



Rain prediction using ANN

Ahmed Alelg



Problem statement

To predicate whether it will rain or not in Australia.



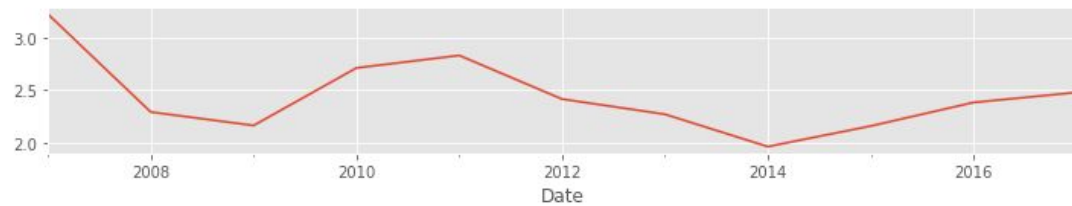
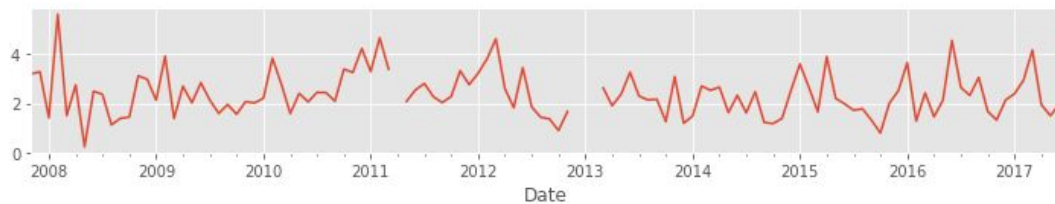
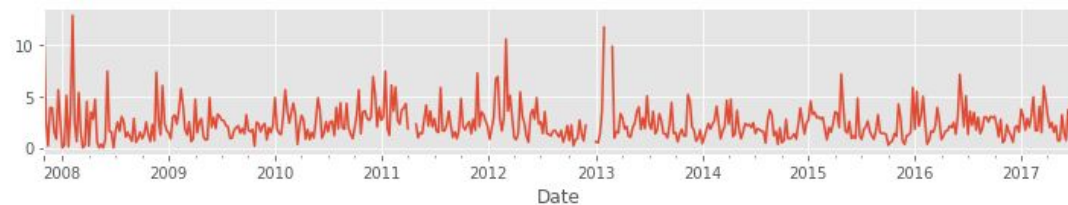
Dataset

- Outsourced from Kaggle.
- 23 Columns
 - 1 Datetime
 - 15 Numerical
 - 7 Categorical
- 145,460 observations

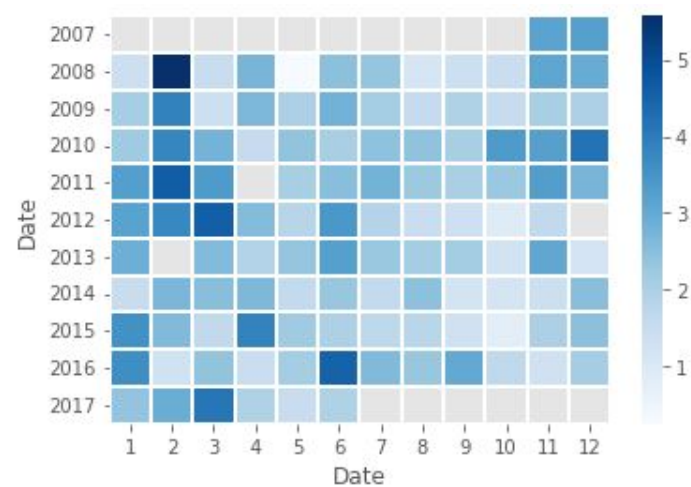
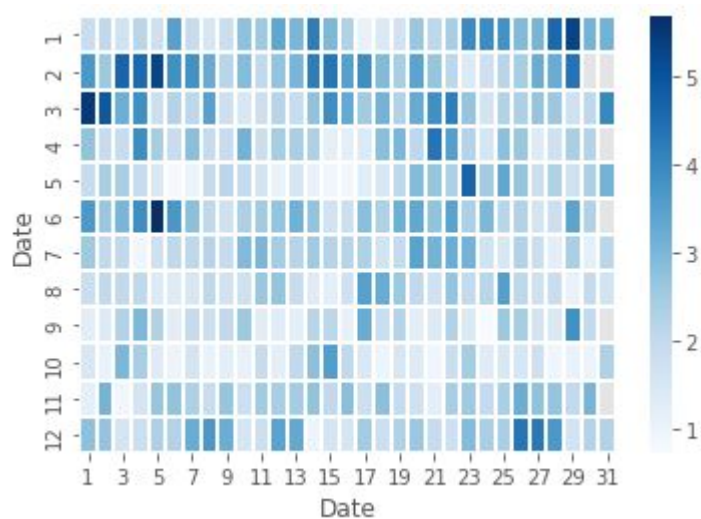
#	Column	Non-Null Count	Dtype
0	Date	145460 non-null	datetime64[ns]
1	Location	145460 non-null	object
2	MinTemp	143975 non-null	float64
3	MaxTemp	144199 non-null	float64
4	Rainfall	142199 non-null	float64
5	Evaporation	82670 non-null	float64
6	Sunshine	75625 non-null	float64
7	WindGustDir	135134 non-null	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: datetime64[ns](1), float64(16), object(6)
memory usage: 25.5+ MB

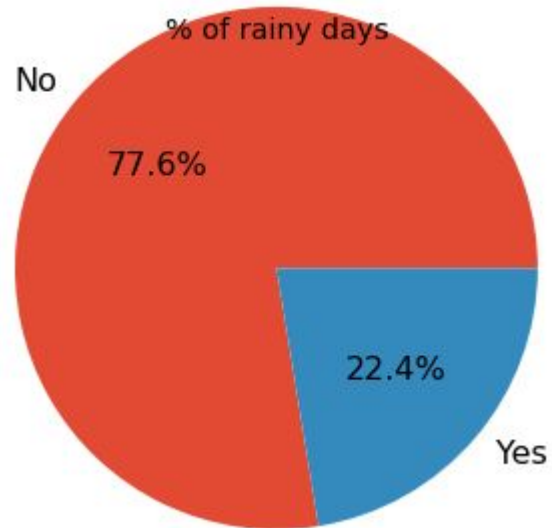
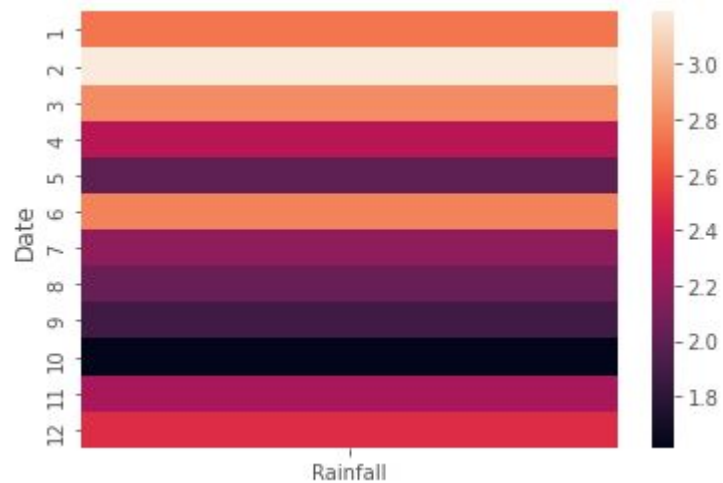
EDA



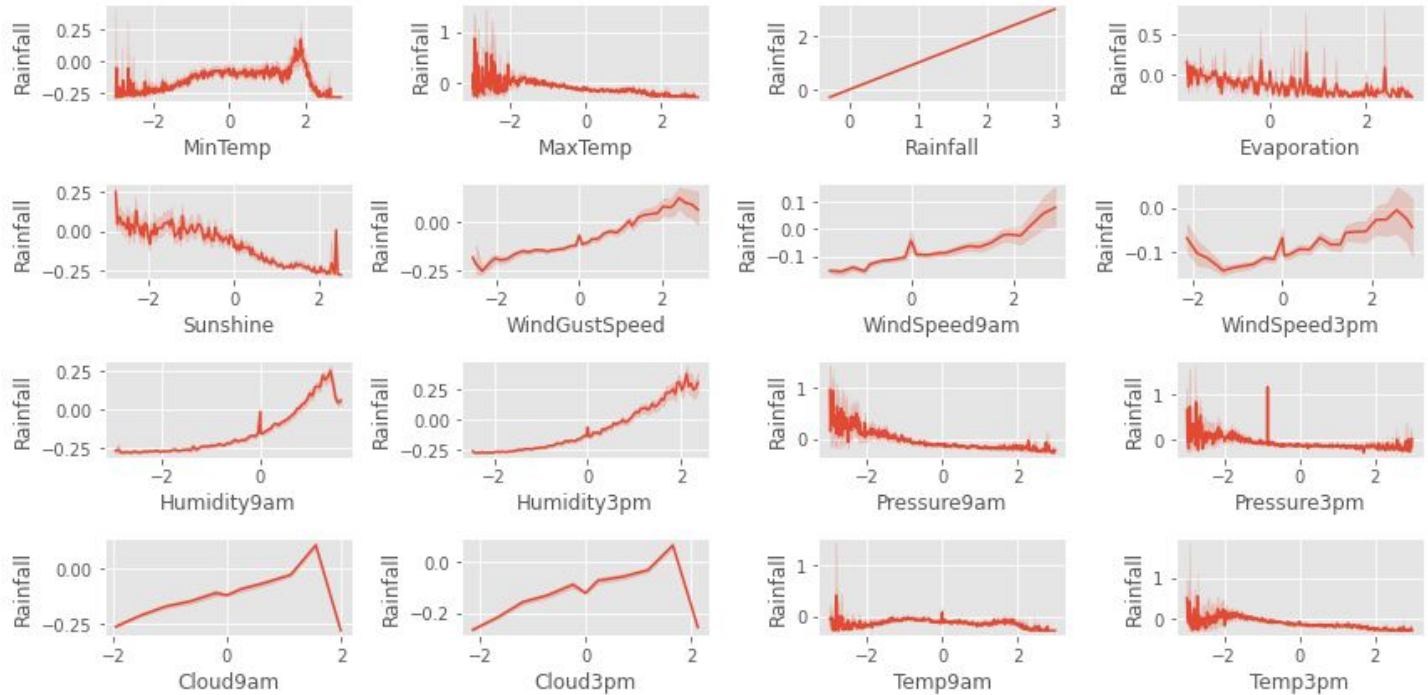
EDA



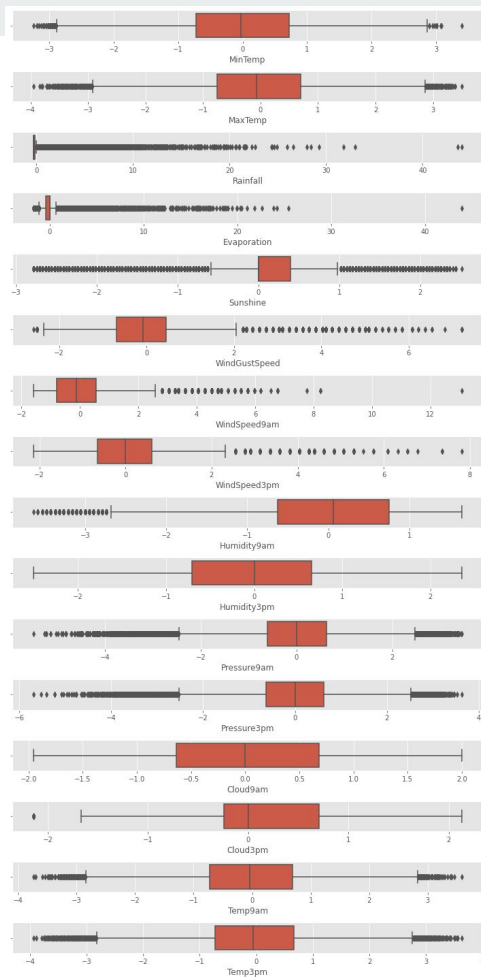
EDA



EDA

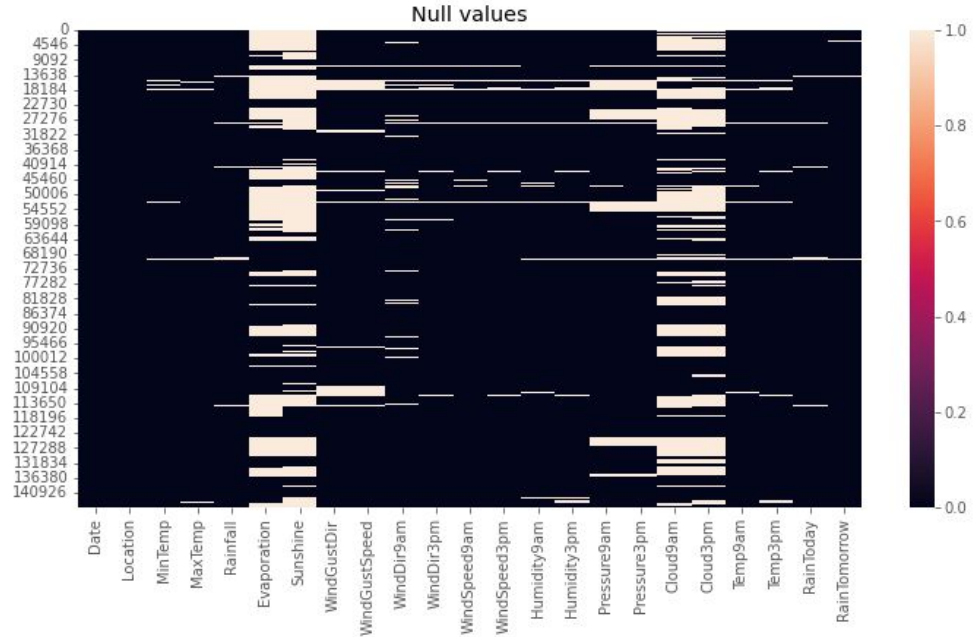


EDA



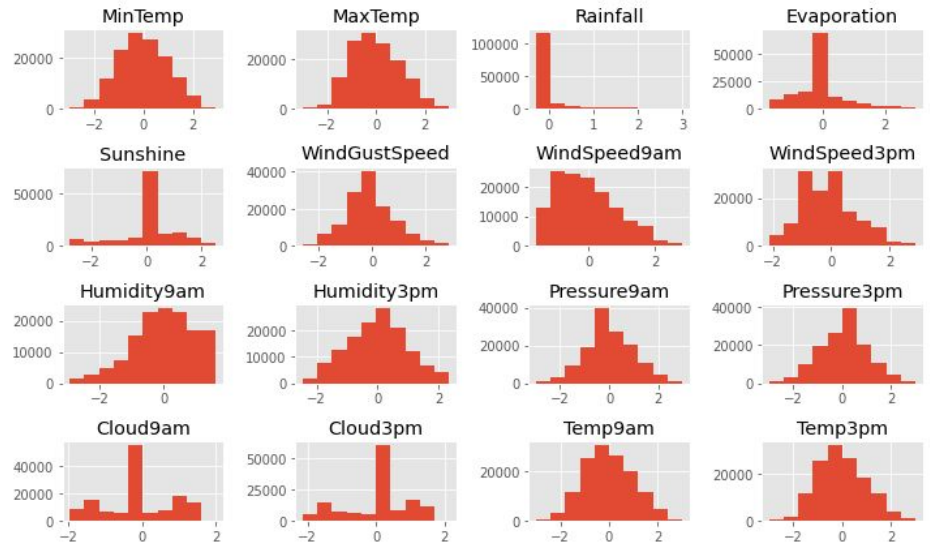
Data engineering

- Many columns with missing values!
 - Numerical values filled with the mean
 - Categorical values filled with the its neighbor.
 - Columns having >50% missing values were dropped.



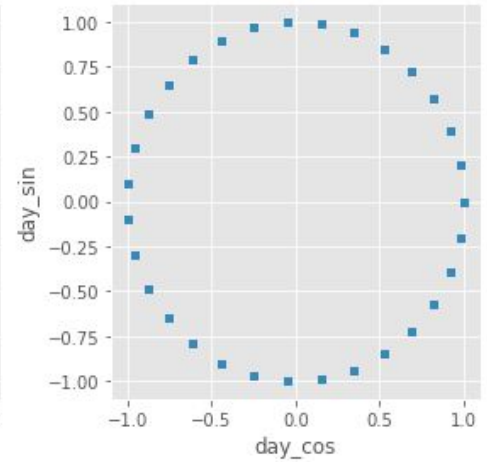
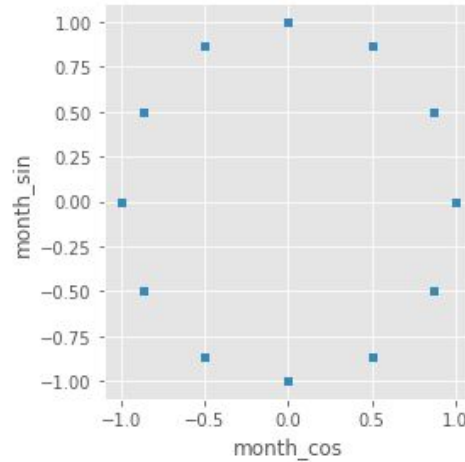
Data engineering

- Standardized the numerical values to have range within $-1, 1$.
- Removed outliers whose values exceeded $[-3, 3]$
 - $n = 8,674$
- Removed columns with high collinearity $VIF > 5$.
 - $n = 4$



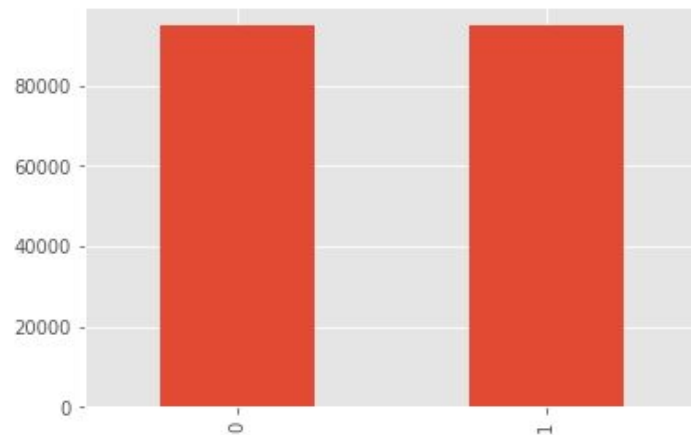
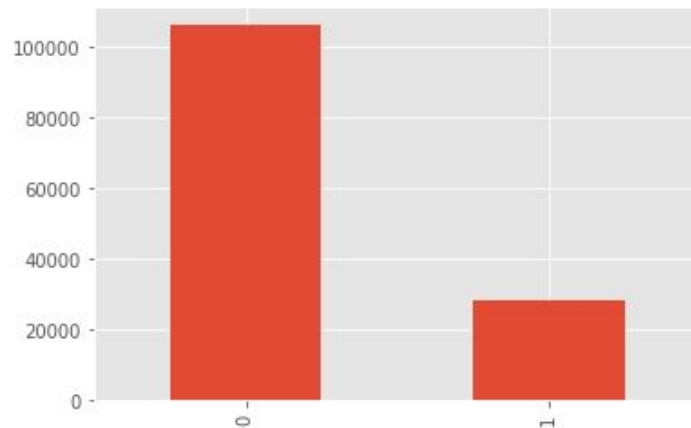
Data engineering

- Encoded categorical values to integers.
- Created vector embeddings for categorical values.
- Encoded Month/Day values into cyclical features.



Data split & balance

- 90% Training data
 - Balanced classes using SMOTE
 - (n = 189,680)
- 5% Validation
- 5% Testing





Model architecture

- ANN with 6 layers
- Used embeddings to encode categorical data.
- ReLU activation functions.
- Included dropout and Batch normalization.
- Adam.
- Binary cross entropy loss.

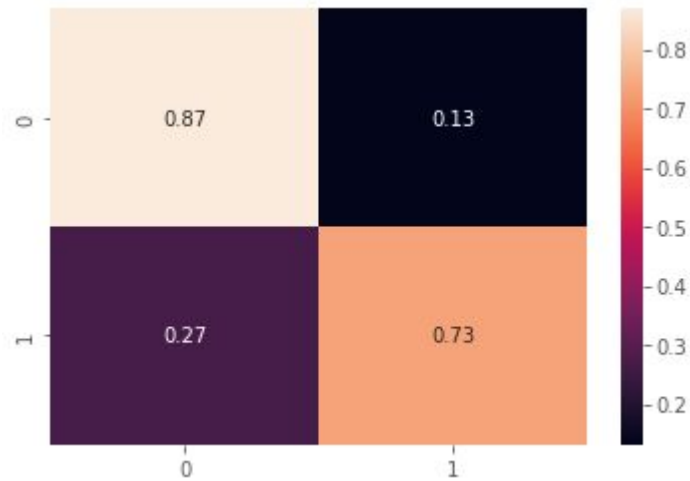
```
NNet(  
  (embs): ModuleList(  
    (0): Embedding(49, 25)  
    (1): Embedding(16, 8)  
    (2): Embedding(16, 8)  
    (3): Embedding(16, 8)  
  )  
  (emb_drop): Dropout(p=0.05, inplace=False)  
  (act): ReLU()  
  (init): Sequential(  
    (0): Linear(in_features=69, out_features=138, bia  
    (1): BatchNorm1d(138, eps=1e-05, momentum=0.1, af  
    (2): ReLU()  
    (3): Dropout(p=0.3, inplace=False)  
    (4): Linear(in_features=138, out_features=256, bi  
    (5): BatchNorm1d(256, eps=1e-05, momentum=0.1, af  
    (6): ReLU()  
    (7): Dropout(p=0.3, inplace=False)  
    (8): Linear(in_features=256, out_features=128, bi  
    (9): BatchNorm1d(128, eps=1e-05, momentum=0.1, af  
    (10): ReLU()  
    (11): Dropout(p=0.3, inplace=False)  
    (12): Linear(in_features=128, out_features=64, bi  
    (13): BatchNorm1d(64, eps=1e-05, momentum=0.1, af  
    (14): ReLU()  
    (15): Dropout(p=0.3, inplace=False)  
    (16): Linear(in_features=64, out_features=16, bia  
    (17): BatchNorm1d(16, eps=1e-05, momentum=0.1, af  
    (18): ReLU()  
    (19): Dropout(p=0.1, inplace=False)
```



Results

	Training	Validation	Testing
Loss	0.339279	0.365065	0.362375
Accuracy	84.71%	84.23%	84.02%
F1	0.8428	0.6689	0.6517

Results: Test data





Libraries used

- Numpy
- Pandas
- Sklearn
- PyTorch
- imbalanced-learn
- seaborn & matplotlib