CS5525 Data analysis

Instructor: **Reza Jafari**

Next day rain prediction

Ran Lyu

05/05/2023

Table of content

Abstract

This report use exploration data analysis (EDA) explores this dataset, and the correlation between different weather variables and the target variable. During EDA outliers are replaced by NA value, and then missing data are filled by mode value. After the data cleansing, feature selection method, include PCA, random forest, etc., are applied to do feature engineering. In phase II, regression analysis methods are applied but found out not fit this classification problem. In phase III, report investigates the performance of various supervised machine learning models, including KNN, Naïve Bayesian in predicting weather patterns. In the last phase, unsupervised machine learning models is used. The findings suggest that all supervised machine learning classification model outperforms the other models in predicting rain patterns, with an average AUC around 0.85, the decision tree and KNN have the highest AUC of 0.87.

Introduction

Rain prediction has been a subject of interest for many years, and various techniques have been developed to forecast rainfall. Traditional methods, such as statistical models and numerical weather prediction, have been in use for a long time. However, these methods have limitations and cannot accurately predict rainfall in all situations.

In recent years, machine learning models have gained popularity in predicting rainfall due to their ability to analyze large datasets and capture complex patterns. Machine learning models such as logistic regression, random forests, and support vector machines have been used to predict rainfall based on historical weather data.

However, the accuracy of these models is highly dependent on the quality of the data used to train them. As a result, there is ongoing research to develop more sophisticated machine learning algorithms and to improve the quality and quantity of the weather data available for training.

Overall, the current state of rain prediction involves a combination of traditional methods and machine learning techniques. The accuracy of these methods is continuously improving, and there is ongoing research to develop better models and improve their performance in predicting rainfall.

Dataset description

This dataset comes from Kaggle: https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package

This dataset contains about 150k around 10 years of daily weather observations from many locations across Australia.
Originally, there are 22 features and 1 target value in the raw dataset.
MinTemp (numerical feature): minimum temperature of today

MaxTemp (numerical feature): maximum temperature of today
Rainfall (numerical feature): Rainfall is a meteorological variable that represents the amount of precipitation that falls to the ground in a particular region over a specified period of time
Evaporation (numerical feature): Evaporation is the process by which a liquid (usually water) changes into a gas (usually water vapor) and escapes into the surrounding air. It occurs when the molecules at the surface of a liquid gain enough energy to break free from the liquid's surface and enter the gas phase.
Sunshine (numerical feature): is a meteorological variable that represents the amount of sunlight that reaches the Earth's surface
WindGustDir (categorical feature): Wind direction
WindGustSpeed (categorical feature): wind speed
WindDir9am (categorical feature): Wind direction at 9 am
WindDir3pm (categorical feature): Wind direction at 3 pm
WindSpeed9am (numerical feature): wind direction at 9 am
WindSpeed3pm (numerical feature): wind direction at 3 pm
Humidity9am (numerical feature): Humidity at 9 am
Humidity3pm (numerical feature): Humidity at 3 pm
Pressure9am (numerical feature): barometric pressure at 9 am
Pressure3pm (numerical feature): barometric pressure at 3 pm
Cloud9am (numerical feature): the amount or coverage of cloud in the sky at 9 am
Cloud3pm (numerical feature): the amount or coverage of cloud in the sky at 3 pm
Temp9am (numerical feature): temperature at 9 am
Temp3pm (numerical feature): temperature at 3 pm
RainToday (categorical feature): is it raining today
RainTomorrow (categorical feature): is the target variable to predict. It means did it rain the next day. This column is Yes if the rain for that day was 1mm or more.

Source & Acknowledgements

Observations were drawn from numerous weather stations. The daily observations are available from http://www.bom.gov.au/climate/data.
An example of latest weather observations in
Canberra: http://www.bom.gov.au/climate/dwo/IDCJDW2801.latest.shtml
Definitions adapted from http://www.bom.gov.au/climate/dwo/IDCJDW0000.shtml
Data source: http://www.bom.gov.au/climate/dwo/ and http://www.bom.gov.au/climate/data.

Experiment

Phase 1: EDA

- Data preprocessing
  - Data cleaning methods:
    1) For features: 1. Check nan data; 2. Anomaly detection and check outlier; 3. Replace outlier with nan; 4. Fill nan with mode

2) For target: 1. Check nan data; 2. Remove nan observations

```
====df.isnull().sum()====
Date                0
Location            0
MinTemp          1485
MaxTemp          1261
Rainfall         3261
Evaporation     62790
Sunshine        69835
WindGustDir     10326
WindGustSpeed   10263
WindDir9am      10566
WindDir3pm       4228
WindSpeed9am     1767
WindSpeed3pm     3062
Humidity9am      2654
Humidity3pm      4507
Pressure9am     15065
Pressure3pm     15028
Cloud9am        55888
Cloud3pm        59358
Temp9am          1767
Temp3pm          3609
RainToday        3261
RainTomorrow     3267
dtype: int64
====After drop NA>40% features; target is NA and fill NA====
(142193, 19)
```

Figure1: nan value count of dataset

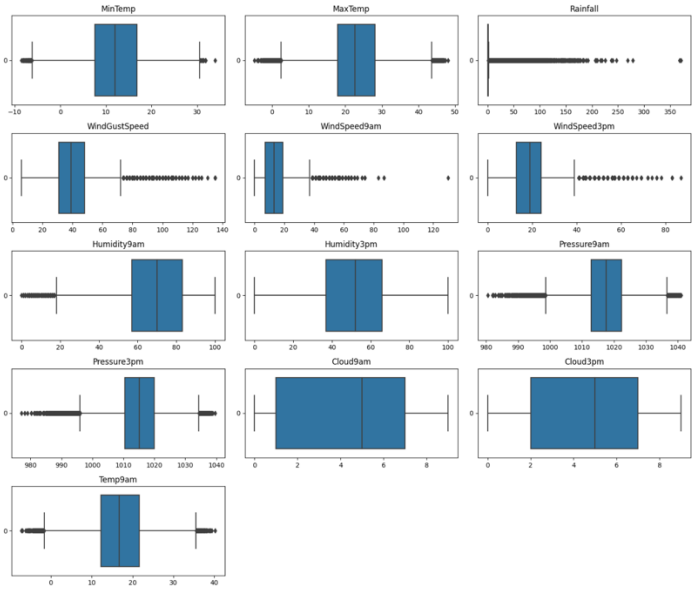


Figure2: outlier value visualization for each feature

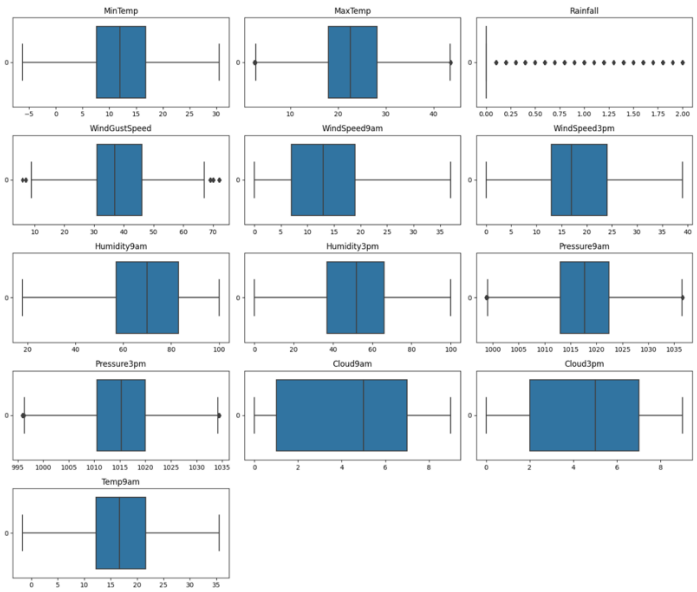


Figure3: After remove outlier value visualization for each feature

- Dimensionality reduction:
  1) Though both ‘Sunshine’, and ‘Evaporation’ have more than 40% missing value, and ‘Cloud9am’and ‘Cloud3pm’have more than 35% missing observations; their feature importance is relatively high; so I cannot safely drop there features. It is possible that the missingness in the feature is related to the outcome variable, and therefore, retaining the feature and handling the missing values appropriately may improve the model's predictive power, so I have to keep these features.
  2) Date and location features are often dropped in rain prediction models because they do not have a direct causal relationship with rainfall. Instead, these features may indirectly influence rainfall through their relationship with other variables such as temperature, humidity, or atmospheric pressure. Moreover, including date and location features in the model can lead to overfitting and poor generalization to new data. Therefore, it is common practice to drop these features unless there is a specific reason to include them.
  3) Down sampling and plot pairwise correlation

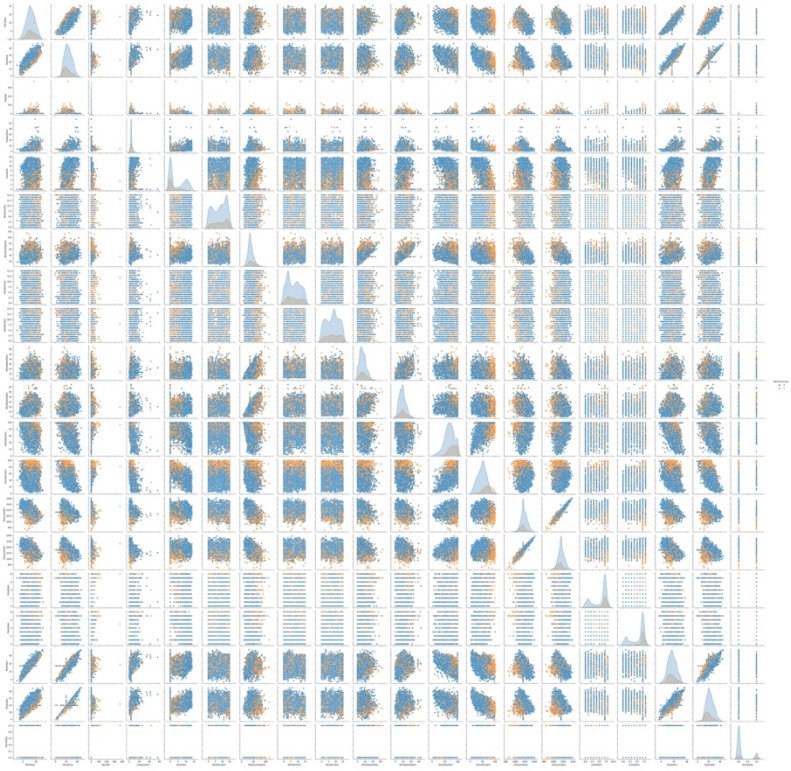


Figure4: features correlation pairwise plots

High correlated features detected between:

Temp3pm and MaxTemp, Temp9am and MinTemp, Temp9am and MaxTemp, Temp3pm and Temp9am, Pressure3pm and Pressure9am

From the above figure, we can see that when the minimum temperature and max temperature are close two each other, there are high chance to rain tomorrow, and there is a higher chance to rain when the humidity is high, and also when the pressure is low.

- One-hot encoding, feature transform and standardization

1) Apply one-hot encoding on categorical feature, such as ‘WindDir3pm’with value: WNW, WSW, E, NW, W, SSE;

   In general, one-hot encoding has several benefits over transforming categorical features into multiple numerical values, for example: One-hot encoding preserves the categorical nature of the feature, which can be important in some machine learning models; and it can improve the performance of some machine learning models, such as decision trees and random forests, by allowing them to make more accurate splits based on categorical features.

2) Feature transform

   Both ‘rainToday’ and ‘rainTomorrow’ have value ‘yes’ or ‘no’, mapping to 0 and 1.

3) Standardization

   Standardized the numerical features (One-hot encoding features not included).

   Without standardization, the importance of a feature would depend on the scale of the feature, which can be misleading. Standardization ensures that all features are on a similar scale and avoids giving more weight to features with larger values, and it helps to improve the performance of machine learning algorithms, such as K-nearest neighbors (KNN), and logistic regression.

■ Feature selection

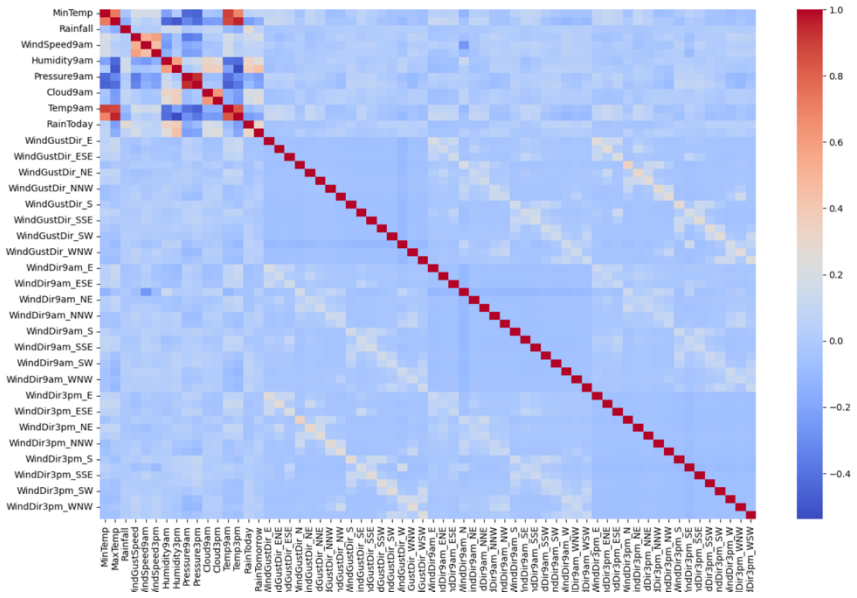


Figure5: features correlation heatmap

After One-hot encoding, there are 40 more features. Too many features reduce the computational efficiency in training process, so I need to drop less important features.

1) Principal Component Analysis

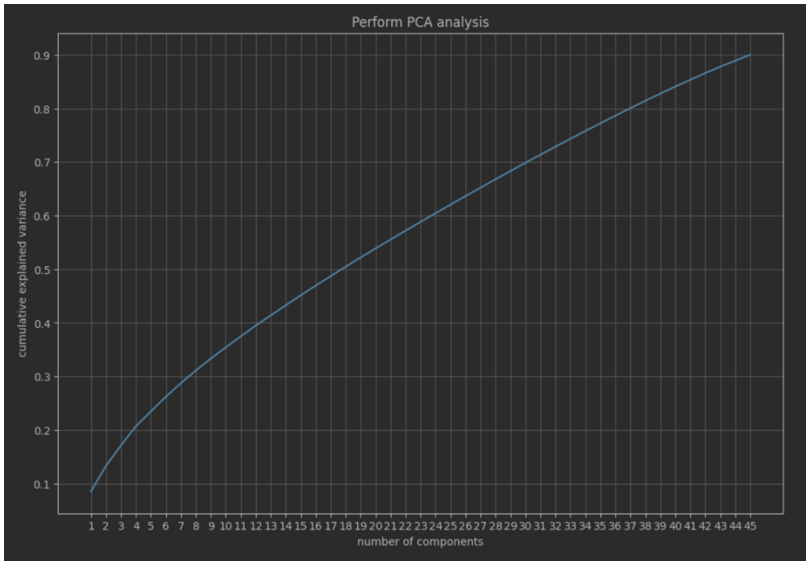


Figure6: Principal Component Analysis

2) Random Forest Analysis

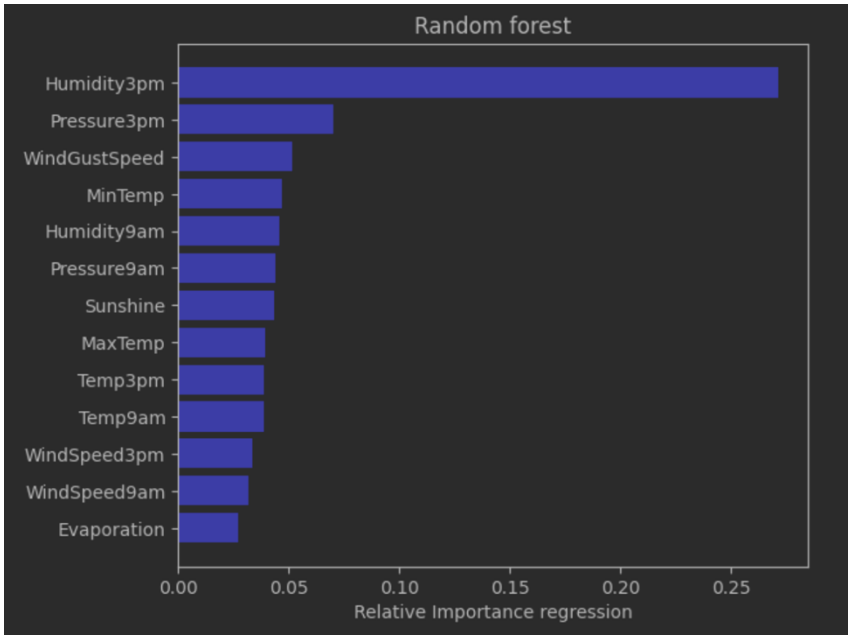


Figure7: random forest using regression classifier

```
Selected:  {'Temp3pm', 'Humidity3pm', 'Sunshine', 'Temp9am', 'WindSpeed3pm', 'MaxTemp',
'MinTemp', 'Pressure9am', 'WindSpeed9am', 'Pressure3pm', 'Humidity9am', 'WindGustSpeed',
'Evaporation'}
eliminated:  {'WindGustDir_SW', 'WindDir9am_NNW', 'Cloud9am', 'WindDir3pm_WNW', 'RainToday',
'WindGustDir_W', 'WindDir9am_SSE', 'WindDir9am_ESE', 'WindDir9am_NE', 'WindGustDir_ENE',
'WindGustDir_NW', 'WindGustDir_WNW', 'WindDir9am_ENE', 'WindDir9am_SE', 'WindDir3pm_SW',
'WindDir3pm_WSW', 'WindDir9am_NW', 'Rainfall', 'WindDir9am_WNW', 'WindGustDir_WSW',
'WindDir3pm_S', 'WindDir3pm_SSE', 'WindGustDir_SE', 'WindGustDir_N', 'WindGustDir_S',
'WindDir3pm_NNE', 'WindDir3pm_NNW', 'WindGustDir_NNW', 'WindGustDir_SSW', 'WindDir9am_S',
'WindGustDir_NE', 'WindGustDir_E', 'WindGustDir_SSE', 'WindDir3pm_SSW', 'WindDir9am_WSW',
'WindGustDir_NNE', 'WindDir3pm_ESE', 'WindDir9am_N', 'WindDir9am_NNE', 'WindDir9am_SW',
'WindDir3pm_N', 'WindDir3pm_W', 'WindDir9am_W', 'WindDir3pm_E', 'WindDir9am_SSW',
'WindDir9am_E', 'WindDir3pm_NW', 'WindGustDir_ESE', 'WindDir3pm_SE', 'WindDir3pm_NE',
'Cloud3pm', 'WindDir3pm_ENE'}
```

Figure8: Selected and eliminated features result

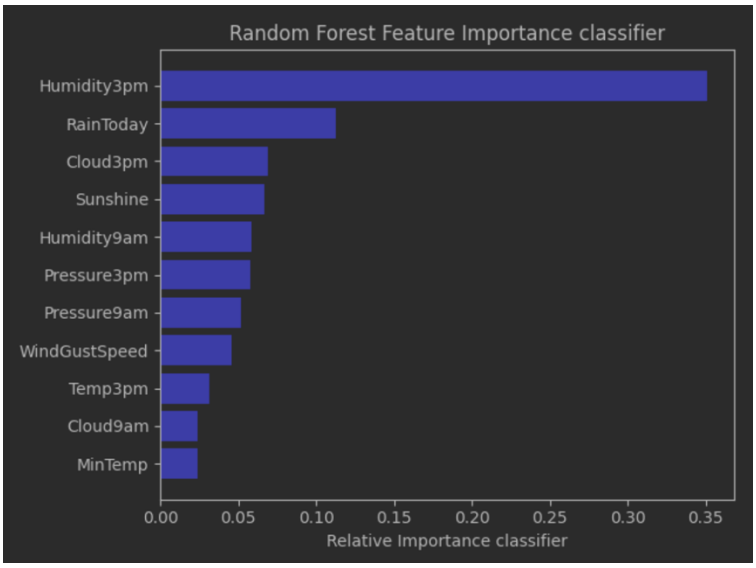


Figure9: Random forest using classification classifier selected features

3) Backward stepwise regression analysis

```
Eliminated features: {'WindGustDir_SW', 'WindDir3pm_NNE', 'WindGustDir_S', 'WindDir9am_NNW',
'Cloud9am', 'Humidity9am', 'WindGustDir_NNW', 'WindGustDir_SSW', 'WindGustDir_E',
'WindGustDir_SSE', 'WindDir3pm_SSW', 'WindGustDir_NW', 'WindGustDir_ENE', 'WindGustDir_WNW',
'Temp3pm', 'WindDir3pm_W', 'WindDir3pm_WSW', 'WindDir9am_E', 'WindGustDir_WSW', 'WindDir3pm_S',
 'MaxTemp', 'WindGustDir_SE', 'WindGustDir_N'}
Selected features: ['MinTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed',
'WindSpeed9am', 'WindSpeed3pm', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud3pm',
'Temp9am', 'RainToday', 'WindGustDir_ESE', 'WindGustDir_NE', 'WindGustDir_NNE',
'WindGustDir_W', 'WindDir9am_ENE', 'WindDir9am_ESE', 'WindDir9am_N', 'WindDir9am_NE',
'WindDir9am_NNE', 'WindDir9am_NW', 'WindDir9am_S', 'WindDir9am_SE', 'WindDir9am_SSE',
'WindDir9am_SSW', 'WindDir9am_SW', 'WindDir9am_W', 'WindDir9am_WNW', 'WindDir9am_WSW',
'WindDir3pm_E', 'WindDir3pm_ENE', 'WindDir3pm_ESE', 'WindDir3pm_N', 'WindDir3pm_NE',
'WindDir3pm_NNW', 'WindDir3pm_NW', 'WindDir3pm_SE', 'WindDir3pm_SSE', 'WindDir3pm_SW',
'WindDir3pm_WNW']
                            OLS Regression Results
==============================================================================
Dep. Variable:           RainTomorrow   R-squared (uncentered):                   0.232
Model:                          OLS   Adj. R-squared (uncentered):              0.232
Method:                Least Squares   F-statistic:                              819.6
Date:               Fri, 05 May 2023   Prob (F-statistic):                        0.00
Time:                       01:10:51   Log-Likelihood:                         -61421.
No. Observations:             113754   AIC:                                 1.229e+05
Df Residuals:                 113712   BIC:                                 1.233e+05
Df Model:                         42
Covariance Type:           nonrobust
```

Figure10: Backward stepwise regression analysis selected features

OLS regression results

Phase 2: Regression Analysis

In the previous phase backward stepwise regression analysis, I got the R-squared and adjust R-squared are 0.232 (whole dataset). This number varies when I use 1000 samples, or 10,000 samples, but not too much difference all below 0.5, which indicates that it is not the best option to use regression analysis (not include logistic regression, will be discussed in the phase 3).

Phase 3: Classification Analysis

- Logistic Regression Classifier

```
Logistic regression time: 10.406398057937622
0.8385315939379022 0.8385315939379022 0.8385315939379022
AUC of 9: 0.85
Confusion matrix of 9: [[20957  1152]
 [ 3440  2890]]
              precision    recall  f1-score   support

           0       0.86      0.95      0.90     22109
           1       0.71      0.46      0.56      6330

    accuracy                           0.84     28439
   macro avg       0.79      0.70      0.73     28439
weighted avg       0.83      0.84      0.82     28439
```

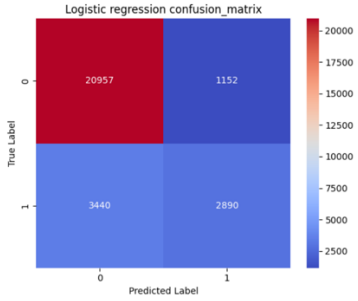


Figure11: prediction result using logistic regression classifier

- Decision Tree Classifier

```
best_params_: {'max_depth': 10}
Decision tree time: 7.588354110717773
precision: 0.84 recal: 0.84 f1: 0.84
AUC of 9: 0.84
Confusion matrix of 9: [[20892  1217]
 [ 3386  2944]]
              precision    recall  f1-score   support

           0       0.86      0.94      0.90     22109
           1       0.71      0.47      0.56      6330

    accuracy                           0.84     28439
   macro avg       0.78      0.71      0.73     28439
weighted avg       0.83      0.84      0.83     28439
```

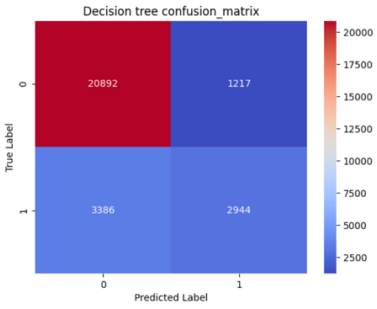


Figure12: prediction result using decision tree classifier

- K nearest neighbors Classifier

```
best_params_: {'n_neighbors': 30}
KNN time: 68.14067602157593
precision: 0.84 recal: 0.84 f1: 0.84
AUC of 9: 0.86
Confusion matrix of 9: [[21188   921]
 [ 3550  2780]]
              precision    recall  f1-score   support

           0       0.86      0.96      0.90     22109
           1       0.75      0.44      0.55      6330

    accuracy                           0.84     28439
   macro avg       0.80      0.70      0.73     28439
weighted avg       0.83      0.84      0.83     28439
```

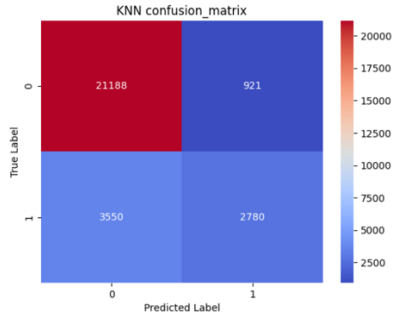


Figure13: prediction result using kNN classifier

- Naive Bayesian Classifier

```
Naive bayesian time: 68.66936492919922
precision: 0.80 recal: 0.80 f1: 0.80
AUC of 9: 0.83
Confusion matrix of 9: [[18757  3352]
 [ 2334  3996]]
              precision    recall  f1-score   support

           0       0.89      0.85      0.87     22109
           1       0.54      0.63      0.58      6330

    accuracy                           0.80     28439
   macro avg       0.72      0.74      0.73     28439
weighted avg       0.81      0.80      0.81     28439
```

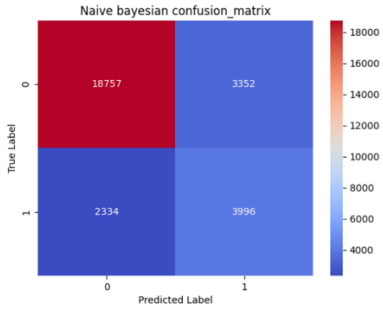


Figure14: prediction result using naive Bayesian classifier

- Random forest Classifier

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
best_params_: {'max_depth': 15, 'n_estimators': 200}
Random forest time: 727.5524237155914
precision: 0.85 recal: 0.85 f1: 0.85
AUC of 9: 0.87
Confusion matrix of 9: [[21136   973]
 [ 3365  2965]]
              precision    recall  f1-score   support

           0       0.86      0.96      0.91     22109
           1       0.75      0.47      0.58      6330

    accuracy                           0.85     28439
   macro avg       0.81      0.71      0.74     28439
weighted avg       0.84      0.85      0.83     28439
```

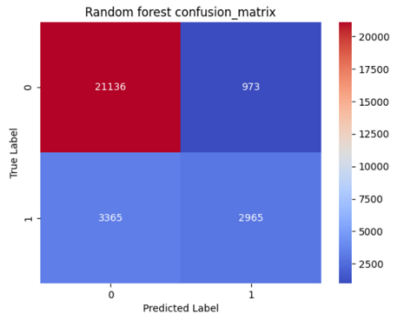


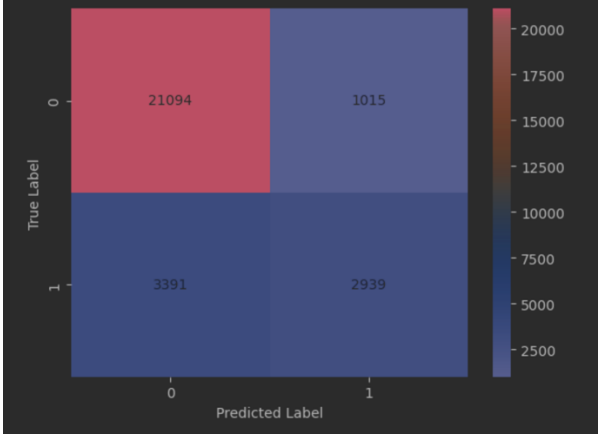
Figure15: prediction result using random forest classifier

- Neural network

```
711/711 [==============================] - 5s 6ms/step - loss: 0.0000e+00 -
accuracy: 0.7754 - val_loss: 0.0000e+00 - val_accuracy: 0.7754
Epoch 9/10
711/711 [==============================] - 4s 6ms/step - loss: 0.0000e+00 -
accuracy: 0.7754 - val_loss: 0.0000e+00 - val_accuracy: 0.7754
Epoch 10/10
711/711 [==============================] - 5s 6ms/step - loss: 0.0000e+00 -
accuracy: 0.7754 - val_loss: 0.0000e+00 - val_accuracy: 0.7754
precision: 0.85 recal: 0.85 f1: 0.85
AUC of 9: 0.87
Confusion matrix of 9: [[21094  1015]
 [ 3391  2939]]
              precision    recall  f1-score   support

           0       0.86      0.95      0.91     22109
           1       0.74      0.46      0.57      6330

    accuracy                           0.85     28439
   macro avg       0.80      0.71      0.74     28439
weighted avg       0.84      0.85      0.83     28439
```



Phase 4: Unsupervised learning feasibility analysis

Unsupervised learning is generally not suitable for rainfall prediction because it requires labeled data (i.e., data with known outcomes) to make predictions. Rainfall prediction is a supervised learning problem, where the goal is to use historical weather data to predict whether it will rain or not in the future. In this case, the data is labeled based on whether it rained or not on a given day. Therefore, supervised learning techniques can be used to train a model on historical data and predict future rainfall.

Unsupervised learning can, however, be used for other tasks related to weather analysis, such as clustering similar weather patterns or identifying anomalies in weather data. These tasks do not require labeled data and can be accomplished using unsupervised learning techniques such as clustering or anomaly detection.

Conclusion

- Considering prediction AUC, random forest Classifier and neural network have the best result, which are 0.87; In some cases, the time required for model training may not be a significant concern, especially if the training can be done offline or during periods of low usage.
- Considering prediction AUC and timing, k-nearest neighbor classifier has the best performance. AUC is 0.86, and takes only less than 10% time consuming by random forest model training. Logistic Regression Classifier model training only takes 1/6 time compare with k-nearest neighbor and has AUC of 0.85. In some real-time or near-real-time applications, the training time can be critical, and the model needs to be trained quickly to make timely predictions.
- Logistic Regression, Decision tree, Naive Bayes have better Interpretability
- false negatives (type II error) vs false positives (type I error/false alarm) rate: k-nearest neighbor and random forest has high type II error, and low type I error; whereas naïve Bayesian has high type I error and low type II error. If the model fails to predict rainfall when it does occur, it could lead to unpreparedness for floods or other weather-related disasters, which can be dangerous and cause significant damage. So, for flood prevention purpose, it is better to use low type I error prediction model, which is naïve Bayesian in this case.

  In conclusion, due to different requirement, the best prediction model varies.

Future work:

1. Test average new records prediction time and accuracy, which are popularly used in the real-life applications.
2. Some optimization. For example: neural network classifier, when the accuracy is constant, stop to save computation time.

Appendix:

Source code: https://github.com/RoseLV/Cs5525/blob/master/FTP.py