	0.834171
9	0.778894	0.824121	0.819095	0.673367	0.773869

Figure 26 K-fold for all the classifiers with accuracies

Above figure 26 shows the table of accuracies for 10-fold cross validation for all the machine learning algorithms.

B. *Test and Train split*

Before building the model, we need to split the data into training and testing. Therefore, the model built using the training and validated using testing set. There are many ways in which data can be split in our project we implemented test train split and cross validation.

In this method the data is divided into testing and training based on percentage split. As our data set is pretty big, we tried different data splits and we got best results at 75 percent split. So, Data is divided to training and testing data where 75 percent is taken as the training data and the remaining 25% is taken as testing data.

C. *Paired T- test*

Paired sample t-test (sometimes called correlation model t-test) is a statistical procedure used to determine whether the mean difference between two sets of observations is zero. In the paired sample t-test, two dimensions are taken for each subject or organization to generate paired observations. Common applications of paired sample t-tests include case-control studies or repeated measures models. Suppose you are interested in assessing the effectiveness of your company's training program. One approach you may consider is to measure the performance of the employee sample before and after completing the project and to analyze the differences using the paired sample t -test. [21].

Classifiers	precision	p-value
Naive Bayes	-3.6573016662596	0.86573012765
Naive & Decision	-3.6608256547875	0.00522880867
Naive & Dummy	13.183643749165	3.4421128373e-07
Naive and K-NN	-3.522100855846	0.0064939802886
Random & Dummy	15.540285686363	8.293088146e-08
Random & K-NN	-1.825906144807	0.1011508826938
Random & Decision	0.1739820059727	0.8657301276576
Dummy and K-NN	-12.27093088406	6.3649007379e-07
Dummy and Decision	-15.650647159333	7.797480626e-08
K-NN and Decision	1.7947992276466	0.10626150519886

Figure 27 Pair T-test values

D. *ROC Curve*

A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate, False Positive Rate [22].

True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (FPR) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$

A ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives [22]. The following figure shows a typical ROC curves for all the classifiers.

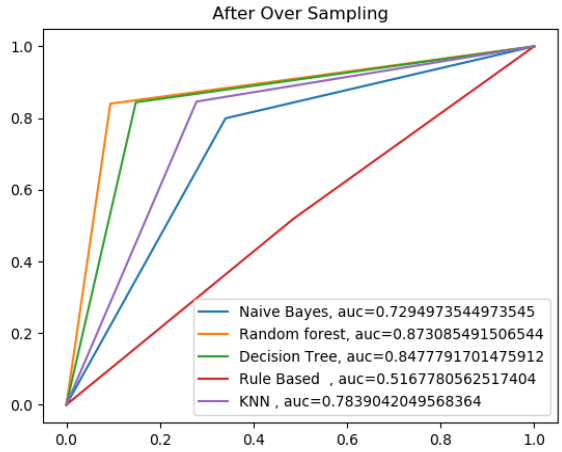

Figure 28 ROC for oversampling for the 5 Classifiers

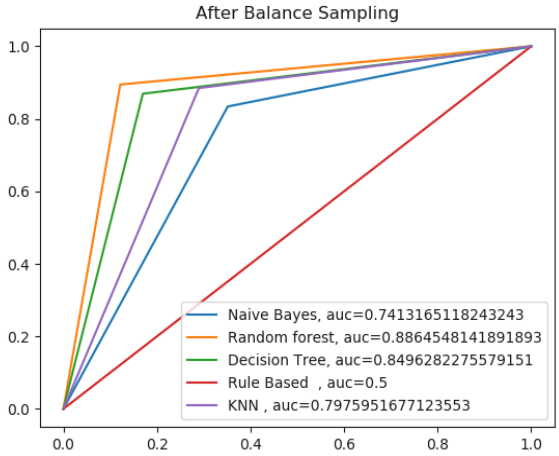

Figure 29 ROC for Balance Sampling for all 5 Algorithms

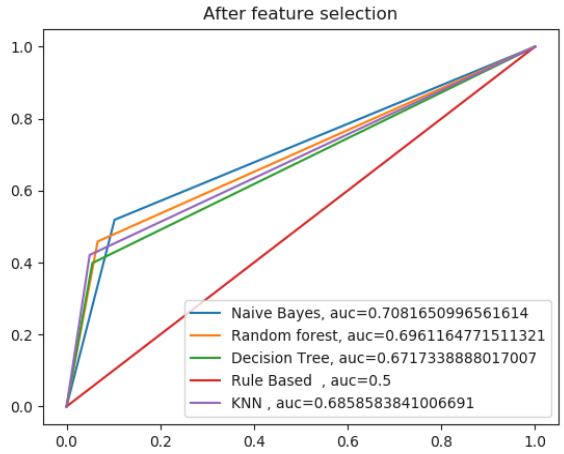

Figure 30 ROC for Feature Selection for all 5 Algorithms

E. Confusion Matrix

In the field of machine learning, especially in statistical classification problems, the confusion matrix is also known as the error matrix [23].
The confusion matrix is a frequently used table to describe the performance of a classification model (or "classification") on test data for a set of known true values. This allows you to visualize the performance of the algorithm.
It can easily detect confusion between classes, for example, one class is often mistaken for another class. Most performance indicators are based on the confusion matrix [23].

Below table illustrates the confusion matrices for all the algorithms

	Over sampling	Balance sampling	Under Sampling	Feature Selection
Navies Bayes	[[364 99] [168 99]]	[[336 85] [182 427]]	[[210 28] [121 120]]	[[473 64] [54 69]]
Random Forest	[[488 83] [44 430]]	[[459 63] [59 449]]	[[312 57] [19 91]]	[[492 70] [35 63]]
Decision Tree	[[448 76] [84 437]]	[[419 61] [99 451]]	[[286 43] [45 105]]	[[498 79] [29 54]]
Dummy Classifier	[[274 247] [258 266]]	[[0 0] [518 512]]	[[331 148] [0 0]]	[[527 133] [0 0]]
KNN	[[384 64] [148 449]]	[[368 59] [150 453]]	[[322 91] [9 57]]	[[501 77] [26 56]]

Figure 31: Confusion Matrices values for all the five algorithms

F. Recall

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (small number of FN) [23].
Recall is given by the relation:

$$Recall = \frac{TP}{TP + FN}$$

	Over sampling	Balance sampling	Under Sampling	Feature Selection
Navies Bayes	[0.7731, 0.6937]	[0.7980, 0.7011]	0.8823, 0.4979]	[0.8808, 0,5609]
Random Forest	[0.8546, 0.8960]	[0.8939, 0.8790]	[0.8559, 0.8558]	[0.8723, 0.6354]
Decision Tree	[0.8499, 0.8457]	[0.8651, 0.8348]	[0.8746, 0.7039]	[0.8615, 0.6463]
Dummy Classifier	[0.5259, 0.5076]	[0.0, 0.4970]	[0.6910, 0.0]	[0.7984, 0.0]
KNN	[0.8293, 0.7457]	[0.8618, 0.7512]	[0.7796, 0.8636]	[0.86678, 0.6829]

Figure32: Recall values for all the five algorithms

G. *Precision*

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled as positive is indeed positive (small number of FP)[23].

Precision is given by the relation:

$$Precision = \frac{TP}{TP + FP}$$

	Over sampling	Balance sampling	Under Sampling	Feature Selection
Navies Bayes	[0.6597, 0.7992]	[0.6486, 0.8339]	[0.6344, 0.8108]	[0.8975, 0.5187]
Random Forest	[0.9060, 0.8401]	[0.8783, 0.8945]	[0.9516, 0.6418]	[0.9335, 0.4586]
Decision Tree	[0.8515, 0.8440]	[0.83011, 0.8691]	[0.8640, 0.7229]	[0.9449, 0.3984]
Dummy Classifier	[0.5150, 0,5185]	[0.0 ,1.0]	[1.0, 0.0]	[1.0, 0.0]
KNN	[0.7218, 0.8460]	[0.7104, 0.8847]	[0.9728, 0.3851]	[0.9506, 0.4210]

Figure 33 Precision values for all the five algorithms

H. *F-Measure*

Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more. The F-Measure will always be nearer to the smaller value of Precision or Recall [23].

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision}$$

	Over sampling	Balance sampling	Under Sampling	Feature Selection
Navies Bayes	[0.7119, 0.7427]	[0.7156, 0.7618]	[0.7381, 0.6169]	[0.8890, 0.5390]
Random Forest	[0.8795, 0.8672]	[0.8860, 0.8867]	[0.8693, 0.7133]	[0.9019, 0.5327]
Decision Tree	[0.8507, 0.8448]	[0.8472, 0.8516]	[0.9010, 0.7335]	[0.9013, 0.4930]
Dummy Classifier	[0.5204, 0.5130]	[0.0, 0.6640]	[0.8172, 0.0]	[0.8879, 0.0]
KNN	[0.7718, 0.7926]	[0.7788, 0.8125]	[0.8655, 0.532]	[0.9067, 0.5209]

Figure 34 F1 score values for all the five algorithms

VII. LESSON LEARNED

By developing a machine learning model, I have gained the capability to solve a real time problem like rain prediction. I have keenly observed the importance of data preprocessing and feature engineering techniques. Capacity to implement different machine learning algorithms and conclude which algorithm works best for my dataset by comparing various evaluation metrics for each algorithm. Additionally, learnt the statistical testing to compare any two algorithms that are implemented in paper.

VIII. FUTURE CONSIDERATIONS

This project can be extended to predict the level of rain fall by considering the seasonal patterns in Australia. For example, the regions in southern hemisphere the monsoon season start from May and ends in September whereas the regions in northern hemisphere the monsoon season is from December to March. By considering the Geographical location of cities perfection of level of rainfall would be more accurate.

IX. CONCLUSION

Both Oversampling and Balance sampling techniques are generating similar results with all the machine learning algorithms that are mentioned in paper with better accuracies compared to under sampling as well as feature selection. After implementing various machine learning algorithms the model works well with Ensemble learning that is Random Forest Algorithm with an accuracy of 87%.On the other side Rule Based Classifier produced low accuracies compared to other classifiers like Decision Tree for Tree Based classifier , K-NN for Distance based classifier and Naive Bayes for Linear Based classifier.

X. REFERENCES

[1]. https://www.quora.com/What-is-the-difference-between-data-analytics-and-Machine-learning

[2].https://influentialpoints.com/Training/skew-and-kurtosis.htm

[3]. Kumar, S., & Chong, I. (2018). Correlation analysis to identify the effective data in machine learning: Prediction of depressive

disorder and emotion states. *International journal of environmental research and public health*, *15*(12), 2907.
[4].https://medium.com/@connor.anderson_42477/hot-or-not-heatmaps-and-correlation-matrix-plots-940088fa2806
[5]. https://medium.com/data-science-bootcamp/data-visualization-in-machine-learning-beyond-the-basics-baf2cbea8989
[6]. https://www.javatpoint.com/data-preprocessing-machine-learning
[7]. https://bambielli.com/til/2018-02-11-one-hot-encoding/
[8].https://www.codecademy.com/articles/normalization
[9]. https://www.ritchieng.com/machine-learning-dimensionality-reduction/
[10]. https://hackernoon.com/supervised-machine-learning-dimensional-reduction-and-principal-component-analysis-614dec1f6b4c
[11]. https://machinelearningmastery.com/k-fold-cross-validation/
[12]. https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18
[13]. Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70-79.
[14]. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
[15]. https://www.python-course.eu/k_nearest_neighbor_classifier.php
[16].http://www.mpia.de/homes/dgoulier/MLClasses/Course%20-%20Supervised%20Learning%20for%20Classification%20with%20R.html
[17]. https://www.geeksforgeeks.org/ml-dummy-classifiers-using-sklearn/
[18]. https://github.com/scikit-learn-contrib/skope-rules
[19]. https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76
[20]. https://machinelearningmastery.com/k-fold-cross-validation/
[21]. https://www.statisticssolutions.com/manova-analysis-paired-sample-t-test/
[22]. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
[23]. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/
[24]. https://www.bogotobogo.com/python/scikit-learn/scikt_machine_learning_Decision_Tree_Learning_Informatio in_Gain_IG_Impurity_Entropy_Gini_Classification_Error.php
[25]. http://benalexkeen.com/decision-tree-classifier-in-python-using-scikit-learn/