ทำนายการตกของฝนในประเทศออสเตรเลีย

Rain in Australia Prediction

โดย

นางสาวธัญชนก นากผสม

เลขทะเบียน 6009680106 สาขาสถิติ

เสนอ

ผู้ช่วยศาสตราจารย์ ดร. ประภาพร รัตนธำรง

รายงานนี้เป็นส่วนหนึ่งของรายวิชาการจำลองคอมพิวเตอร์และเทคนิคการพยากรณ์สำหรับธุรกิจ

CS358 COMPUTER SIMULATION AND FORECASTING TECHNIQUES IN BUSINESS

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ภาคเรียนที่ 2 ปีการศึกษา 2563

คำนำ

รายงานการศึกษาโครงงานฉบับนี้เป็นส่วนหนึ่งของวิชาการจำลองคอมพิวเตอร์และเทคนิคการพยากรณ์ สำหรับธุรกิจจัดทำขึ้นเพื่อศึกษาเรื่อง ทำนายการตกของฝนในประเทศออสเตรเลีย โดยศึกษาตามแผน CRISP-DM และดำเนินการในแต่ละขั้นตอนโดยใช้ Google Cloud Platform ผ่านการใช้ Cloud Storage, BigQuery, Data Studio และ Cloud AI Platform Notebooks

ผู้จัดทำขอขอบพระคุณผู้ช่วยศาสตร์ตราจารย์ ดร.ประภาพร รัตนธำรง ผู้ให้ความรู้ ให้คำแนะนำ ตลอดจนให้ความช่วยเหลือในด้านต่าง ๆ ในการศึกษาและจัดทำรายงานฉบับนี้จนสำเร็จลุล่วงด้วยดี

ผู้จัดทำหวังเป็นอย่างยิ่งว่าเนื้อหาในรายงานฉบับนี้จะเป็นประโยชน์ต่อผู้สนใจหากมีสิ่งใดในรายงานฉบับนี้จะต้องปรับปรุง คณะผู้จัดทำขอน้อมรับในข้อชี้แนะและจะนำไปแก้ไขให้ถูกต้องสมบูรณ์ต่อไป

ผู้จัดทำ

นางสาวธัญชนก นากผสม

สารบัญ

สารบัญรูปภาพ

สารบัญตาราง

ที่มาและความสำคัญ

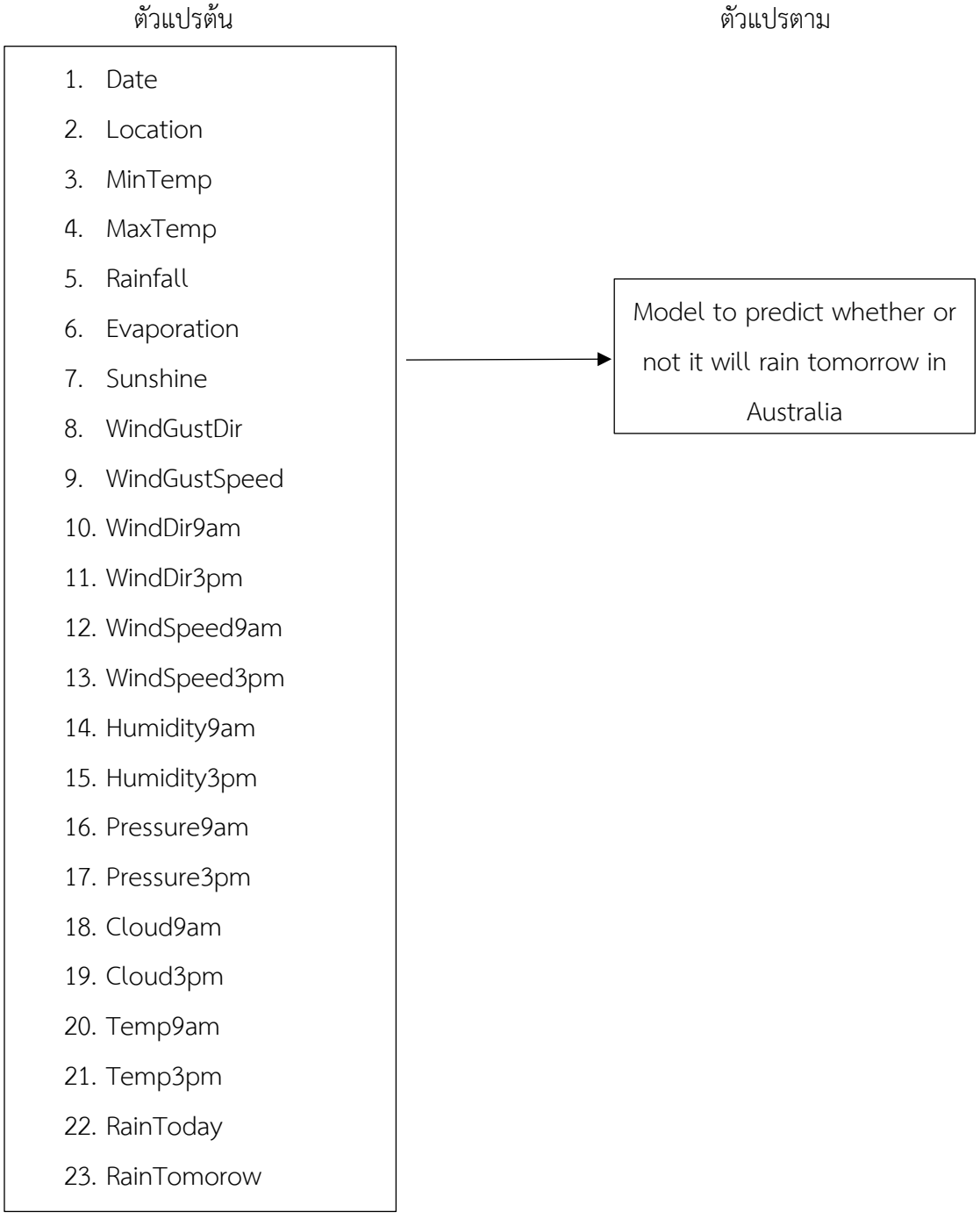
ปัจจุบันหลายประเทศทั่วโลกต่างกำลังเผชิญกับปัญหาภัยพิบัติที่เกิดจากการเปลี่ยนแปลงของสภาพภูมิอากาศ ไม่ว่าจะเป็นพายุเฮอริเคน น้ำท่วม ดินถล่ม ซึ่งทำให้มีผู้เสียชีวิตและได้รับความเดือดร้อนเป็นจำนวนมาก ดังนั้น หากนักวิทยาศาสตร์สามารถทำนายสภาพอากาศได้ล่วงหน้าอย่างถูกต้องแม่นยำ จะส่งผลให้ทางรัฐบาลหรือหน่วยงานที่เกี่ยวข้องในแต่ละประเทศสามารถวางแผนเพื่อป้องกันและลดความเสียหายที่อาจเกิดขึ้นได้จากการเกิดภัยพิบัติที่เกิดจากการเปลี่ยนแปลงสภาพภูมิอากาศได้

ในการทำนายสภาพอากาศล่วงหน้าให้ถูกต้องแม่นยำนั้น จำเป็นต้องใช้ข้อมูลด้านภูมิศาสตร์เป็นจำนวนมากไม่ว่าจะเป็น ภาพถ่ายดาวเทียม, ข้อมูลชั้นบรรยากาศ, ความชื้น, ฯลฯ ซึ่งในปัจจุบันมีเทคโนโลยีใหม่ๆมากมายที่ช่วยให้สามารถเก็บบันทึกข้อมูลทางด้านภูมิศาสตร์ได้อย่างรวดเร็วมากขึ้น ส่งผลให้แนวโน้มของปริมาณข้อมูลทางด้านภูมิศาสตร์มีเพิ่มมากขึ้นเรื่อยๆ โดยมีหลายหน่วยงานทั่วโลกที่ศึกษา รวบรวมและจัดการกับข้อมูลทางด้านภูมิศาสตร์ ซึ่งปัจจุบันหน่วยงานเหล่านั้นได้เริ่มมีการนำ Big Data เข้ามาใช้ในการวิเคราะห์การเปลี่ยนแปลงของสภาพภูมิอากาศแล้ว ตัวอย่างเช่น Korean Meteorological Administration (KMA) เป็นต้น และตัวอย่างจากการนำ Big Data เข้ามาช่วยในการทำนายการเปลี่ยนแปลงสภาพภูมิอากาศ ได้แก่ นักวิจัยของ IBM ได้ใช้อัลกอริทึมของพวกเขาในการวิเคราะห์ข้อมูลเพื่ออธิบายลักษณะของชั้นบรรยากาศโดนสร้างเป็นโมเดลทางคณิตศาสตร์อธิบายการเกิดของพายุในเมือง Rio de Janeiro ซึ่งเป็นพื้นที่ที่เกิดน้ำท่วมและดินถล่มบริเวณใกล้เนินเขาบ่อยครั้ง โดยสามารถวิเคราะห์ได้ล่วงหน้ากว่า 40 ชั่วโมง โดยมีความถูกต้องประมาณ 90%

นอกเหนือจากสภาพภูมิอากาศ อาทิเช่น น้ำท่วม พายุ อุณหภูมิต่างๆ ที่ได้กล่าวมาข้างต้นแล้ว ยังมีสภาพภูมิอากาศที่ใกล้ตัวและมีผลต่อการใช้ชีวิตประจำวันอีกด้วย นั่นคือ ฝน ถ้าหากเราสามารถนำ Big Data มาวิเคราะห์และทำนายล่วงหน้าได้อย่างถูกต้องแม่นยำว่าฝนจะตกในวันพรุ่งนี้หรือไม่ จะส่งผลให้ผู้คนสามารถใช้ชีวิตได้ดีและสะดวกยิ่งขึ้น ยกตัวอย่างเช่น ชาวประมงจะสามารถตัดสินใจได้ง่ายขึ้น ว่าควรนำเรือออกไปทำการประมงหรือไม่ เนื่องจากฝนก็เป็นอีกหนึ่งปัจจัยที่มีผลต่อการทำประมง หรือจะเป็นเรื่องที่ใกล้ตัวมากกว่านั้น อย่างเช่น ถ้าหากเรารู้ว่าวันนี้ฝนจะตกและจำเป็นต้องเดินทางโดยการขับขี่ยานพาหนะ จะทำให้เราสามารถระมัดระวังได้มากขึ้น ไม่ประมาทกับการขับขี่ ส่งผลให้อุบัติเหตุอาจลดน้อยลงได้นั่นเอง

จากข้างต้นเห็นได้ว่าการนำ Big Data มาวิเคราะห์สภาพภูมิอากาศ เริ่มมีบทบาทสำคัญต่อผู้คนมากยิ่งขึ้น ทางผู้จัดทำจึงได้เล็งเห็นถึงความสำคัญนี้ และได้ทำการนำข้อมูลการตกของฝนในประเทศออสเตรเลียมาทำการสร้างโมเดลเพื่อทำนายว่าในวันพรุ่งนี้ฝนจะตกหรือไม่ โดยการใช้ Google Cloud Platform (GCP) เป็นแพลตฟอร์มหลักในการดำเนินงาน ซึ่งถ้าหากเราสามารถทราบได้ล่วงหน้าว่าฝนจะตกหรือไม่จากโมเดลที่สร้างขึ้นอาจส่งผลให้ประชาชนสามารถวางแนวทางป้องกันและลดความเสียหายอุบัติเหตุ และความไม่สะดวกสบายในการใช้ชีวิตได้อย่างมีประสิทธิภาพ นอกจากนี้ยังสามารถนำโมเดลที่ได้มาประยุกต์ใช้กับข้อมูลเกี่ยวกับฝนในประเทศไทยได้อีกด้วย

กรอบแนวคิด



รูป 1 กรอบแนวคิด

เป้าหมาย

จากข้อมูลเกี่ยวกับฝนในประเทศออสเตรเลีย ผู้จัดทำมีเป้าหมายในสร้างความเข้าใจเกี่ยวกับข้อมูล สามารถบอกได้ว่าปัจจัยใดมีผลต่อการเกิดฝนตก และสร้างโมเดลที่สามารถทำนายว่า ในวันพรุ่งนี้ฝนจะตกในประเทศออสเตรเลียหรือไม่ ได้อย่างถูกต้องแม่นยำ โดยมีเกณฑ์การตัดสินใจคือ เราจะทำการเตรียมตัวรับมือกับฝนตก ถ้าหากโอกาสที่ฝนจะตกอยู่ที่ประมาณ 70%

รายละเอียดข้อมูล

ข้อมูลเกี่ยวกับน้ำฝนในประเทศออสเตรเลีย ทางผู้จัดทำได้ทำการลืบค้นมาจาก www.kaggle.com โดยข้อมูลมีชื่อว่า “Rain in Australia (Predict next-day rain in Australia)” ผู้จัดทำได้ทำการดาวน์โหลดไฟล์ข้อมูล weatherAUS.csv และได้ทำการดูข้อมูลเบื้องต้นดังนี้

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Date           145460 non-null  object 
 1   Location       145460 non-null  object 
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object 
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object 
 10  WindDir3pm     141232 non-null  object 
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object 
 22  RainTomorrow   142193 non-null  object 
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

รูป 2 ผลลัพธ์จาก JupyterLab แสดง info ของข้อมูล

Out[12]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	...
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	...
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	...
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	...
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	...
...	...	...	...	...	...	...	...	...	...	...	...
145455	2017-06-21	Uluru	2.8	23.4	0.0	NaN	NaN	E	31.0	SE	...
145456	2017-06-22	Uluru	3.6	25.3	0.0	NaN	NaN	NNW	22.0	SE	...
145457	2017-06-23	Uluru	5.4	26.9	0.0	NaN	NaN	N	37.0	SE	...
145458	2017-06-24	Uluru	7.8	27.0	0.0	NaN	NaN	SE	28.0	SSE	...
145459	2017-06-25	Uluru	14.9	NaN	0.0	NaN	NaN	NaN	NaN	ESE	...

145460 rows × 23 columns

รูป 3 ผลลัพธ์จาก JupyterLab แสดงตารางข้อมูลเบื้องต้น

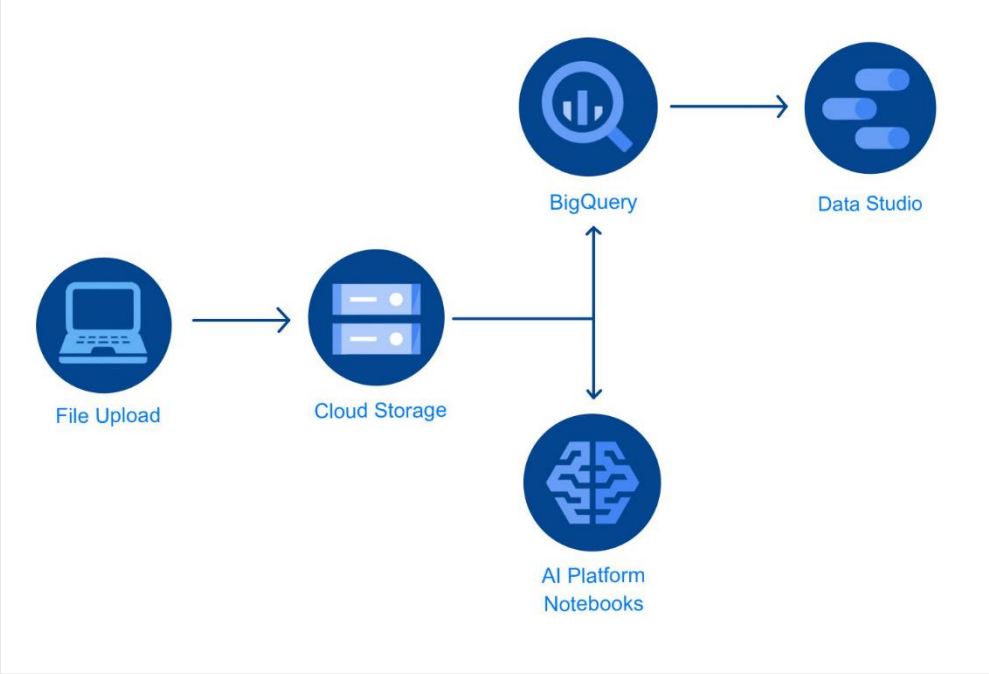
ข้อมูลมีทั้งหมด 145460 แถว 23 คอลัมน์ โดยมีรายละเอียดแต่ละคอลัมน์ดังต่อไปนี้

ตาราง 1 ตารางแสดงรายละเอียดข้อมูล

	ชื่อตัวแปร	ชนิด	คำอธิบาย	ตัวอย่าง
1.	Date	String	วันที่สังเกต	2008-12-01
2.	Location	String	ชื่อที่ตั้งของสถานีตรวจอากาศ	Albury
3.	MinTemp	Float	อุณหภูมิต่ำสุด (องศาเซลเซียส)	13.4
4.	MaxTemp	Float	อุณหภูมิสูงสุด (องศาเซลเซียส)	25.1
5.	Rainfall	Float	ปริมาณน้ำฝนที่บันทึกไว้แต่ละวัน (มิลลิเมตร)	0.6
6.	Evaporation	Float	การระเหย ณ เวลา 9.00 น. โดยใช้เครื่องมือวัดแบบ Class A pan (มิลลิเมตร)	2.8
7.	Sunshine	Float	จำนวนชั่วโมงที่มีแสงแดดในแต่ละวัน	7.6
8.	WindGustDir	String	ทิศทางลมกระโชกแรงที่สุดในช่วง 24 ชั่วโมงถึงเที่ยงคืน	NE
9.	WindGustSpeed	Float	ความเร็วของลมกระโชกแรงที่สุด (กม./ชม.)	44.0
10.	WindDir9am	String	ทิศทางของลมเวลา 9.00 น.	SW
11.	WindDir3pm	String	ทิศทางของลมเวลา 15.00 น.	E
12.	WindSpeed9am	Float	ความเร็วลม (กม./ชม.) เวลา 9.00 น.	20.0
13.	WindSpeed3pm	Float	ความเร็วลม (กม./ชม.) เวลา 15.00 น.	39.0
14.	Humidity9am	Float	ความชื้น (เปอร์เซ็นต์) เวลา 9.00 น.	91.0
15.	Humidity3pm	Float	ความชื้น (เปอร์เซ็นต์) เวลา 15.00 น.	76.0
16.	Pressure9am	Float	ความดันบรรยากาศ (hpa) เวลา 9.00 น	1018.4

	ชื่อตัวแปร	ชนิด	คำอธิบาย	ตัวอย่าง
17.	Pressure3pm	Float	ความดันบรรยากาศ (hpa) เวลา15.00 น	1015.6
18.	Cloud9am	Float	มาตราส่วนกำหนดเมฆปกคลุมเวลา 9.00 น. (มีหน่วยวัดเป็น "oktas" บันทึกว่ามีเมฆบดบังท้องฟ้ากี่จุด การวัด 0 หมายถึงท้องฟ้าปลอดโปร่งในขณะที่ตัวเลข 8 แสดงว่ามืดครึ้มอย่างสมบูรณ์)	6
19.	Cloud3pm	Float	มาตราส่วนกำหนดเมฆปกคลุมเวลา 15.00 น.	2
20.	Temp9am	Float	อุณหภูมิ (องศาเซลเซียส) เวลา 9.00 น.	24.5
21.	Temp3pm	Float	อุณหภูมิ (องศาเซลเซียส) เวลา 15.00 น.	26.1
22.	RainToday	String	ปริมาณน้ำฝน (มิลลิเมตร) หากเกิน 1 มม. จะมีค่าเท่ากับ 1 ถ้าไม่เกินเท่ากับ 0	Yes
23.	RainTomorow	String	ปริมาณฝนในวันถัดไป (มิลลิเมตร) หากเกิน 1 มม. จะมีค่าเท่ากับ 1 ถ้าไม่เกินเท่ากับ 0	No

สถาปัตยกรรมระบบเบื้องต้น



รูป 4 แผนผังแสดงสถาปัตยกรรมเบื้องต้น

1. Cloud Storage
   - ใช้ในการเก็บไฟล์ข้อมูล "weatherAUS.csv" ลงใน Bucket ชื่อว่า "cs358-finalproj" folder "data/"

2. BigQuery
   - ใช้ในการคิวรี่ข้อมูล เพื่อนำไปใช้งานต่อใน Platform อื่นๆ ในที่นี้ทำการสร้าง Dataset ชื่อว่า “Rain” และสร้าง Table ขึ้นมา 2 table เพื่อใช้ในการสร้างกราฟที่น่าสนใจ โดยมี Table ดังนี้
     1. Table ชื่อ “explore” : เป็น table ที่มี Dataframe ข้อมูลจากไฟล์ weatherAUS.csv หลังจากที่ทำการ clean ข้อมูลบางส่วนแล้ว สร้าง table นี้ขึ้นเพื่อให้สามารถนำไป explore data ที่มีเบื้องต้นได้ โดยการสร้างกราฟใน Data Studio
     2. Table ชื่อ “corr” : เป็น table เก็บค่า correlation ระหว่างตัวแปรแต่ละตัว สร้าง table นี้ขึ้นเพื่อให้สามารถนำไปสร้างกราฟใน Data Studio ได้

3. Data Studio
   - ใช้ในการสร้าง Dashboard ที่น่าสนใจประกอบการวิเคราะห์ต่างๆ ในที่นี้ได้ทำการสร้าง report ที่ชื่อว่า cs358-finalproj-explore data ใน report นี้ มีทั้งหมด 3 หน้า ประกอบไปด้วย

1. กราฟ Count of RainTomorrow : เพื่อแสดงจำนวนที่มีและรูปแบบของตัวแปรที่สนใจ (ตัวแปร Y) ในที่นี้มีค่า Yes เท่ากับ 31,877 ค่า และมีค่า No เท่ากับ 110,316 ค่า
2. กราฟ Correlation Heatmap of Rain in Australia Dataset : เพื่อแสดงความสัมพันธ์ระหว่างตัวแปรแต่ละตัว และทำให้เห็นภาพชัดขึ้นด้วยสีของ heatmap
3. กราฟ Correlation Heatmap of Rain in Australia Dataset (ต่อ) : เพื่อแสดงความสัมพันธ์ระหว่างตัวแปรแต่ละตัวที่เหลือ ที่หน้าที่ 2 แสดงไม่
4. Dashboard : เพื่อแสดงข้อมูลที่น่าสนใจที่เกี่ยวข้องกับตัวแปร RainTomorrow

สามารถดูเพิ่มเติมได้ที่ https://datastudio.google.com/s/pMh9bok5g0Y

4. Cloud AI Platform Notebooks
   - ใช้ในการสร้างโมเดล หรือการทำ Evaluation และการปรับปรุงแก้ไขต่างๆ ในที่นี่ทำการสร้าง Instance Name ชื่อว่า cs358-finalproj โดยภายใน instane นี้เมื่อเปิดผ่าน JupyterLab แล้ว จะมีไฟล์ cs358-project.ipynb ซึ่งมีทั้งหมด 15 ส่วน ดังนี้
     1. Installing dependencies : ติดตั้ง dependencies ที่จำเป็น
     2. Import Library : นำเข้าไลบรารี่ที่จำเป็น
     3. Import Data from Bucket : นำเข้าไฟล์ข้อมูล "weatherAUS.csv" จาก Bucket ชื่อว่า "cs358-finalproj" folder "data/" และทำการสร้าง Dataframe
     4. Exploratory data analysis : ทำการ explore ข้อมูลเบื้องต้น เช่น ดู info ของข้อมูล ดู descriptive statistics ของตัวแปรแต่ละตัว
     5. Check Seasonal of Data : ทำการเช็คตัวแปรแต่ละตัวว่ามี seasonal หรือไม่ เพื่อใช้ในการตัดสินใจในการเลือกวิธีจัดการกับข้อมูลในขั้นตอนการทำ Feature Engineering
     6. Univariate Analysis : ทำการวิเคราะห์ตัวแปร RainTomorrow ซึ่งเป็นตัวแปรที่เราสนใจ
     7. Bivariate Analysis : ทำการวิเคราะห์ตัวแปรต่างๆ โดยแบ่งเป็นตัวแปร Categorical และตัวแปร Numerical ทำการวิเคราะห์ข้อมูลเบื้องต้นเช่น รูปแบบข้อมูล, จำนวน missing value, outliner ในแต่ละตัวแปร และทำการหาช่วงของ outliner นั้นๆ
     8. Export Dataframe to Bigquery : นำ dataframe ที่มี export ไปยัง bigquery เพื่อนำไปใช้สร้างกราฟใน Data Studio ต่อไป
     9. Multivariate Analysis : ทำการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรแต่ละตัว โดยใช้ Correlation ในการวิเคราะห์
     10. Declare feature vector and target variable : จัดข้อมูลให้เป็น x และ y

11. Split data into separate training and test set : แบ่งข้อมูลให้เป็นชุด test และ train
12. Feature Engineering : ทำการ clean ข้อมูลโดยการจัดการกับ missing value และ ค่า outliner ที่มี และทำการสร้างตัวแปร dummy ให้แก่ตัวแปรคุณภาพ
13. Feature Scaling : ทำการ scaling data
14. Model training : ทำการสร้าง Model โดยใช้ Logistic Regression
15. Evaluation : สร้าง Confusion Metrix และทำการ Add threshold level เพื่อใช้ในการตัดสินใจว่าควรจะเลือก threshold ไหนในการตัดสินใจตามเป้าหมายที่ตั้งไว้

แผนการดำเนินงาน

ตาราง 2 ตารางแสดงแผนการดำเนินงาน

ลำดับ	กิจกรรม	เม.ย.				พ.ค.				มิ.ย.	
		1	2	3	4	1	2	3	4	1	2
1	วางแผน หาข้อมูล และเป้าหมายที่สนใจ	←									
2	ดำเนินการเพื่อให้ได้ผลเบื้องต้น เช่น การนำข้อมูลเข้าไปเก็บใน Storage	←									
3	นำเสนอเค้าโครงโปรเจค		↔								
4	ปรับปรุงแก้ไข		←	→							
5	Data Understanding			←	→						
6	Data Preparing			←	→						
7	Modeling เบื้องต้น				←	→					
8	รายงานความก้าวหน้า					↔					
9	ปรับปรุงแก้ไข					←	→				
10	Modeling (ต่อ)						←	→			
11	Evaluation							←	→		
12	ปรับปรุงแก้ไข								←	→	
13	นำเสนอผลงานโปรเจคทั้งหมด										↔

วิธีการดำเนินงานตามหลัก CRISP-DM

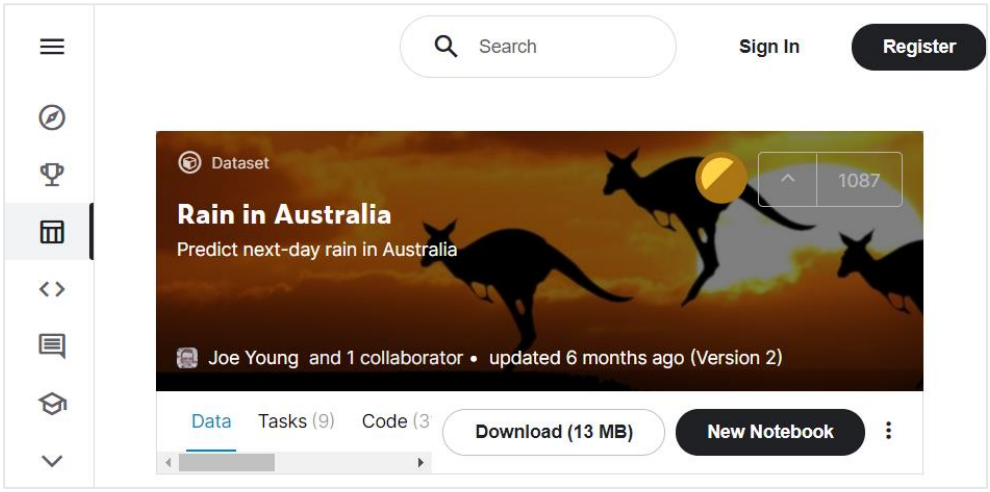
1. **Business Understanding**

การทำโปรเจคนี้จะเป็นการศึกษา Machine Learning ประเภท Supervised Learning โดยใช้ Classification Model เพื่อทำนายว่าฝนจะตกหรือไม่ในวันต่อไปในประเทศออสเตรเลีย หากเราสามารถทราบได้ล่วงหน้าว่าฝนจะตกหรือไม่จากโมเดลที่สร้างขึ้น และโมเดลนี้มีความแม่นยำเพียงพอ อาจส่งผลให้ประชาชนสามารถวางแนวทางป้องกันและลดความเสียหายอุบัติเหตุ และความไม่สะดวกสบายในการใช้ชีวิตได้อย่างมีประสิทธิภาพ

2. **Data Understanding**

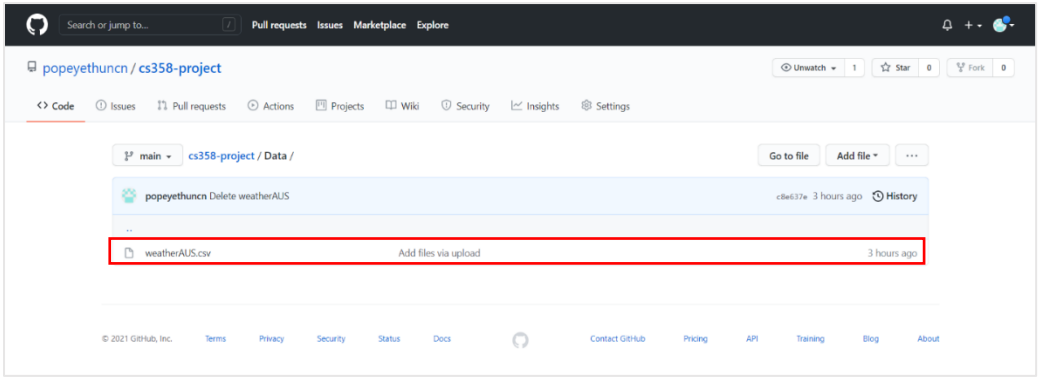
ศึกษาการทำนายการตกของฝนในวันต่อไปในประเทศออสเตรเลีย โดยศึกษาจากตัวแปร RainTomorrow เป็นหลัก โดยมีตัวแปรที่เกี่ยวข้อง เช่น RainToday, MinTemp, MaxTemp เป็นต้น ซึ่งการศึกษานี้ได้ทำการวิเคราะห์ลักษณะข้อมูลเบื้องต้น วิเคราะห์ความสัมพันธ์ ทำความเข้าใจข้อมูลว่าข้อมูลมีลักษณะที่ผิดปกติอย่างไร เช่น มี Missing Value มากน้อยเพียงใด หรือตัวแปรใดมีค่า outliner บ้าง เป็นต้น โดยมีวิธีการดังนี้

2.1 Download dataset “Rain in Australia” จาก www.kaggle.com โดยจะได้ไฟล์ข้อมูล"weatherAUS.csv" ซึ่งมีทั้งหมด 145460 แถว 23 คอลัมน์



รูป 5 หน้าเว็บไซต์ Kaggle ที่ใช้ในการ Download ข้อมูล Rain in Australia

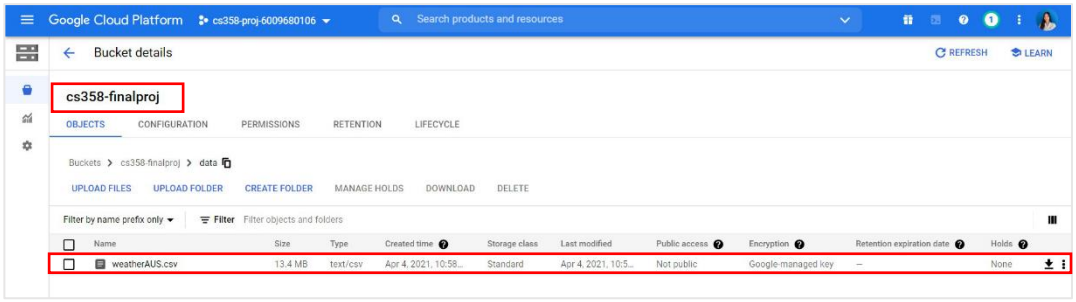
2.2 ทำการสร้าง Github เพื่อเก็บข้อมูลที่จำเป็นต้องใช้ในการทำโปรเจคนี้ทำการนำเข้าไฟล์ข้อมูล "weatherAUS.csv" ไปยัง Cloud Storage



รูป 6 หน้าเว็บไซต์ Github ที่ใช้ในการเก็บข้อมูล

https://github.com/popeyethuncn/cs358-project

2.3 การสร้าง Project ชื่อว่า “cs358-proj-6009680106” และได้ทำการสร้าง Bucket ที่มีชื่อว่า “cs358-finalproj” เพื่อเก็บไฟล์ข้อมูล weatherAUS.csv ลงใน Folder “data”



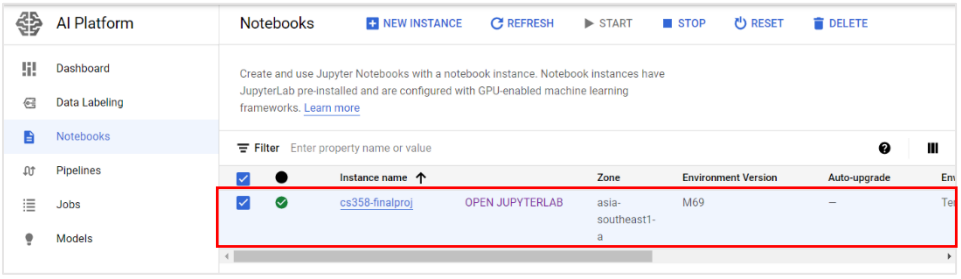
รูป 7 แสดงหน้า Bucket “cs358-finalproj”

2.4 อัพโหลดไฟล์ข้อมูล weatherAUS.csv ไปที่ Cloud Storage ผู้จัดทำได้ทำการดำเนินการ ดังนี้

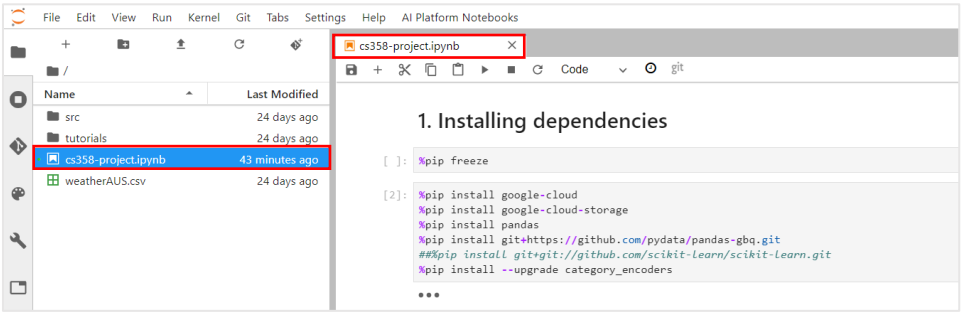
```
popeyethunchanok@cloudshell:~ (cs358-proj-6009680106)$ git clone https://github.com/popeyethuncn/cs358-project
Cloning into 'cs358-project'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (14/14), done.
popeyethunchanok@cloudshell:~ (cs358-proj-6009680106)$ ls
10  cs358-project  data-science-on-gcp  README-cloudshell.txt
popeyethunchanok@cloudshell:~ (cs358-proj-6009680106)$ cd cs358-project/
popeyethunchanok@cloudshell:~/cs358-project (cs358-proj-6009680106)$ cd Data/
popeyethunchanok@cloudshell:~/cs358-project/Data (cs358-proj-6009680106)$ ls
weatherAUS.csv
popeyethunchanok@cloudshell:~/cs358-project/Data (cs358-proj-6009680106)$ gsutil cp weatherAUS.csv gs://cs358-finalproj/data/
Copying file://weatherAUS.csv [Content-Type=text/csv]...
\ [1 files][ 13.4 MiB/ 13.4 MiB]
Operation completed over 1 objects/13.4 MiB.
```

รูป 8 แสดงหน้า Cloud Shell ที่ทำการนำเข้าไฟล์ข้อมูลลง Bucket

2.5 ผู้จัดทำได้ทำการ Enable Cloud AI Platforms Notebook และทำการสร้าง Instance ที่มีชื่อว่า cs358-finalproj โดยภายใน instane นี้เมื่อเปิดผ่าน JupyterLab แล้ว จะมีไฟล์ cs358-project.ipynb

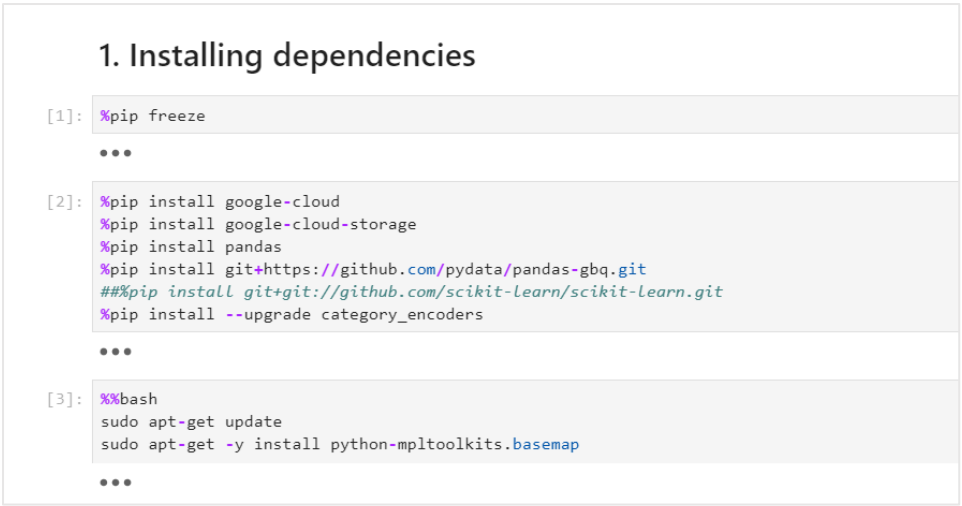


รูป 9 แสดงหน้า Cloud AI Platforms Notebook



รูป 10 แสดงหน้า JupyterLab ไฟล์ cs358-project.ipynb

2.6 ทำ Data Understanding ในไฟล์ cs358-project.ipynb โดยทำการ Installing dependencies หรือติดตั้ง dependencies ที่จำเป็น



รูป 11 แสดง code บางส่วนในขั้นตอน Installing dependencies

2.7 Import Library : นำเข้าไลบรารี่ที่จำเป็น

2. Import Library

[4]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from pandas.io import gbq
```

รูป 12 แสดง code บางส่วนในขั้นตอน Import Library

2.8 Import Data from Bucket : นำเข้าไฟล์ข้อมูล "weatherAUS.csv" จาก Bucket ชื่อว่า "cs358-finalproj" folder "data/" และทำการสร้าง Dataframe ชื่อ df

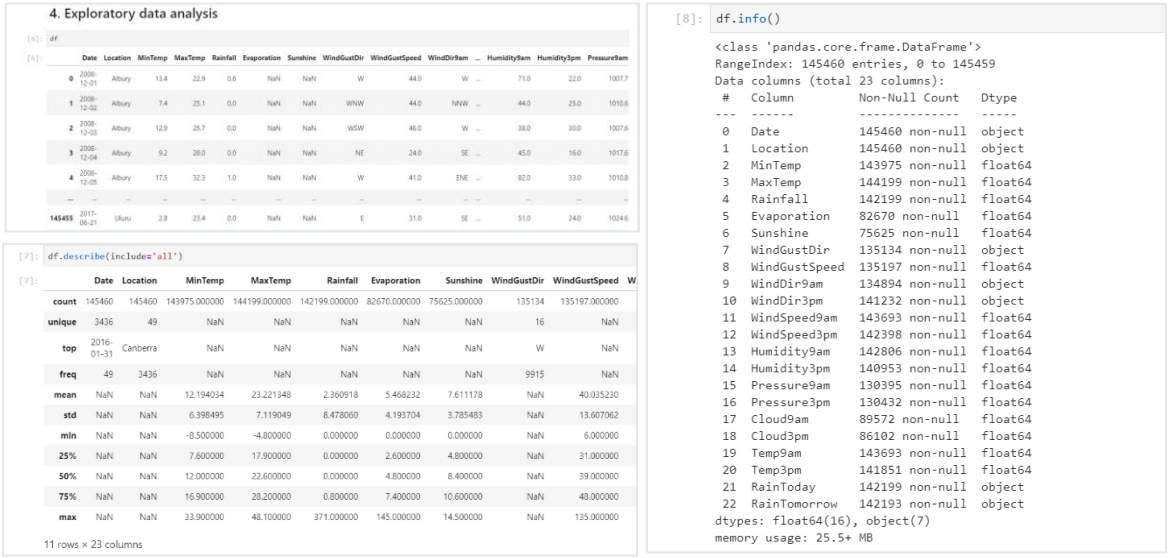
3. Import Data from Bucket

[5]:
```
from google.cloud import storage
import pandas as pd

bucket_name = "cs358-finalproj"

storage_client = storage.Client()
bucket = storage_client.get_bucket(bucket_name)

#When you have your files in a subfolder of the bucket.
my_prefix = "data/" #the name of subfolder
blobs = bucket.list_blobs(prefix = my_prefix, delimiter = '/')

for blob in blobs:
    if(blob.name != my_prefix): #ignore the subfolder itself
        file_name = blob.name.replace(my_prefix, "")
        blob.download_to_filename(file_name) #download the file to the machine
        df = pd.read_csv(file_name) #load the data
        print(df)
```

```
             Date Location  MinTemp  MaxTemp  Rainfall  Evaporation  \
0      2008-12-01   Albury     13.4     22.9       0.6          NaN
1      2008-12-02   Albury      7.4     25.1       0.0          NaN
2      2008-12-03   Albury     12.9     25.7       0.0          NaN
3      2008-12-04   Albury      9.2     28.0       0.0          NaN
4      2008-12-05   Albury     17.5     32.3       1.0          NaN
...           ...      ...      ...      ...       ...          ...
145455 2017-06-21    Uluru      2.8     23.4       0.0          NaN
145456 2017-06-22    Uluru      3.6     25.3       0.0          NaN
145457 2017-06-23    Uluru      5.4     26.9       0.0          NaN
145458 2017-06-24    Uluru      7.8     27.0       0.0          NaN
145459 2017-06-25    Uluru     14.9      NaN       0.0          NaN
```

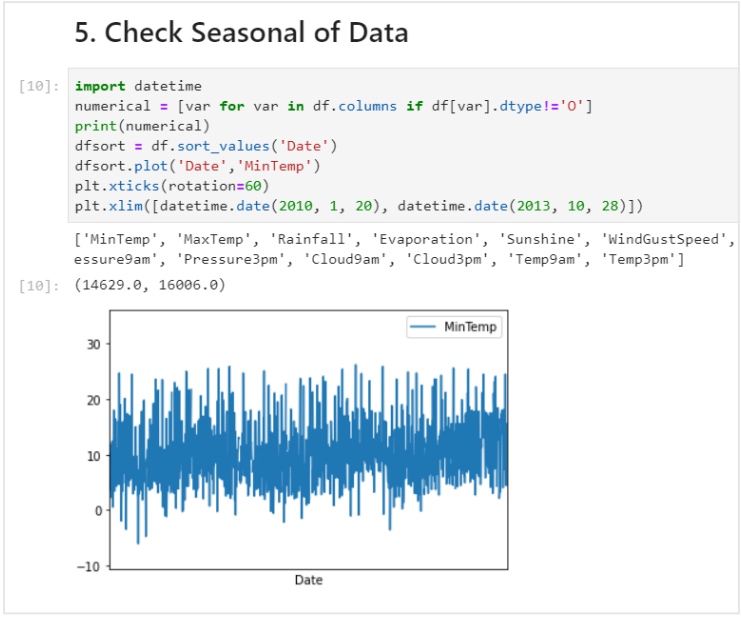
รูป 13 แสดง code บางส่วนในขั้นตอน Import Data from Bucket

2.9 Exploratory data analysis : ทำการ explore ข้อมูลเบื้องต้น เช่น ดู info ของข้อมูล ดู descriptive statistics ของตัวแปรแต่ละตัว โดยมีตัวแปรที่เป็น object ทั้งหมด 7 ตัว และตัวแปร Float 16 ตัว



รูป 14 แสดง code บางส่วนในขั้นตอน Exploratory data analysis

2.10 Check Seasonal of Data : ทำการเช็คตัวแปรแต่ละตัวว่ามี seasonal หรือไม่ เพื่อใช้ในการตัดสินใจในการเลือกวิธีจัดการกับข้อมูลในขั้นตอนการทำ Feature Engineering



รูป 15 แสดง code บางส่วนในขั้นตอน Check Seasonal of Data

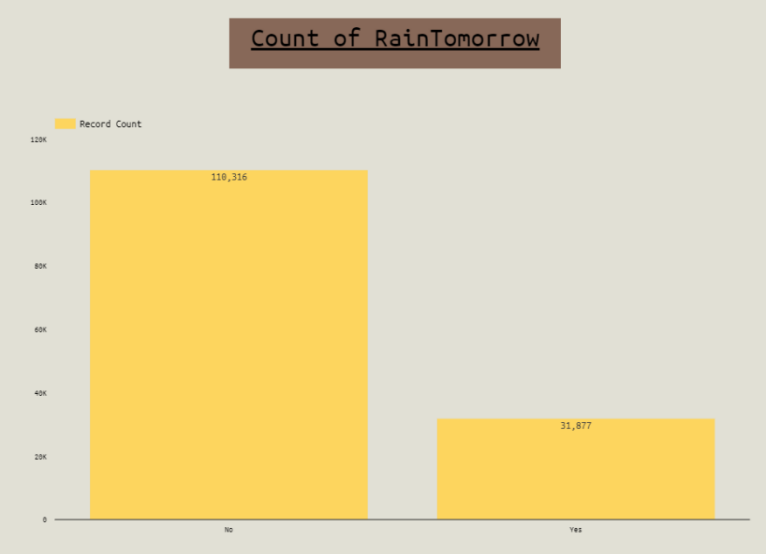
2.11 Univariate Analysis : ทำการวิเคราะห์ตัวแปร RainTomorrow ซึ่งเป็นตัวแปรที่เราสนใจ โดยมีค่า Missing Value เท่ากับ 3267 ค่า Yes 31877 ค่า และ No 110316 ค่า

6. Univariate Analysis

Explore RainTomorrow target variable

```
[10]: ###Univariate Analysis###

      df['RainTomorrow'].isnull().sum()

[10]: 3267

[11]: df['RainTomorrow'].unique()

[11]: array(['No', 'Yes', nan], dtype=object)

[12]: #count
      df['RainTomorrow'].value_counts()

[12]: No     110316
      Yes     31877
      Name: RainTomorrow, dtype: int64
```

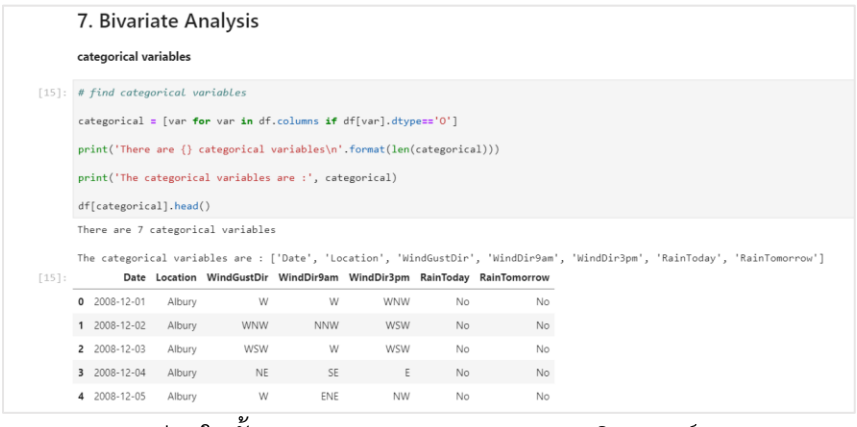
รูป 16 แสดง code บางส่วนในขั้นตอน Univariate Analysis



รูป 17 กราฟแสดง Count of RainTomorrow ที่ได้จาก Data Studio

ในที่นี้ได้ทำการ plot graph ใน Data Studio เพื่อให้เห็นภาพของตัวแปร RainTomorrow ชัดขึ้น จากรูปที่ 17 จะเห็นว่าตัวแปร RainTomorrow มีค่า No มากกว่า Yes อยู่พอสมควร

2.12 Bivariate Analysis : ทำการวิเคราะห์ตัวแปรต่างๆ โดยแบ่งเป็นตัวแปร Categorical และตัวแปร Numerical ทำการวิเคราะห์ข้อมูลเบื้องต้นเช่น รูปแบบข้อมูล, จำนวน missing value, outliner ในแต่ละตัวแปร และทำการหาช่วงของ outliner นั้นๆ



7. Bivariate Analysis

categorical variables

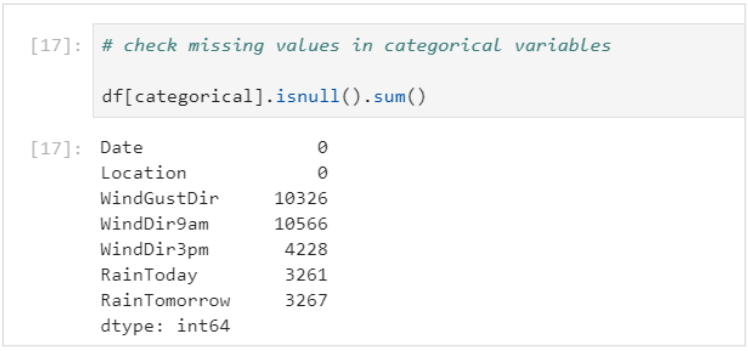
```
[15]: # find categorical variables

categorical = [var for var in df.columns if df[var].dtype=='O']

print('There are {} categorical variables\n'.format(len(categorical)))

print('The categorical variables are :', categorical)

df[categorical].head()
```

```
There are 7 categorical variables

The categorical variables are : ['Date', 'Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday', 'RainTomorrow']
```

[15]:

	Date	Location	WindGustDir	WindDir9am	WindDir3pm	RainToday	RainTomorrow
0	2008-12-01	Albury	W	W	WNW	No	No
1	2008-12-02	Albury	WNW	NNW	WSW	No	No
2	2008-12-03	Albury	WSW	W	WSW	No	No
3	2008-12-04	Albury	NE	SE	E	No	No
4	2008-12-05	Albury	W	ENE	NW	No	No

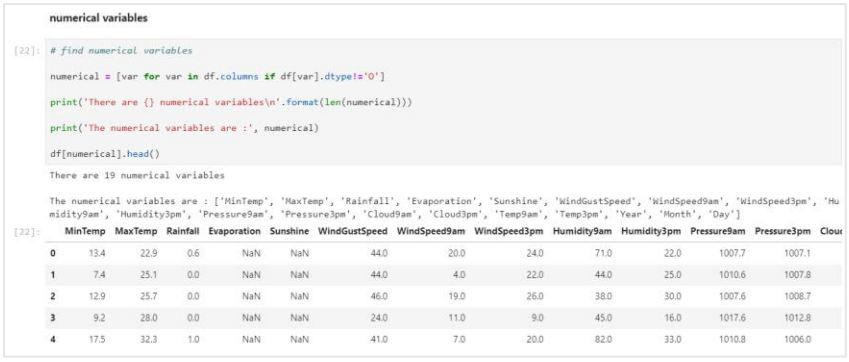
รูป 18 แสดง code บางส่วนในขั้นตอน Bivariate Analysis การวิเคราะห์ Categorical Variables



```
[17]: # check missing values in categorical variables

      df[categorical].isnull().sum()
```

```
[17]: Date              0
      Location          0
      WindGustDir     10326
      WindDir9am      10566
      WindDir3pm       4228
      RainToday        3261
      RainTomorrow     3267
      dtype: int64
```

รูป 19 แสดง code ในการตรวจสอบค่า Missing Value ของตัวแปร Categorical



numerical variables

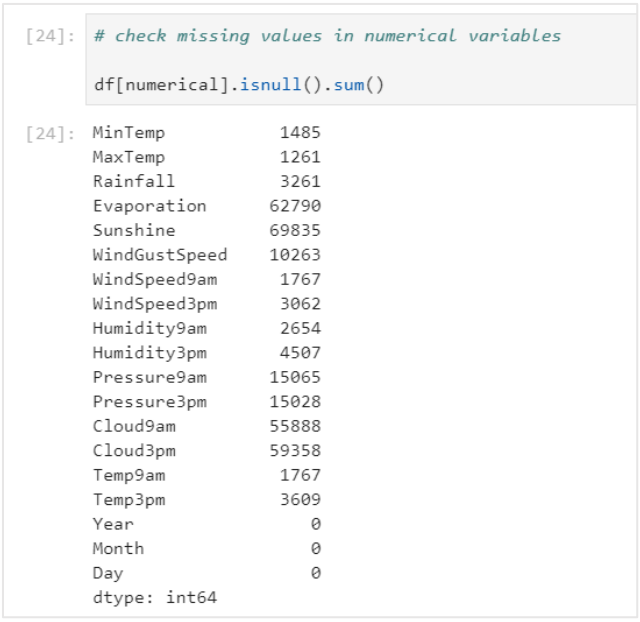
```
[22]: # find numerical variables

numerical = [var for var in df.columns if df[var].dtype!='O']

print('There are {} numerical variables\n'.format(len(numerical)))

print('The numerical variables are :', numerical)

df[numerical].head()
```

```
There are 19 numerical variables

The numerical variables are : ['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Hu
midity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm', 'Year', 'Month', 'Day']
```

[22]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Clou
0	13.4	22.9	0.6	NaN	NaN	44.0	20.0	24.0	71.0	22.0	1007.7	1007.1	
1	7.4	25.1	0.0	NaN	NaN	44.0	4.0	22.0	44.0	25.0	1010.6	1007.8	
2	12.9	25.7	0.0	NaN	NaN	46.0	19.0	26.0	38.0	30.0	1007.6	1008.7	
3	9.2	28.0	0.0	NaN	NaN	24.0	11.0	9.0	45.0	16.0	1017.6	1012.8	
4	17.5	32.3	1.0	NaN	NaN	41.0	7.0	20.0	82.0	33.0	1010.8	1006.0	

รูป 20 แสดง code บางส่วนในขั้นตอน Bivariate Analysisการวิเคราะห์ Numerical Variables



```
[24]: # check missing values in numerical variables

      df[numerical].isnull().sum()

[24]: MinTemp          1485
      MaxTemp          1261
      Rainfall         3261
      Evaporation     62790
      Sunshine        69835
      WindGustSpeed   10263
      WindSpeed9am     1767
      WindSpeed3pm     3062
      Humidity9am      2654
      Humidity3pm      4507
      Pressure9am     15065
      Pressure3pm     15028
      Cloud9am        55888
      Cloud3pm        59358
      Temp9am          1767
      Temp3pm          3609
      Year                0
      Month               0
      Day                 0
      dtype: int64
```

รูป 21 แสดง code ในการตรวจสอบค่า Missing Value ของตัวแปร Numerical

จากรูปที่ 19 จะเห็นว่ามีตัวแปร WindGustDir, WindDir9am, WindDir3pm, RainToday และ RainTomorrow มีค่า Missing เท่ากับ 10326, 10566, 4228, 3261 และ 3267 ตามลำดับ ในขณะที่รูปที่ 21 จะเห็นว่ามีตัวแปร MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am และ Temp3pm มีค่า Missing เท่ากับ 1485, 1261, 3261, 62790, 69835, 10263, 1767, 3062, 2654, 4507, 15065, 15028, 55888, 59358, 1767 และ 3609 ตามลำดับ

หลังจากนั้นได้ทำการแปลง field “Date” ให้แยกเป็น Date Month Year เพื่อลดค่า label ใน field “Date” เนื่องจาก Label จำนวนมากภายในตัวแปร จะเรียกว่ามีค่า cardinality ที่สูง ซึ่งอาจก่อให้เกิดปัญหาในโมเดลแมชชีนเลิร์นนิ่งได้ ดังนั้นเมื่อทำการแปลงเสร็จแล้วจึงทำการ drop “Date” อันเก่าลง

```
[17]: # check for cardinality in categorical variables

      for var in categorical:

          print(var, ' contains ', len(df[var].unique()), ' labels')
```

```
Date  contains  3436  labels
Location  contains  49  labels
WindGustDir  contains  17  labels
WindDir9am  contains  17  labels
WindDir3pm  contains  17  labels
RainToday  contains  3  labels
RainTomorrow  contains  3  labels
```

รูปที่ 21 แสดง code ในการเช็คค่า Cardinality

```
[18]: #Feature Engineering of Date Variable

      # parse the dates, currently coded as strings, into datetime format

      df['Date'] = pd.to_datetime(df['Date'])

      # extract year from date

      df['Year'] = df['Date'].dt.year

      # extract month from date

      df['Month'] = df['Date'].dt.month

      # extract day from date

      df['Day'] = df['Date'].dt.day
```

รูป 22 แสดง code ในการ Feature Engineering ของตัวแปร Date

```
[20]: # drop the original Date variable

      df.drop('Date', axis=1, inplace = True)

      df.head()
```

[20]:

e	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	...	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow	Year	Month	Day
N	W	44.0	W	WNW	...	1007.1	8.0	NaN	16.9	21.8	No	No	2008	12	1
N	WNW	44.0	NNW	WSW	...	1007.8	NaN	NaN	17.2	24.3	No	No	2008	12	2
N	WSW	46.0	W	WSW	...	1008.7	NaN	2.0	21.0	23.2	No	No	2008	12	3
N	NE	24.0	SE	E	...	1012.8	NaN	NaN	18.1	26.5	No	No	2008	12	4
N	W	41.0	ENE	NW	...	1006.0	7.0	8.0	17.8	29.7	No	No	2008	12	5

รูป 23 แสดง code ในการ Drop ตัวแปร Date เก่าออก

นอกจากนี้ผู้จัดทำได้ทำการตรวจสอบ Outliner ในแต่ละตัวตัวแปรโดยเริ่มดูจากตาราง Descriptive Statistics

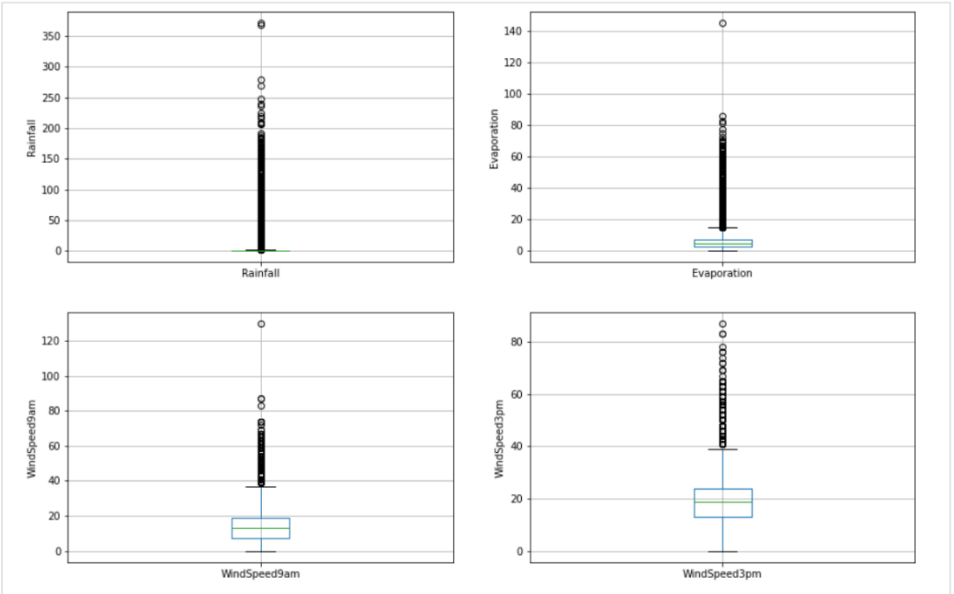
```
[25]: #Outliers in numerical variables
      #Rainfall, Evaporation, WindSpeed9am, WindSpeed3pm

      print(round(df[numerical].describe()),2)
```

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	\
count	143975.0	144199.0	142199.0	82670.0	75625.0	135197.0	
mean	12.0	23.0	2.0	5.0	8.0	40.0	
std	6.0	7.0	8.0	4.0	4.0	14.0	
min	-8.0	-5.0	0.0	0.0	0.0	6.0	
25%	8.0	18.0	0.0	3.0	5.0	31.0	
50%	12.0	23.0	0.0	5.0	8.0	39.0	
75%	17.0	28.0	1.0	7.0	11.0	48.0	
max	34.0	48.0	371.0	145.0	14.0	135.0	

	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	\
count	143693.0	142398.0	142806.0	140953.0	130395.0	
mean	14.0	19.0	69.0	52.0	1018.0	
std	9.0	9.0	19.0	21.0	7.0	
min	0.0	0.0	0.0	0.0	980.0	
25%	7.0	13.0	57.0	37.0	1013.0	
50%	13.0	19.0	70.0	52.0	1018.0	
75%	19.0	24.0	83.0	66.0	1022.0	
max	130.0	87.0	100.0	100.0	1041.0	

รูป 24 แสดง code ในตรวจสอบค่า Outliner

จากรูปที่ 24 จะเห็นว่า ตัวแปร Rainfall Evaporation WindSpeed9am และ WindSpeed3pm มีค่า Mean ที่ค่อนข้างใกล้กับค่า Min ไม่ได้อยู่ตรงกลางระหว่างค่า Min และ Max แสดงให้เห็นว่าทั้ง 4 ตัวแปรนี้ น่าจะมีค่า Outliner จึงทำการตรวจสอบ Box Plot เพื่อให้แน่ใจว่าตัวแปรดังกล่าวมีค่า Outliner



รูป 25 Box Plot ตรวจสอบ Outliner

จากรูปที่ 25 จะเห็นว่าตัวแปรทั้ง 4 ตัวมีค่า Outliner จริงๆ ดังนั้นจึงทำการหาขอบเขตของ Outliner เพื่อที่จะได้ทำการจัดการกับค่าเหล่านั้นต่อไป

```
[28]: # find outliers for Rainfall variable

IQR = df.Rainfall.quantile(0.75) - df.Rainfall.quantile(0.25)
Lower_fence = df.Rainfall.quantile(0.25) - (IQR * 3)
Upper_fence = df.Rainfall.quantile(0.75) + (IQR * 3)
print('Rainfall outliers are values < {lowerboundary} or > {upperboundary}'.format(lowerboundary=Lower_fence, upperboundary=Upper_fence))

# find outliers for Evaporation variable

IQR = df.Evaporation.quantile(0.75) - df.Evaporation.quantile(0.25)
Lower_fence = df.Evaporation.quantile(0.25) - (IQR * 3)
Upper_fence = df.Evaporation.quantile(0.75) + (IQR * 3)
print('Evaporation outliers are values < {lowerboundary} or > {upperboundary}'.format(lowerboundary=Lower_fence, upperboundary=Upper_fence))

# find outliers for WindSpeed9am variable

IQR = df.WindSpeed9am.quantile(0.75) - df.WindSpeed9am.quantile(0.25)
Lower_fence = df.WindSpeed9am.quantile(0.25) - (IQR * 3)
Upper_fence = df.WindSpeed9am.quantile(0.75) + (IQR * 3)
print('WindSpeed9am outliers are values < {lowerboundary} or > {upperboundary}'.format(lowerboundary=Lower_fence, upperboundary=Upper_fence))

# find outliers for WindSpeed3pm variable

IQR = df.WindSpeed3pm.quantile(0.75) - df.WindSpeed3pm.quantile(0.25)
Lower_fence = df.WindSpeed3pm.quantile(0.25) - (IQR * 3)
Upper_fence = df.WindSpeed3pm.quantile(0.75) + (IQR * 3)
print('WindSpeed3pm outliers are values < {lowerboundary} or > {upperboundary}'.format(lowerboundary=Lower_fence, upperboundary=Upper_fence))

Rainfall outliers are values < -2.4000000000000004 or > 3.2
Evaporation outliers are values < -11.800000000000002 or > 21.800000000000004
WindSpeed9am outliers are values < -29.0 or > 55.0
WindSpeed3pm outliers are values < -20.0 or > 57.0
```

รูป 26 Code แสดงการหาขอบเขต Outliner

จากรูป 26 จะได้ว่า ตัวแปร Rainfall มีขอบเขตเท่ากับ (-2.4 , 3.2) ตัวแปร Evaporation มีขอบเขตเท่ากับ (-11.8 , 21.8) ตัวแปร WindSpeed9am มีขอบเขตเท่ากับ (-29.0, 55.0) และตัวแปร WindSpeed3pm มีขอบเขตเท่ากับ (-20.0, 57.0) ซึ่งถ้าหากมีค่าไหนที่เกินจากขอบเขตเหล่านี้ เราจะสามารถสรุปได้ว่าค่าเหล่าเป็น Outliner

2.13 Export Dataframe to Bigquery : นำ dataframe หลังจากที่ทำการแปลง Date แล้ว export ไปยัง BigQuery เพื่อนำไปใช้สร้างกราฟใน Data Studio ต่อไป

8. Export Dataframe to Bigquery

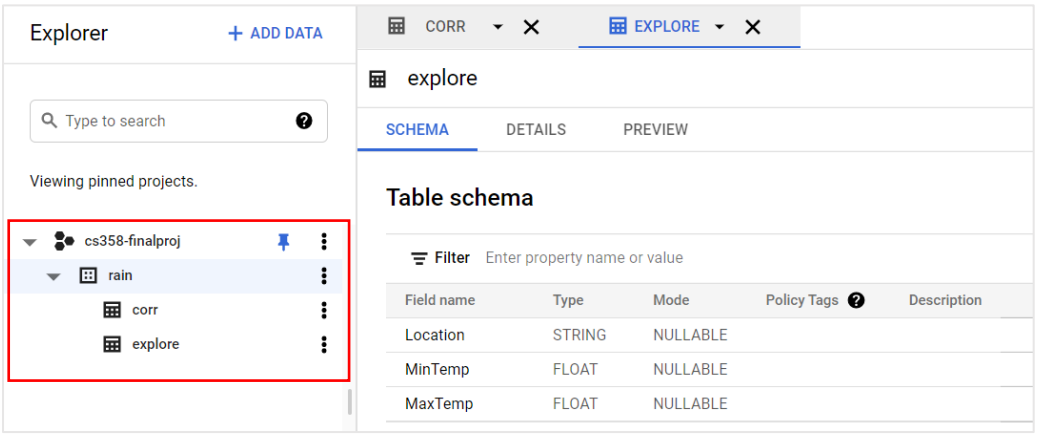
for plot graph in Data Studio

```
[34]: #table explore
df.to_gbq(destination_table='rain.explore',project_id='cs358-finalproj',if_exists='replace')

1it [00:19, 19.68s/it]

[35]: #table correlation
name = list(correlation.columns)
correlation['Name']= name
correlation.to_gbq(destination_table='rain.corr',project_id='cs358-finalproj',if_exists='replace')

1it [00:04,  4.73s/it]
```

รูป 27 แสดง code บางส่วนในการนำออก dataframe ไปยัง BigQuery

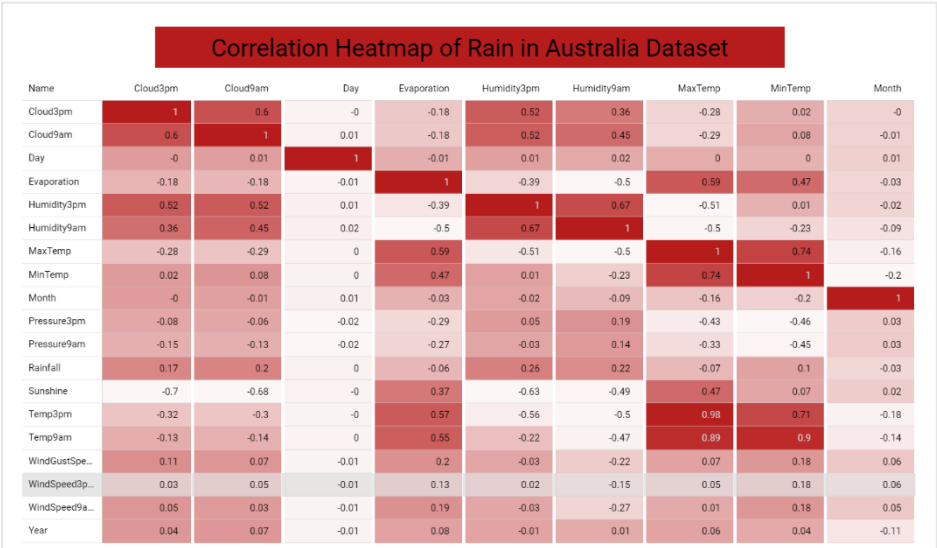
ภายใน BigQuery จะแสดงผลดังนี้ โดยมี Table ดังนี้

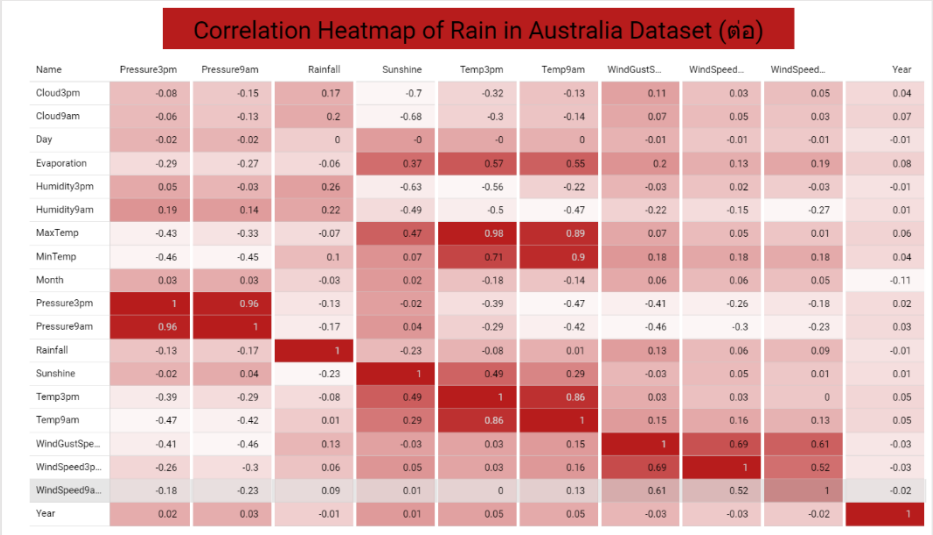
1. Table ชื่อ “explore” : เป็น table ที่มี Dataframe ข้อมูลจากไฟล์ weatherAUS.csv หลังจากที่ทำการ clean ข้อมูลบางส่วนแล้ว สร้าง table นี้ขึ้นเพื่อให้สามารถนำไป explore data ที่มีเบื้องต้นได้ โดยการสร้างกราฟใน Data Studio
2. Table ชื่อ “corr” : เป็น table เก็บค่า correlation ระหว่างตัวแปรแต่ละตัว สร้าง table นี้ขึ้นเพื่อให้สามารถนำไปสร้างกราฟใน Data Studio ได้



รูป 28 แสดงหน้า Table ใน BigQuery

2.14 Multivariate Analysis : ทำการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรแต่ละตัว โดยใช้ Correlation ในการวิเคราะห์ โดยจะทำการสร้างกราฟ Heatmap ภายใน Data Studio และทำการสร้าง Pair Plot ภายใน JupyterLab





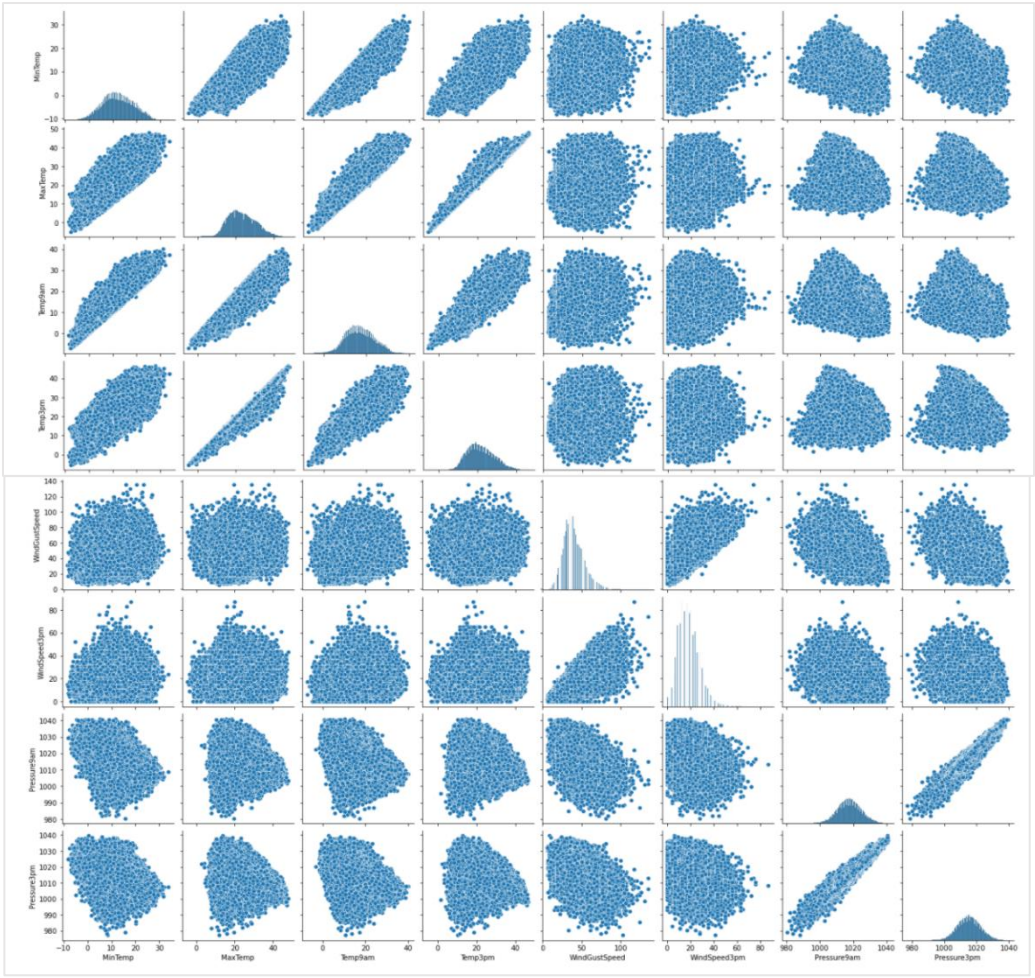
Name	Pressure3pm	Pressure9am	Rainfall	Sunshine	Temp3pm	Temp9am	WindGustS...	WindSpeed...	WindSpeed...	Year
Cloud3pm	-0.08	-0.15	0.17	-0.7	-0.32	-0.13	0.11	0.03	0.05	0.04
Cloud9am	-0.06	-0.13	0.2	-0.68	-0.3	-0.14	0.07	0.05	0.03	0.07
Day	-0.02	-0.02	0	-0	-0	0	-0.01	-0.01	-0.01	-0.01
Evaporation	-0.29	-0.27	-0.06	0.37	0.57	0.55	0.2	0.13	0.19	0.08
Humidity3pm	0.05	-0.03	0.26	-0.63	-0.56	-0.22	-0.03	0.02	-0.03	-0.01
Humidity9am	0.19	0.14	0.22	-0.49	-0.5	-0.47	-0.22	-0.15	-0.27	0.01
MaxTemp	-0.43	-0.33	-0.07	0.47	0.98	0.89	0.07	0.05	0.01	0.06
MinTemp	-0.46	-0.45	0.1	0.07	0.71	0.9	0.18	0.18	0.18	0.04
Month	0.03	0.03	-0.03	0.02	-0.18	-0.14	0.06	0.06	0.05	-0.11
Pressure3pm	1	0.96	-0.13	-0.02	-0.39	-0.47	-0.41	-0.26	-0.18	0.02
Pressure9am	0.96	1	-0.17	0.04	-0.29	-0.42	-0.46	-0.3	-0.23	0.03
Rainfall	-0.13	-0.17	1	-0.23	-0.08	0.01	0.13	0.06	0.09	-0.01
Sunshine	-0.02	0.04	-0.23	1	0.49	0.29	-0.03	0.05	0.01	0.01
Temp3pm	-0.39	-0.29	-0.08	0.49	1	0.86	0.03	0.03	0	0.05
Temp9am	-0.47	-0.42	0.01	0.29	0.86	1	0.15	0.16	0.13	0.05
WindGustSpe...	-0.41	-0.46	0.13	-0.03	0.03	0.15	1	0.69	0.61	-0.03
WindSpeed3p...	-0.26	-0.3	0.06	0.05	0.03	0.16	0.69	1	0.52	-0.03
WindSpeed9a...	-0.18	-0.23	0.09	0.01	0	0.13	0.61	0.52	1	-0.02
Year	0.02	0.03	-0.01	0.01	0.05	0.05	-0.03	-0.03	-0.02	1

รูป 29 กราฟ Heatmap ที่ได้จาก Data Studio

จากรูปข้างต้นแสดงความสัมพันธ์ระหว่างตัวแปรแต่ละตัว โดยสามารถดูได้จากค่า Correlation ยิ่งถ้ามีค่าเข้าใกล้ -1 ยิ่งมีความสัมพันธ์กันสูงในทิศทางตรงกันข้าม หรือถ้ามีค่าเข้าใกล้ 1 แสดงว่ามีความสัมพันธ์กันสูงในทิศทางเดียวกัน หากค่าเข้าใกล้ 0 แสดงว่ามีความสัมพันธ์กันน้อย หรืออีกวิธีหนึ่งที่สามารถดูได้คือ ดูจากสีของกราฟถ้าหากช่องใดมีสีเข้มนั่นหมายถึง ตัวแปรทั้งสองตัวนั้นมีความสัมพันธ์ต่อกันสูงนั่นเอง โดยตัวแปรที่มีความสัมพันธ์ต่อกันสูงมีดังนี้

- MinTemp และ MaxTemp มีค่า correlation เท่ากับ 0.74
- MinTemp และ Temp3pm มีค่า correlation เท่ากับ 0.71
- MinTemp และ Temp9am มีค่า correlation เท่ากับ 0.90
- MaxTemp และ Temp9am มีค่า correlation เท่ากับ 0.89
- MaxTemp และ Temp3pm มีค่า correlation เท่ากับ 0.98
- WindGustSpeed และ WindSpeed3pm มีค่า correlation เท่ากับ 0.69
- Pressure9am และ Pressure3pm มีค่า correlation เท่ากับ 0.96
- Temp9am และ Temp3pm มีค่า correlation เท่ากับ 0.86

นอกจากนี้ยังสามารถดูได้จากกราฟ Pair Plot ที่แสดงกราฟ Scatter Plot ระหว่างตัวแปรแต่ละตัวที่มีความสัมพันธ์กันสูงดังที่กล่าวไปข้างต้น กราฟนี้จะช่วยให้เห็นภาพมากขึ้นว่าแต่ละตัวมีความสัมพันธ์กันในทิศทางใด



รูป 30 Pair Plot แสดงแผนภาพการกระจายระหว่างตัวแปรที่มีความสัมพันธ์กันสูง

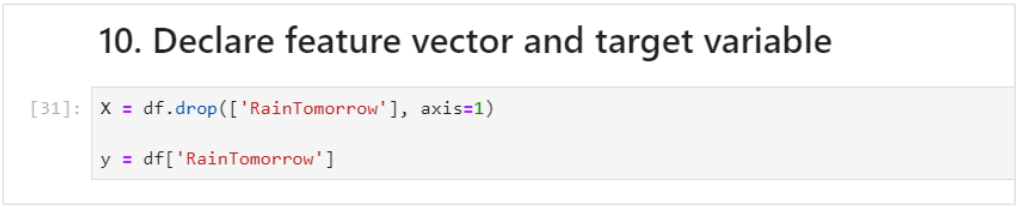
จากกราฟ Pair Plot ข้างต้นจะเห็นชัดว่ามีตัวแปรที่มีความสัมพันธ์กันสูงในทิศทางเดียวกัน ดังนี้

- MinTemp และ MaxTemp
- MinTemp และ Temp9am
- MinTemp และ Temp3pm
- MaxTemp และ Temp9am
- MaxTemp และ Temp3pm
- Temp9am และ Temp3pm
- Pressure9am และ Pressure3pm

3. Data Preparation

ทำการ Feature Engineering โดยการจัดการกับ Missing Value หรือ Outliner ให้ถูกวิธี และทำการ Feature Scaling เพื่อปรับให้ข้อมูลแต่ละตัวแปรมีลักษณะที่ใกล้เคียงกัน โดยมีขั้นตอนดังนี้

3.1 Declare feature vector and target variable : จัดข้อมูลให้ X เป็น dataframe ที่ประกอบไปด้วยตัวแปรอิสระ และ y เป็น dataframe ที่ประกอบไปด้วยตัวแปรตามหรือ rainTomorrow



รูป 31 แสดงหน้า code ในขั้นตอน Declare feature vector and target variable

3.2 Split data into separate training and test set : แบ่งข้อมูลให้เป็นชุด test และ train ใช้อัตราส่วน 20:80

รูป 32 แสดงหน้า code ในขั้นตอนการ Split data

3.3 Feature Engineering : ทำการ clean ข้อมูลโดยแบ่งเป็นการจัดการกับตัวแปร Categorical และตัวแปรNumerical โดยทำการจัดการกับ missing value และ ค่า outliner ที่มี และทำการสร้างตัวแปร dummy ให้แก่ตัวแปรคุณภาพ

12. Feature Engineering

```
[35]: # display categorical variables

      categorical = [col for col in X_train.columns if X_train[col].dtypes == 'O']

      categorical

[35]: ['Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday']

[36]: # display numerical variables

      numerical = [col for col in X_train.columns if X_train[col].dtypes != 'O']

      numerical

[36]: ['MinTemp',
       'MaxTemp',
       'Rainfall',
       'Evaporation',
       'Sunshine',
       'WindGustSpeed',
       'WindSpeed9am',
       'WindSpeed3pm',
       'Humidity9am',
       'Humidity3pm',
       'Pressure9am',
       'Pressure3pm',
       'Cloud9am',
       'Cloud3pm',
       'Temp9am',
       'Temp3pm',
       'Year',
       'Month',
```

รูป 33 แสดงหน้า code บางส่วนในขั้นตอน Feature Engineering

- การจัดการกับ Missing Value ของตัวแปร Categorical : ทำการแทนค่า missing ใน X_train, X_test, y_train และ y_test ด้วย Mode ของ train dataset

```
[40]: # impute missing categorical variables with most frequent value

for df1 in [X_train, X_test]:
    df1['WindGustDir'].fillna(X_train['WindGustDir'].mode()[0], inplace=True)
    df1['WindDir9am'].fillna(X_train['WindDir9am'].mode()[0], inplace=True)
    df1['WindDir3pm'].fillna(X_train['WindDir3pm'].mode()[0], inplace=True)
    df1['RainToday'].fillna(X_train['RainToday'].mode()[0], inplace=True)

for df1_2 in [X_train2, X_test2]:
    df1_2['WindGustDir'].fillna(X_train2['WindGustDir'].mode()[0], inplace=True)
    df1_2['WindDir9am'].fillna(X_train2['WindDir9am'].mode()[0], inplace=True)
    df1_2['WindDir3pm'].fillna(X_train2['WindDir3pm'].mode()[0], inplace=True)
    df1_2['RainToday'].fillna(X_train2['RainToday'].mode()[0], inplace=True)
```

```
[45]: # impute missing categorical variables with most frequent value

for df2 in [y_train, y_test]:
    df2.fillna(y_train.mode()[0], inplace=True)
```

รูป 34 แสดงหน้า code ในการจัดการกับ Missing Value ของตัวแปร Categorical

- การจัดการกับ Missing Value ของตัวแปร Numerical : ทำการแทนค่า missing ใน X_train และ X_test ด้วย Median ของ train dataset

```
[51]: # impute missing values in X_train and X_test with respective column median in X_train

for df3 in [X_train, X_test]:
    for col in numerical:
        col_median=X_train[col].median()
        df3[col].fillna(col_median, inplace=True)
```

รูป 35 แสดงหน้า code ในการจัดการกับ Missing Value ของตัวแปร Numerical

- การจัดการกับ outliner ของตัวแปร Numerical : โดยค่าที่นำมาใช้เป็น min และ max value หรือขอบเขต outliner นั้น นำมาจากขั้นตอนที่ 7. Bivariate Analysis ที่ได้ทำการหาขอบเขตของ outliner ไว้แล้ว

Engineering outliers in numerical variables

```
[58]: def max_value(df3, variable, top):
          return np.where(df3[variable]>top, top, df3[variable])

      for df3 in [X_train, X_test]:
          df3['Rainfall'] = max_value(df3, 'Rainfall', 3.2)
          df3['Evaporation'] = max_value(df3, 'Evaporation', 21.8)
          df3['WindSpeed9am'] = max_value(df3, 'WindSpeed9am', 55)
          df3['WindSpeed3pm'] = max_value(df3, 'WindSpeed3pm', 57)
```

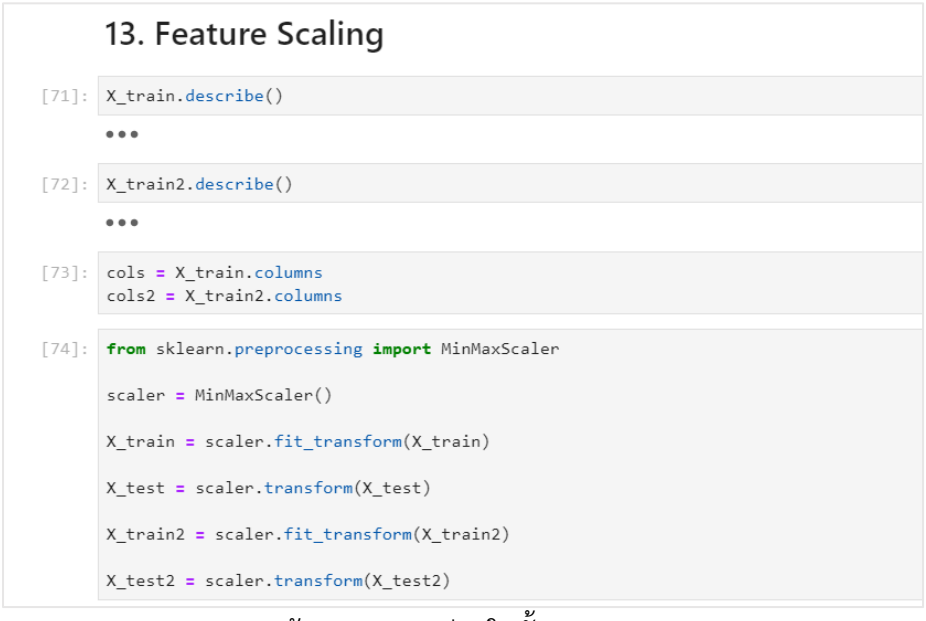
รูป 36 แสดงหน้า code ในการจัดการกับ outliner ของตัวแปร Numerical

- ทำการ encode และสร้างตัวแปร Dummy : ทำการ encode ให้กับตัวแปร RainToday และสร้าง Dummy ให้กับตัวแปรคุณภาพที่เหลือ

```
[65]: X_train = pd.concat([X_train[numerical], X_train[['RainToday_0', 'RainToday_1']],
                           pd.get_dummies(X_train.Location),
                           pd.get_dummies(X_train.WindGustDir),
                           pd.get_dummies(X_train.WindDir9am),
                           pd.get_dummies(X_train.WindDir3pm)], axis=1)

      X_train2 = pd.concat([X_train2[numerical], X_train2[['RainToday_0', 'RainToday_1']],
                           pd.get_dummies(X_train2.Location),
                           pd.get_dummies(X_train2.WindGustDir),
                           pd.get_dummies(X_train2.WindDir9am),
                           pd.get_dummies(X_train2.WindDir3pm)], axis=1)
```

รูป 37 แสดงหน้า code บางส่วนในการ encode และสร้างตัวแปร Dummy

3.4 Feature Scaling : ทำการ scaling data โดยใช้วิธี MinMaxScaler และสุดท้ายข้อมูลจะมีค่า min เท่ากับ 0 และ Max เท่ากับ 1



13. Feature Scaling

[71]:
```
X_train.describe()
```
...

[72]:
```
X_train2.describe()
```
...

[73]:
```
cols = X_train.columns
cols2 = X_train2.columns
```

[74]:
```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)

X_train2 = scaler.fit_transform(X_train2)

X_test2 = scaler.transform(X_test2)
```

รูป 38 แสดงหน้า code บางส่วนในขั้นตอน Feature Scaling

4. Modeling

ทำการ train model โดยใช้ Logistic Regression โดยผลที่ได้จากโมเดลจะเป็นการทำนายค่าของตัวแปร RainTomorrow ว่าจะมีค่าเป็น Yes หรือ No นั่นเอง

14. Model training

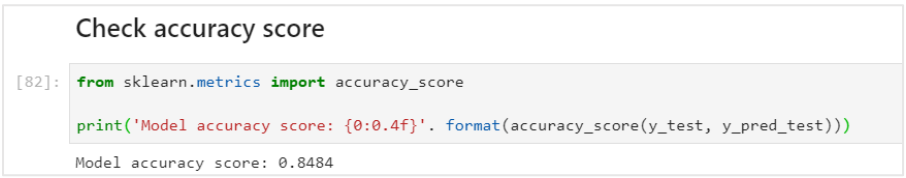
[78]:
```
# train a logistic regression model on the training set
from sklearn.linear_model import LogisticRegression


# instantiate the model
logreg = LogisticRegression(solver='liblinear', random_state=0)


# fit the model
logreg.fit(X_train, y_train)
```

[78]:
```
LogisticRegression(random_state=0, solver='liblinear')
```

รูป 39 แสดงหน้า code บางส่วนในขั้นตอน Model training

5. Evaluation

ทำการตรวจสอบค่า Accuracy และทำการตรวจสอบ Confusion Metrix หากผลที่ได้ยังไม่เป็นที่น่าพอใจ อาจมีการย้อนกลับไป fit model ใหม่อีกครั้ง



รูป 40 แสดงหน้า code ในการตรวจสอบค่า Accuracy

จากรูปที่ 40 เมื่อทำการหา Accuracy Score แล้ว มีค่าเท่ากับ 0.8484 ซึ่งแสดงให้เห็นว่าโมเดลนี้มีความแม่นยำสูง และเมื่อตรวจสอบ Confusion Metrix

รูป 41 แสดงหน้า code ในการหา Confusion Metrix

จากรูปที่ 41 จะได้ confusion metrix ดังรูป แต่เมื่อนำมาคิดค่า sensitivity โดยคำนวณได้จาก $\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$ จะมีค่าเท่ากับ 0.4931 หรือ 49.31% ซึ่งมีค่าที่น้อย และไม่ตรงกับเป้าหมายที่ต้องการให้มากกว่า 70% จึงได้ทำการ Add threshold level โดยใช้ค่า prob 0.1, 0.2, 0.3 และ 0.4 เพื่อใช้ในการตัดสินใจว่าควรจะเลือก threshold ไหนในการตัดสินใจตามเป้าหมายที่ตั้งไว้

```
[97]: from sklearn.preprocessing import binarize

for i in range(1,6):

    cm1=0

    y_pred1 = logreg.predict_proba(X_test)[:,1]

    y_pred1 = y_pred1.reshape(-1,1)

    y_pred2 = binarize(y_pred1, threshold=i/10)

    y_pred2 = np.where(y_pred2 == 1, 'Yes', 'No')

    cm1 = confusion_matrix(y_test, y_pred2)

    print ('With',i/10,'threshold the Confusion Matrix is ','\n\n',cm1,'\n\n',

            'with',cm1[0,0]+cm1[1,1],'correct predictions, ', '\n\n',

            cm1[0,1],'Type I errors( False Positives), ','\n\n',

            cm1[1,0],'Type II errors( False Negatives), ','\n\n',

           'Accuracy score: ', (accuracy_score(y_test, y_pred2)), '\n\n',

           'Sensitivity: ',cm1[1,1]/(float(cm1[1,1]+cm1[1,0])), '\n\n',

           'Specificity: ',cm1[0,0]/(float(cm1[0,0]+cm1[0,1])),'\n\n',

            '=====================================================', '\n\n')
```

รูป 42 แสดงหน้า code บางส่วนในการเพิ่ม threshold

```
With 0.1 threshold the Confusion Matrix is

 [[13291  9435]
 [  571  5795]]

 with 19086 correct predictions,

 9435 Type I errors( False Positives),

 571 Type II errors( False Negatives),

 Accuracy score:  0.6560566478757046

 Sensitivity:  0.9103047439522463

 Specificity:  0.5848367508580481

 =====================================================
```

```
With 0.2 threshold the Confusion Matrix is

 [[17742  4984]
 [ 1365  5001]]

 with 22743 correct predictions,

 4984 Type I errors( False Positives),

 1365 Type II errors( False Negatives),

 Accuracy score:  0.7817613089509143

 Sensitivity:  0.7855796418473139

 Specificity:  0.7806917187362492

 =====================================================
```

```
With 0.3 threshold the Confusion Matrix is

 [[19744  2982]
 [ 2043  4323]]

 with 24067 correct predictions,

 2982 Type I errors( False Positives),

 2043 Type II errors( False Negatives),

 Accuracy score:  0.8272721022961639

 Sensitivity:  0.679076343072573

 Specificity:  0.8687846519405087

 =====================================================
```

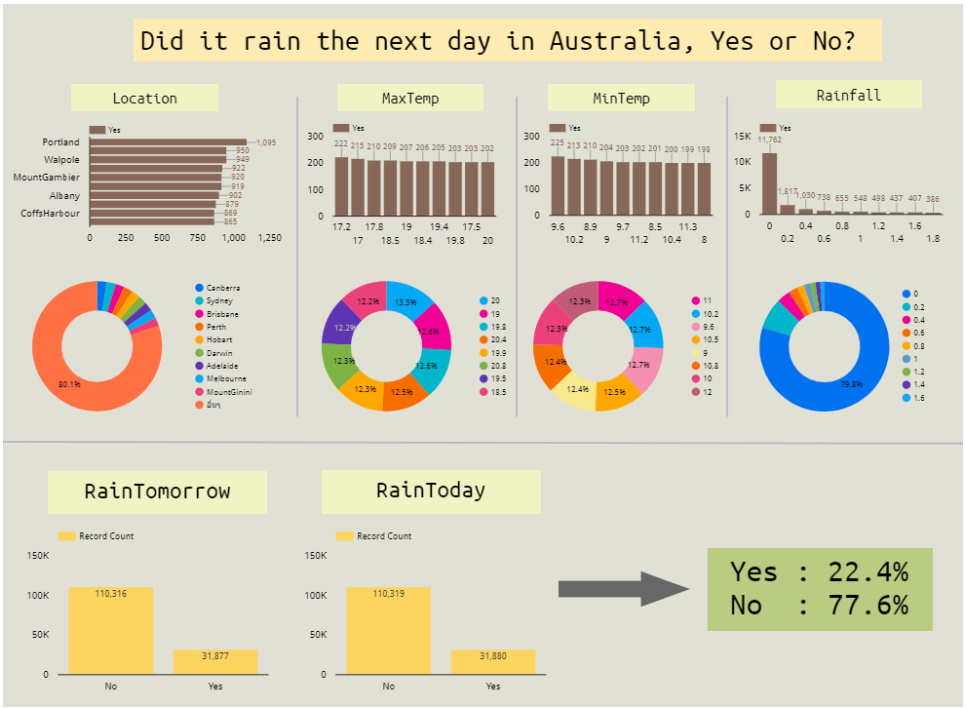
```
With 0.4 threshold the Confusion Matrix is

 [[20840  1886]
 [ 2645  3721]]

 with 24561 correct predictions,

 1886 Type I errors( False Positives),

 2645 Type II errors( False Negatives),

 Accuracy score:  0.8442527155231678

 Sensitivity:  0.5845114671693371

 Specificity:  0.9170113526357476

 =====================================================
```

รูป 43 แสดงผลลัพธ์จากการเพิ่ม threshold

จากรูปที่ 43 จะเห็นว่าโมเดลที่ใช้ค่าความน่าจะเป็นที่ 0.1 เป็นเกณฑ์มีค่า Sensitivity เท่ากับ 0.9103 โมเดลที่ใช้ค่าความน่าจะเป็นที่ 0.2 เป็นเกณฑ์มีค่า Sensitivity เท่ากับ 0.7856 โมเดลที่ใช้ค่าความน่าจะเป็นที่ 0.3 เป็นเกณฑ์มีค่า Sensitivity เท่ากับ 0.6791 และโมเดลที่ใช้ค่าความน่าจะเป็นที่ 0.4 เป็นเกณฑ์มีค่า Sensitivity เท่ากับ 0.5845 โดยจะทำการเลือกโมเดลที่ใช้ค่าความน่าจะเป็นที่ 0.2 เป็นเกณฑ์เนื่องจากมีค่า sensitivity เท่ากับ 0.7856 ซึ่งอยู่ในเกณฑ์ที่ได้ตั้งเป้าหมายไว้ นั่คือมีโอกาสที่ฝนจะตกมากกว่า 70% ดังนั้นจึงทำการเลือกโมเดลนี้เป็นโมเดลที่เหมาะสม โดยเราสามารถใช้โมเดลนี้เป็นเกณฑ์ในการตัดสินใจได้ว่าสรุปแล้วเราควรเตรียมตัวรับมือกับฝนหรือไม่

6. Deployment

อาจนำผลลัพธ์ที่ได้จากการทำโปรเจคนี้ไปเผยแพร่ให้ users ได้ลองใช้งานจริง หรืออาจนำไปเผยแพร่ใน Kaggle เพื่อที่จะได้รับคำแนะนำ หรือเห็นข้อผิดพลาดในโมเดลเรามากขึ้น เพื่อเป็นแนวทางในการแก้ไขและพัฒนาโมเดลของเราต่อไป ในที่นี้ได้ลองเผยแพร่ใน Google Data Studio เพื่อแสดงให้ users เห็น Dashboard ที่น่าสนใจ



รูป 44 แสดง Dashboard ที่น่าสนใจ

https://datastudio.google.com/s/pMh9bok5g0Y

ผลลัพธ์ที่ได้ และสรุปผล

โมเดลที่ได้นั้นเป็น classification model ทำนายการตกของฝนในวันต่อไปในประเทศออสเตรเลีย โดยโมเดลที่ได้เป็นโมเดลที่ใช้เกณฑ์ความน่าจะเป็น 0.5 ซึ่งมีค่า Accuracy score เท่ากับ 0.8484 หลังจากนั้นทำการแบ่ง threshold อีก 4 threshold ดังนี้

ตาราง 3 ตารางแสดงรายละเอียดของโมเดลแต่ละ threshold

Threshold	Confusion Metrix	Accuracy	Sensitivity
0.1	[[13291 9435] [571 5795]]	0.6561	0.9103
0.2	[[17742 4984] [1365 5001]]	0.7818	0.7856
0.3	[[19744 2982] [2043 4323]]	0.8273	0.6791
0.4	[[20840 1886] [2645 3721]]	0.8443	0.5845
0.5	[[21543 1183] [3227 3139]]	0.8484	0.4931

จากการแบ่งเกณฑ์นี้เพื่อเลือกโมเดลที่เหมาะสมที่สุดในการตัดสินใจ Goal โดยมีเกณฑ์การตัดสินใจคือ เราจะทำการเตรียมตัวรับมือกับฝนตก ถ้าหากโอกาสที่ฝนจะตกอยู่ที่ประมาณ 70% ซึ่งสามารถสรุปได้ว่าเราจะเลือกโมเดลที่มี threshold ที่ใช้ค่าความน่าจะเป็นที่ 0.2 เป็นเกณฑ์ มีค่า Accuracy score เท่ากับ 0.7818 ซึ่งมีค่าน้อยกว่า Accuracy score ของโมเดลที่ใช้ค่าความน่าจะเป็นที่ 0.5 แต่เนื่องจากมีค่า sensitivity เท่ากับ 0.7856 ซึ่งอยู่ในเกณฑ์ที่ได้ตั้งเป้าหมายไว้ ดังนั้นจึงทำการเลือกโมเดลนี้เป็นโมเดลที่เหมาะสม โดยเราสามารถใช้โมเดลนี้เป็นเกณฑ์ในการตัดสินใจได้ว่าสรุปแล้ว เราควรเตรียมตัวรับมือกับฝนหรือไม่

อภิปรายสิ่งที่ได้เรียนรู้และแนวทางในการพัฒนาต่อยอด

จากการทำโปรเจคครั้งนี้ ผู้จัดทำได้เรียนรู้การใช้ Google Cloud Platform ในการจัดการกับข้อมูล เก็บข้อมูล และนำข้อมูลที่มีไปใช้ต่อยอดได้ และได้ฝึกฝน พัฒนาการเขียนภาษา Python ได้เรียนรู้ในลักษณะของ CRISP-DM ตั้งแต่การทำความเข้าใจธุรกิจ การทำความเข้าใจข้อมูล การจัดการข้อมูล clean ข้อมูลด้วยวิธีที่เหมาะสม การสร้างโมเดล เพื่อช่วยในการตัดสินใจ Goal ที่เราต้องการ รวมไปถึงการ Evaluation เพื่อวิเคราะห์และพัฒนาโมเดลที่เราได้มา โดยในที่นี้ได้นำข้อมูลเกี่ยวกับการตกของฝนในออสเตรเลียมาเป็นตัวหลักในการช่วยศึกษา ซึ่งจะเป็นการทำ Classification Model เพื่อทำนายผลลัพธ์ว่าฝนจะตกหรือไม่ในวันพรุ่งนี้ จากการนำตัวแปรที่มีผลต่อการตกของฝนมาพิจารณาร่วมด้วย และมีการใช้ Platform ต่างๆ ใน GCP เช่น Google Cloud Storage ที่ใช้ในการเก็บข้อมูล หรือ BigQuery ที่ใช้ในการคิวรี่ข้อมูล สร้าง table เพื่อนำไปสร้าง Dashboard ที่น่าสนใจใน Google Data Studio เป็นต้น โดยโปรเจคนี้สามารถนำมาต่อยอดได้ในประเทศไทยได้ โดยการประยุกต์ข้อมูล ประยุกต์วิธีการจัดการกับข้อมูลต่างๆ เนื่องจากอาจมีวิธีที่เหมาะสม หรือโมเดลที่เหมาะสมกับข้อมูลของประเทศไทยมากกว่านี้ เพื่อทำนายการตกของฝนในประเทศไทย หรืออาจจะมีการใช้ Platform ใน Google Cloud Platform ให้มากกว่านี้ เช่น Google Cloud Dataprep อาจช่วยให้การจัดเตรียมข้อมูลก่อนนำเข้าโมเดลง่ายและได้ประสิทธิภาพที่ดีกว่า ซึ่งโมเดลในการทำนายการตกของฝนในประเทศไทยนี้ หากนำมาต่อยอดจากสิ่งที่ทำมาจะมีประโยชน์ในด้านของการทำให้ผู้คนสามารถใช้ชีวิตได้ดีและสะดวกยิ่งขึ้น ยกตัวอย่างเช่น ชาวประมงจะสามารถตัดสินใจได้ง่ายขึ้น ว่าควรนำเรือออกไปทำการประมงหรือไม่ เนื่องจากฝนก็เป็นอีกหนึ่งปัจจัยที่มีผลต่อการทำประมง หรือจะเป็นเรื่องที่ใกล้ตัวมากกว่านั้น อย่างเช่น ถ้าหากเรารู้ว่าวันนี้ฝนจะตกและจำเป็นต้องเดินทางโดยการขับขี่ยานพาหนะ จะทำให้เราสามารถระมัดระวังได้มากขึ้น ไม่ประมาทกับการขับขี่ ส่งผลให้อุบัติเหตุอาจลดน้อยลงได้นั่นเอง