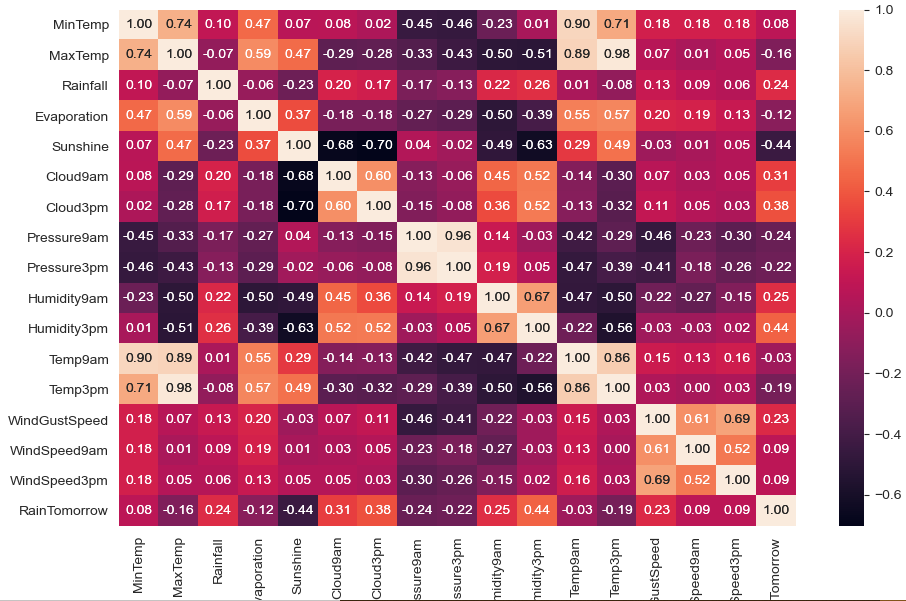
***A. Problem Statement:*** Predict next-day rain by training classification models on the target variable **RainTomorrow** based on dataset containing past 10 years of weather observations from many locations across Australia.

***B. Dataset Description***: The file **‘weatherAUS.csv’** contains *142,193* data records each containing 23 columns. We have a large input feature set of 22 columns consisting of numeric as well as categorical features and one target variable ‘RainTomorrow’ as described below-

*Numerical variables:*  'MinTemp', 'MaxTemp', 'Rainfall','WindGustSpeed','WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am','Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm'

*Categorical variables:* 'Date', 'Location','WindGustDir','WindDir9am','WindDir3pm','RainToday', 'RainTomorrow'.

Below is the correlation mapping of the features with the target variable ‘RainTomorrow’.



***C. Data Pre-processing:*** From the correlation plotting, it can be seen that some features like 'Humidity3pm', ‘Humidity9am', 'Cloud9am', 'Cloud3pm' have stronger say in predicting the next day rain while features like 'Pressure9am', 'Pressure3pm', ‘Sunshine’ has negative correlation with the target variable ‘RainTomorrow’. Also we have features like ……….. which contains Null values. So as a part of pre-processing we used the below pipeline –

i) *Data Cleaning*:

Many of the above dataset had Null values in some of the features like 'Cloud9am', 'Cloud3pm' which were refilled by the mean value of the respective features. Some Null values were present in the target variable ‘RainTomorrow’ and important feature ‘RainToday’ which had to be dropped to avoid misfitting of data.

ii) *Feature Selection*:

Features like ‘'Pressure9am', 'Pressure3pm', ‘Sunshine’, ‘Evaporation’, 'Temp9am', 'Temp3pm' etc have negative (or negligible) correlation with the target variable, hence dropped to avoid overfitting. Features having stronger say in predicting rain like ‘Rainfall’, 'Cloud9am', 'Cloud3pm', 'Humidity3pm', ‘Humidity9am' were selected for the training purpose.

The target variable ‘RainTomorrow’ and ‘RainToday’ has strong relation for not raining and class imbalance ratio of Yes and No is around 0.28 before cleaning ,as it can be seen from the below graphsFig2(a)(b), that shows that when ‘RainToday’ has a value ‘No’ then majority of examples in our target variable is also value ‘No’. So, for the ‘NaN’ values in target variable we used the ‘RainToday’ value, and then used the average of value in the next entry and previous day entry. Similar approach was done for the ‘RainToday’ feature.

Chart

Description automatically generated

Fig 2(a)

Chart, bar chart

Description automatically generated

Fig 2(b)

iii) *Standard Scalar*: Features like 'Rainfall','WindGustSpeed', 'Humidity9am’ etc have different scale (in respective units), hence were normalized using Scikit’s off the box StandardScalar transformer.

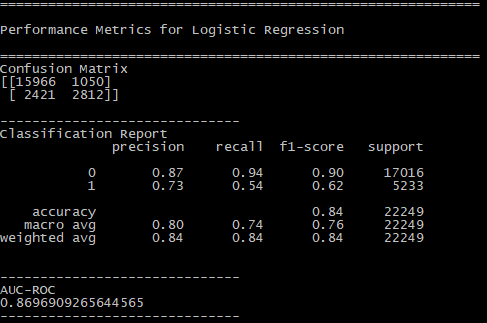
***D. Model Selection and Performance Evaluation:*** After pre-processing the data, we are ready to train different ML models and evaluate the performance on test set. Below are some points which needs to be highlighted -

i) *Imbalanced data:* The dataset, after pre-processing, had 74132 records out of which only 17504 records have ‘RainTomorrow’ value ‘Yes’. That means, if we map ***‘RainTomorrow=yes’*** as our **‘positive’** class then we only have **23% positive examples** in our dataset which is significantly less. Also, we used 70:30 as Train: Test split ratio to evaluate the models.

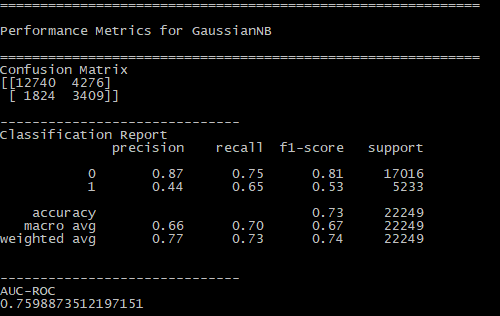
ii) *Choosing best metric* for such an imbalanced dataset can be tricky. Clearly, ***accuracy*** is not a good metric to go for which such class imbalance. ***Precision-Recall*** and ***AUCROC*** can be good choice for evaluation.

iii) While tuning the models, we have emphasized on ***Precision-0*** and ***Recall-0*** where ‘0’ denotes the negative class, in our case it is equivalent to predicting ‘No Rain’ for tomorrow. The logic behind is, *if we predict a day as ‘non-rainy’ and if it actually rains, it might have more adverse affect than predicting a day ‘Rainy’ and getting it wrong*. *So, the goal was to minimize the number of false negatives.*

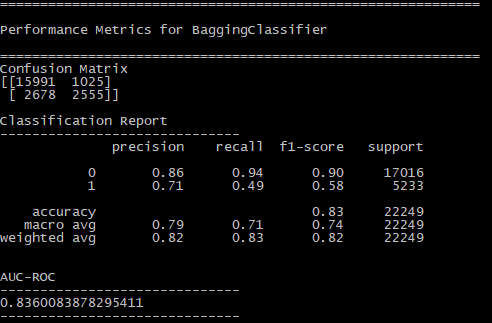
***(a) Logistic Regression:*** A logistic regression model with max\_iter=1000 and regularizing parameter C=1 resulted in the below performance metrics-



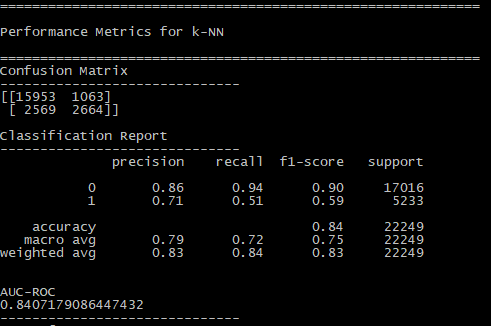
***(b) Gaussian Naïve Bayes:*** A GaussianNB model resulted in the below performance metrics-



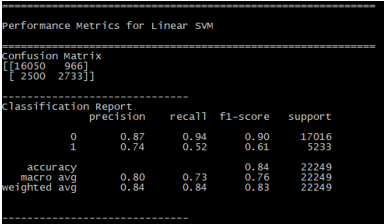
***(c) Bagging Classifier:*** A bagging classifier with Decision Tree as base-estimator and ten ensemble voters yielded a performance metrics of -



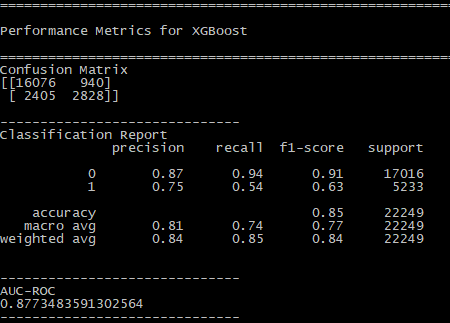
***(d) k-Nearest Neighbours:*** A k-NN model with 9 neighbours (tuned via GridSearch and Cross-validation) resulted in the below performance metrics-



***(e) Linear SVM:*** A linear support vector machine with default regulariszer C=1 and max\_iter=1000 yielded the below performance –



***(f) XGBoost :*** A parallel tree gradient boosting model with default learning rate=0.3 and max\_depth=6 resulted in the below performance metrics-



**E. Model Comparison and Performance Evaluation:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Class -0 (RainTomorrow='No')** | | | **Class -1 (RainTomorrow='Yes')** | | | **AUCROC** |
|  | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** |
| **Log Regression** | **87** | **94** | **90** | **73** | **54** | **62** | **87%** |
| **Gaussian NB** | **87** | **75** | **81** | **44** | **65** | **53** | **76%** |
| **Decision Tree** | **82** | **98** | **89** | **83** | **28** | **42** | **79%** |
| **BaggingClassifier** | **86** | **94** | **90** | **71** | **49** | **58** | **84%** |
| **k-NN** | **86** | **94** | **90** | **71** | **51** | **59** | **84%** |
| **Linear SVM** | **87** | **94** | **90** | **74** | **52** | **61** | **N/A** |
| **XGBoost** | **87** | **94** | **91** | **75** | **54** | **63** | **88%** |

**G: Future Scope:**

After trying different approaches for feature selection which are by examining correlations, using averages of 9am and 3pm for numerical features and once with all the features as it is, and by using different cleaning methods for filling NaN values instead of dropping those entries. We have observed that there is no significant change in the overall performance which may be accounted to the presence of outliers, which we also observed that the Rainfall ,WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity3pm columns may contain outliers, these features cannot be directly dropped as these features have strong correlations for occurrence of rain (science for rain). So, a model with good logical approach to eliminate the outliers might help in the making the prediction much better, and Neural networks will definitely be more effective in this prediction.

Chart, box and whisker chart

Description automatically generated