Weather Prediction in Australia

Group 1 - Daniel Poon, Faiyam Islam, Geetha Balakrishnan, Tin Duong

1. Project Overview

The "Rain in Australia" dataset available on Kaggle contains daily weather observations from numerous weather stations across Australia, spanning over a period of 10 years from 2008 to 2018. The dataset includes features such as location, date, temperature, humidity, rainfall, wind speed and direction, and other atmospheric measurements. The objective of this project is to analyse this dataset and develop a model that can accurately predict whether or not it will rain tomorrow in a given location in Australia based on the weather conditions. Various classification techniques will be used to determine the target variable, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM) and Naive Bayes techniques. Based on these techniques, we can utilise evaluation metrics including: Accuracy, Precision, Recall and F1 Score.

1.1. Project Timeline

Week 1 (May 1st - May 7th):

- (May 1-2): Meet with the team and review the dataset. Assign roles and responsibilities to each team member. [Everyone]
- (May 3-4): Conduct preliminary data exploration and perform data cleaning, including dealing with missing values, outlier detection, and data normalization. Document any changes made to the dataset. [Everyone]
- (May 5-7): Perform exploratory data analysis, generate descriptive statistics and visualizations, and gain insights into the relationships between different variables in the dataset. [Faiyam, Geetha]

Week 2 (May 8th - May 14th):

- (May 8-9): Decide on the classification algorithm to use and split the dataset into training and testing sets. [Tin, Daniel, Faiyam]
- (May 10-12): Train the selected classification algorithm on the training set and fine-tune the hyperparameters. [Tin, Daniel, Faiyam]
- (May 13-14): Test the trained model on the testing set, evaluate its performance using appropriate evaluation metrics, and document the results. [Geetha]

Week 3 (May 15th - May 21st):

- (May 15-16): Analyze the results of the model and identify areas for improvement. Explore alternative algorithms or techniques to improve the model's performance. [Faiyam, Daniel]
- (May 17-19): Fine-tune the model and repeat the testing and evaluation process to determine if there is an improvement in performance. [Geetha, Tin]
- (May 20-21): Finalise the project, summarize the findings, and prepare the report or presentation. [Everyone]

2. Data Pre-processing and Cleaning

```
# Import weatherAUS dataset
weather_dat <- read.csv("weatherAUS.csv", header = TRUE, stringsAsFactors = FALSE)

# Load the required packages
library(dplyr, warn.conflicts = FALSE) # For data manipulation
library(ggplot2) # For data visualisation
```

2.1. Missing values and summary of dataset

```
# Calculate the percentage of missing values
missing_percentage <- sum(is.na(weather_dat)) / (nrow(weather_dat) * ncol(weather_dat)) *

# Remove missing values
weather_dat <- na.omit(weather_dat)

# Quick summary of the dataset
# summary(weather_dat) # commenting to save pages
```

There are quite a few challenges with this dataset as the data is imbalanced for target variable-RainTomorrow as there are more no-rain observations than rain. Also, we can see that around 10.2597457% of missing values in this dataset which could have certain level of impact in the accuracy of classification models.

2.2. Data Transformation on `weather_dat`

Converting the categorical features to dummies will allow them to be usable in our classification models in the later stages of the project, often called encoding.

```r
# Convert categorical variables to factors
cat_cols <- c("Location", "WindGustDir", "WindDir9am", "WindDir3pm", "RainToday", "RainTom

weather_dat[cat_cols] <- lapply(weather_dat[cat_cols], factor)
# Convert RainToday and RainTomorrow to binary variables
weather_dat$RainToday <- as.integer(weather_dat$RainToday == "Yes")
weather_dat$RainTomorrow <- as.integer(weather_dat$RainTomorrow == "Yes")
```

3. Exploratory Data Analysis

```r
# Select variables of interest
vars <- c("MinTemp", "MaxTemp", "Rainfall", "Evaporation")

# Create a histogram for each variable and combine them into one output
suppressWarnings(library(tidyr))
ggplot(data = gather(weather_dat, key = "Variable", value = "Value", vars)) +
  geom_histogram(aes(x = Value, fill = Variable),
                 binwidth = 2, alpha = 0.5, position = "dodge") +
  facet_wrap(~ Variable, scales = "free") +
  labs(title = "Histograms of Weather Variables",
       x = "Value", y = "Frequency") +
  theme_minimal()
```