Rain prediction model using ANN

Dataset:

-Link in Kaggle: https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package

- This dataset contains daily weather observations from 2008 to 2017 for the most of Australia`s states

- Our targe here is predicting if it would rain the next day or not

- The data original shape is(145461,23)

Data Wrangling & Preprocessing:

First, we have a time series at Date column –splitting it into year,month,day at month and day we convert them to cyclic continuous feature and encoding them : without it the model will train the day 1 in specific month and day 2 arent near also with months it doesn't consider that month 12 not near to month 1
and this increases the model`s accuracy

References:

deep learning - Encoding Date/Time (cyclic data) for Neural Networks - Cross Validated (stackexchange.com)

Encoding Cyclical Features for Deep Learning (kaggle.com)

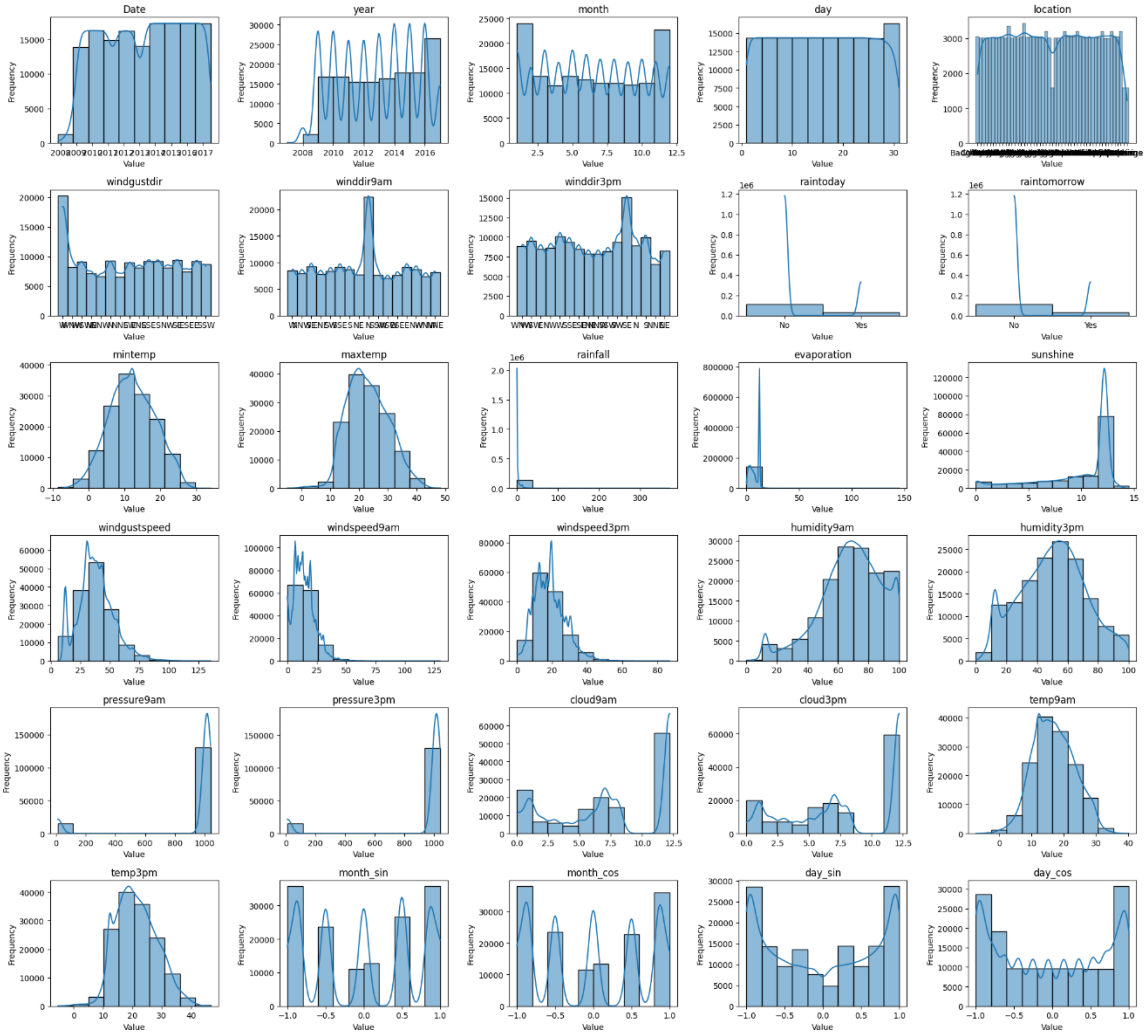
-Dealing with null values

```
...   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
      year                0
      month               0
      month_sin           0
      ...
      month_cos           0
      day                 0
      day_sin             0
      day_cos             0
      dtype: int64
```

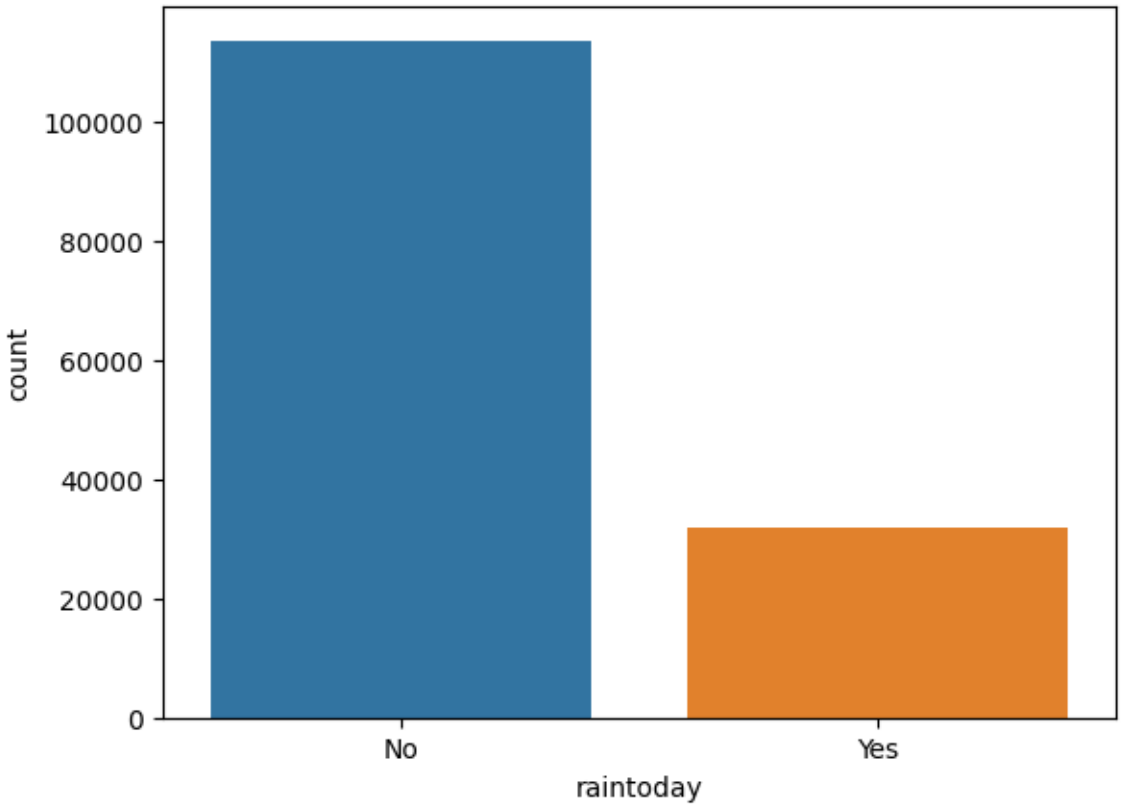
We got that percentage of null values is not large with respect to all data rows so I`ve filled categorical features with mode

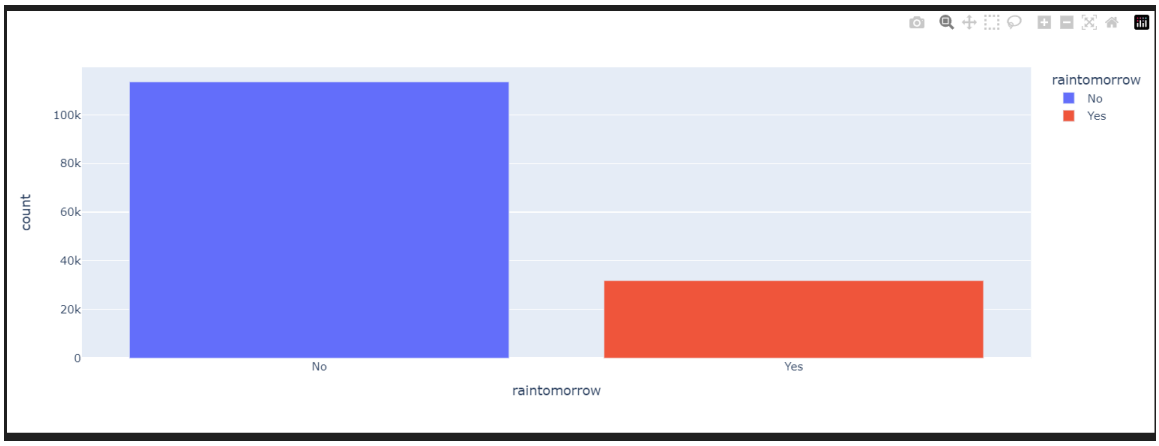
And numerical columns with median because most of columns aren’t normal distribution and there are many outliers will affect on the values using mean

This a visualization of distribution after filling null values



From problems of this data that there are columns are imbalanced like these:





There is high difference between raining or not this causes data leckage during training the model

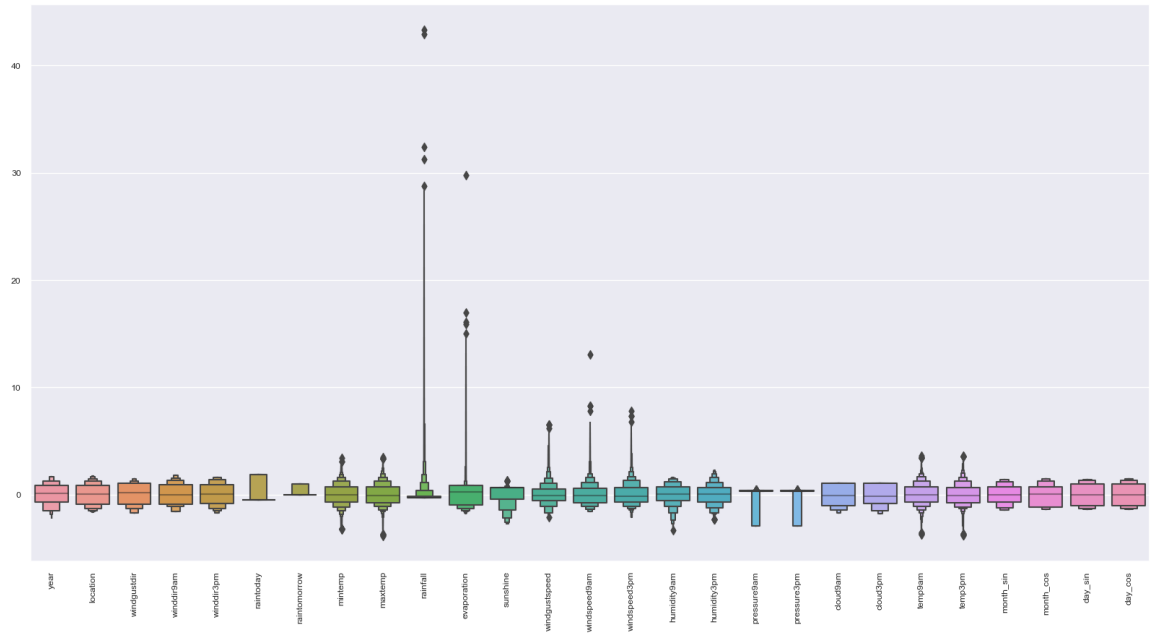
We`ve handled it using oversampling Smote algorithm to make a balance between columns

---

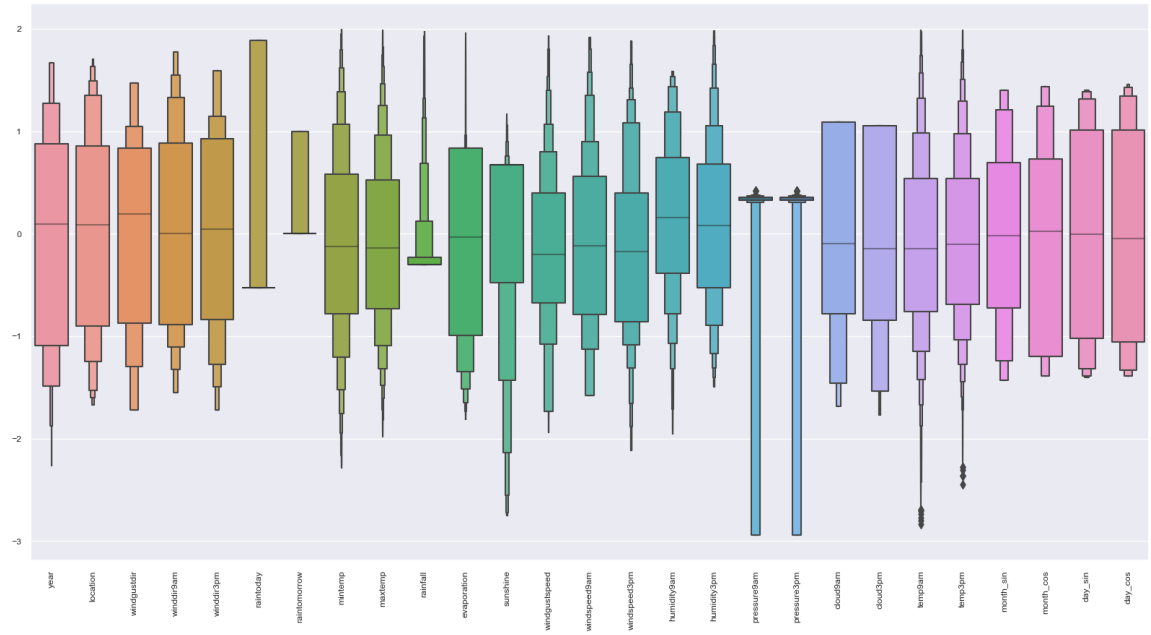
Deleting outliers:

Using z-score and quantile range

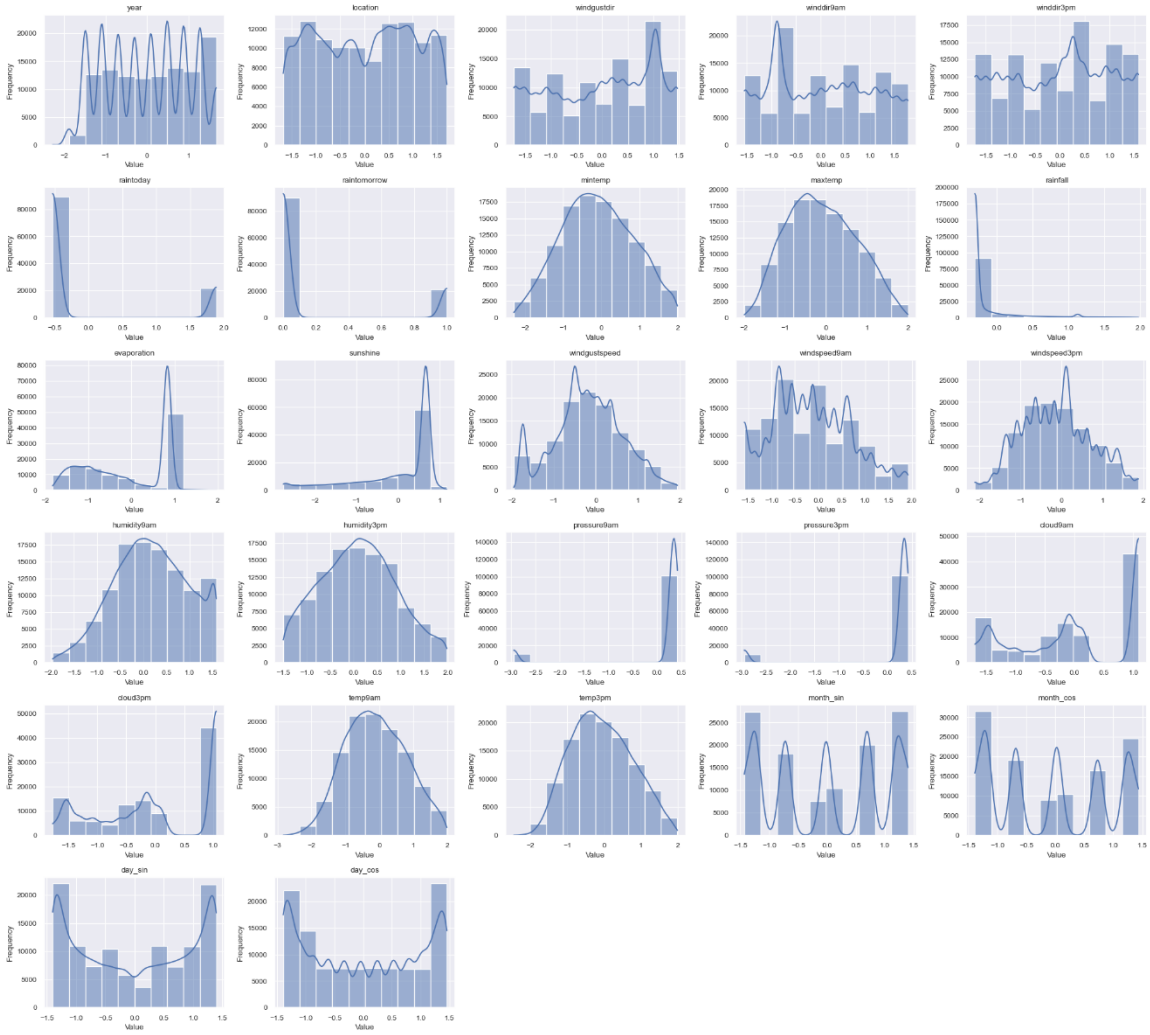
Visualization before deleting outliers



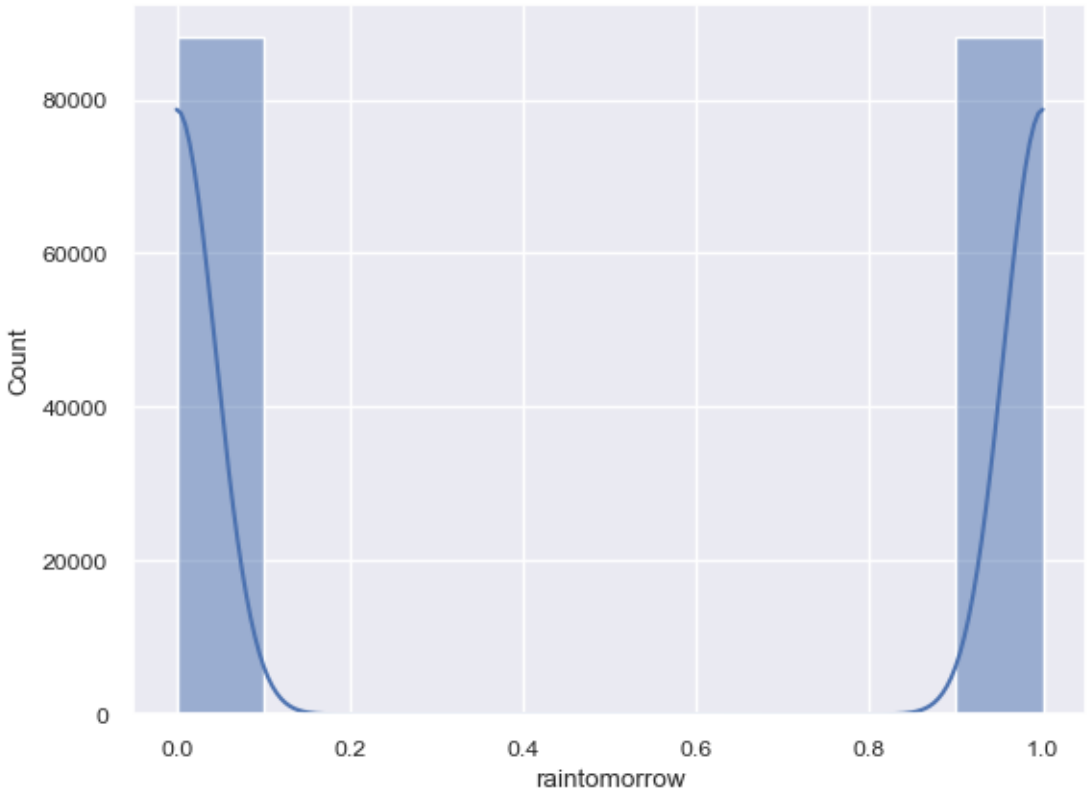
And after deleting:



Distribution after deleting outliers:



After using oversampling



Encoding categorical columns

-using LableEncoder

Before encoding:

	Date	year	month	day	location	windgustdir	winddir9am	winddir3pm	raintoday	raintomorrow	...	pressure9am	pressure3pm	cloud9am	cloud3pm	temp
20604	2016-01-28	2016	1	28	NorahHead	NE	NNW	NE	Yes	Yes	...	1009.000000	1004.700000	12.194034	12.194034	
15749	2010-10-05	2010	10	5	Newcastle	W	N	SE	Yes	Yes	...	12.194034	12.194034	7.000000	4.000000	
90501	2009-08-19	2009	8	19	GoldCoast	S	SSE	SE	No	No	...	1027.200000	1023.300000	12.194034	12.194034	
128336	2013-05-16	2013	5	16	Walpole	W	N	SE	No	Yes	...	1013.000000	1009.500000	12.194034	12.194034	
20025	2014-06-28	2014	6	28	NorahHead	NNW	N	N	No	No	...	1008.500000	1000.800000	12.194034	12.194034	

5 rows × 30 columns

After encoding:

```
df1.head()
```

Open 'df1' in Data Wrangler — Python

	Date	year	month	day	location	windgustdir	winddir9am	winddir3pm	raintoday	raintomorrow	...	pressure9am	pressure3pm	cloud9am	cloud3pm	temp9am	t
0	2008-12-01	2008	12	1	2	13	13	14	0	0	...	1007.7	1007.1	8.000000	12.194034	16.9	
1	2008-12-02	2008	12	2	2	14	6	15	0	0	...	1010.6	1007.8	12.194034	12.194034	17.2	
2	2008-12-03	2008	12	3	2	15	13	15	0	0	...	1007.6	1008.7	12.194034	2.000000	21.0	
3	2008-12-04	2008	12	4	2	4	9	0	0	0	...	1017.6	1012.8	12.194034	12.194034	18.1	
4	2008-12-05	2008	12	5	2	13	1	7	0	0	...	1010.8	1006.0	7.000000	8.000000	17.8	

5 rows × 30 columns

Normalization

-using StandardScaler

Splitting Data:

Training------- 99.98%

Test---------- 0.02

X_train.shape(176155,26)

Y_train.shape(176155,)

X_train.shape(3595,26)

X_test.shape(3595,)

Modeling

-Using keras sequence model with:

6layers:

Input layer with

3 hidden layers with output neuons respectively (32,32,16,8)

Activation function is used in hidden layers is : Relu function

Optimizer: ADAM(Adaptive Moment Estimation)

Loss function: Binary Crossentropy

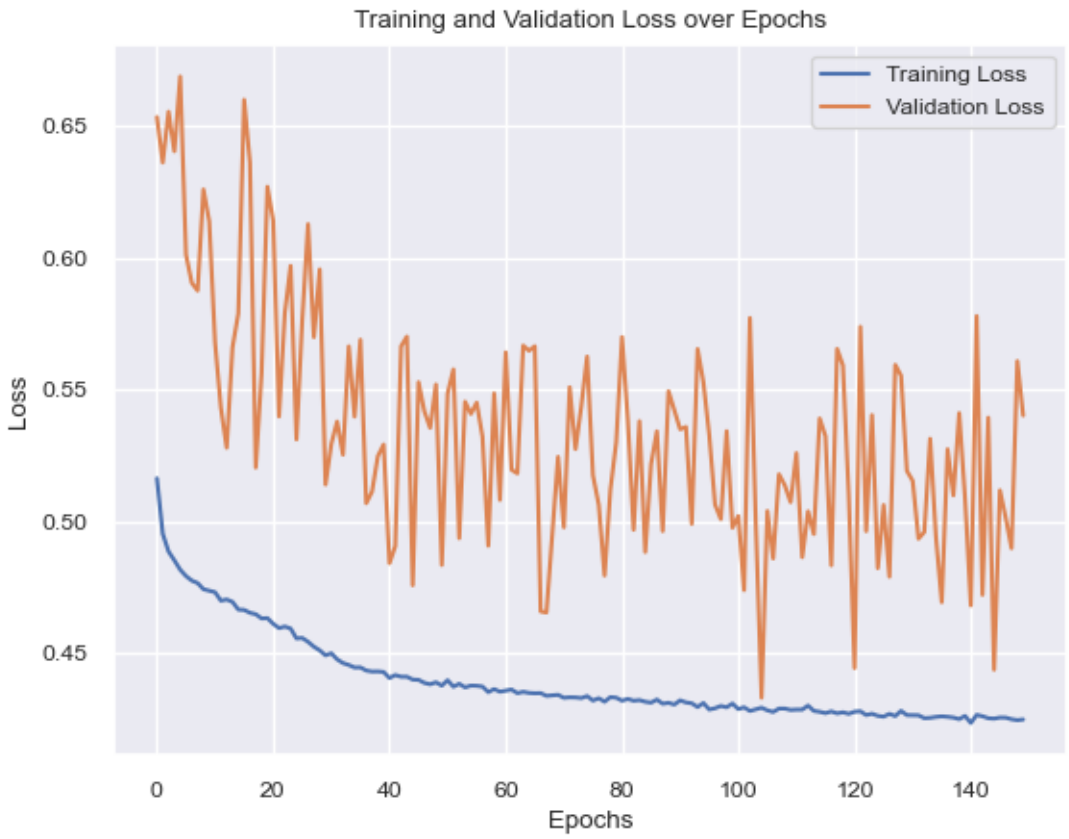
Metrics: Accuracy

Batch Size = 32

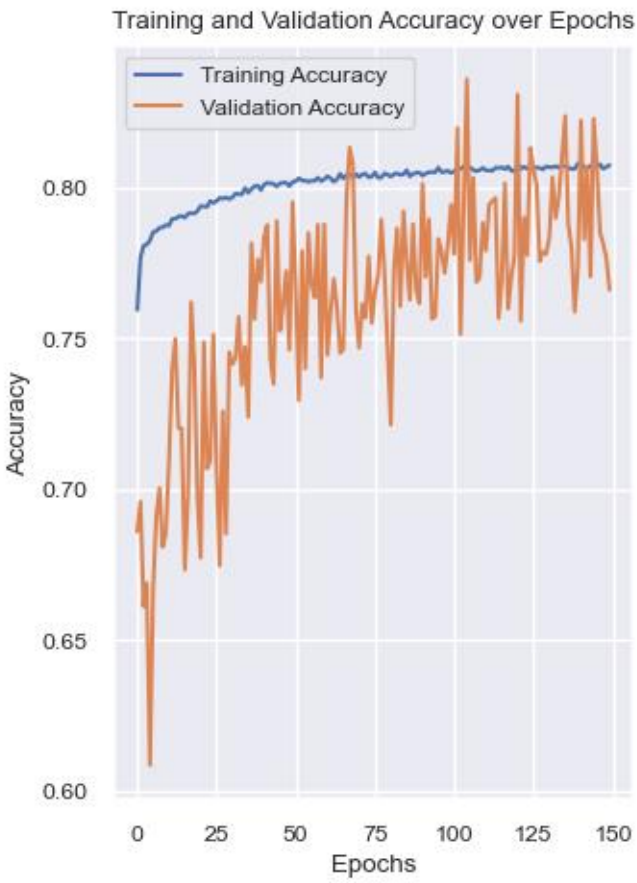
Epochs = 150

Validation Split = 0.2

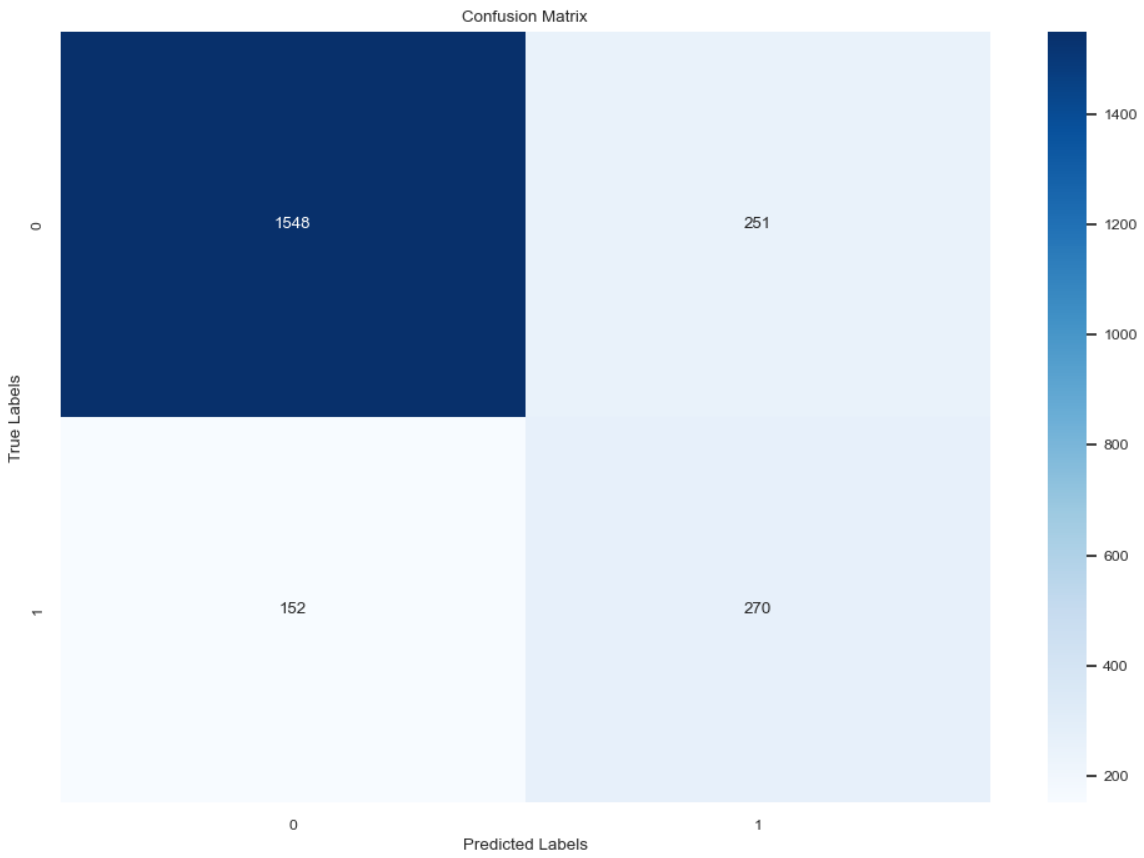
Loss Curve over epochs:



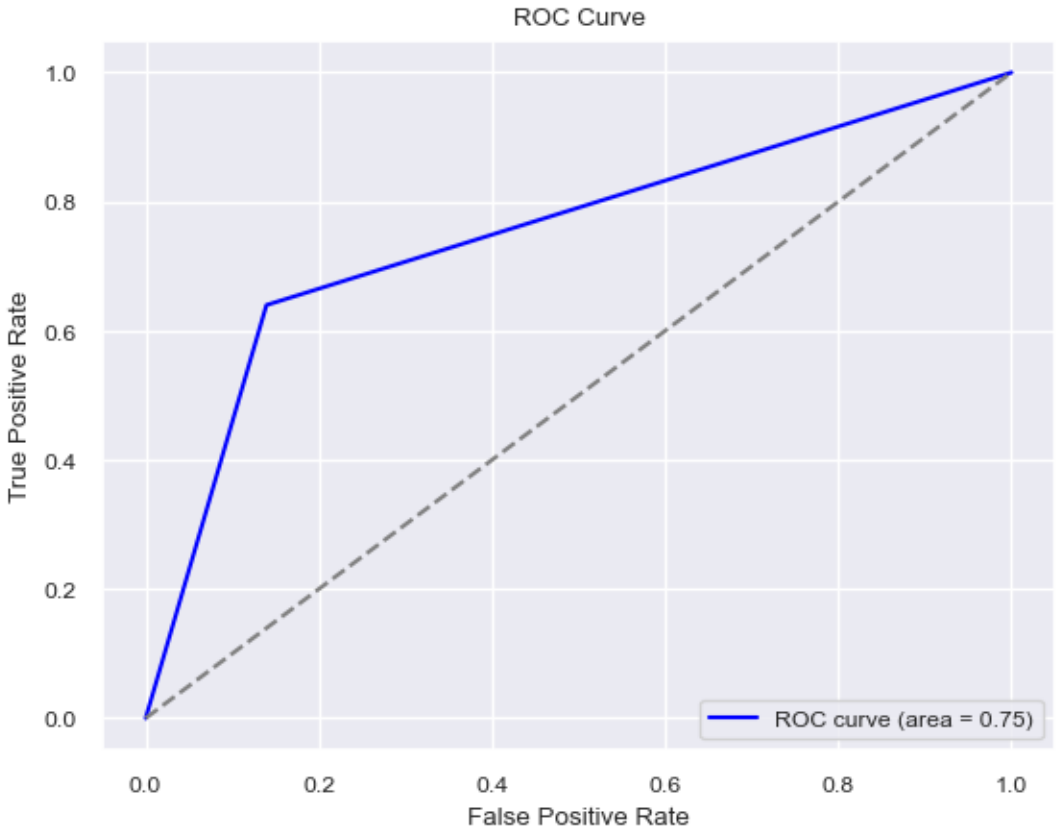
Accuracy over epochs:


Training and Validation Accuracy over Epochs
Training Accuracy
Validation Accuracy
Accuracy
0.80
0.75
0.70
0.65
0.60
0
25
50
75
100
125
150
Epochs

Confusion matrix:



Roc curve:


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.75)

```
..                precision    recall  f1-score   support

             0       0.91      0.86      0.88      1799
             1       0.52      0.64      0.57       422
    ...
      accuracy                           0.82      2221
     macro avg       0.71      0.75      0.73      2221
  weighted avg       0.84      0.82      0.83      2221
```