BrainDead2k23 (Revelation23)

GitHub Link: [BrainDead2k23-Duo-Lipa]

Team Members: Jeevesh Mahajan (IIT Kharagpur)
Aritra Sinha (IIT Kharagpur)

Problem Statement 1: Analyze Placement Data: Code Link: Problem 1

Challenge Description:

In this challenge, you are supposed to analyze the placement records of the students of an MBA college.

The dataset includes secondary and higher secondary school percentages and specializations. It also contains degree specialization, work experience, and the salary offered to the students. Your main task is to analyze the factors that affect the placement and salary of students.

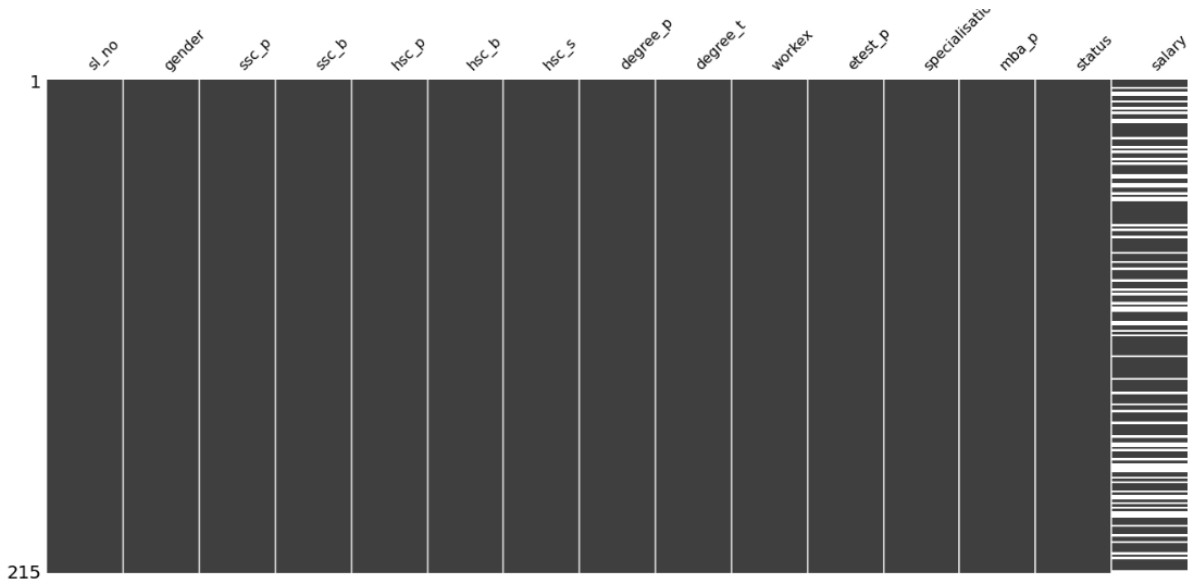
Analyze the dataset and derive meaningful insights from the data.

1. Data Description:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0

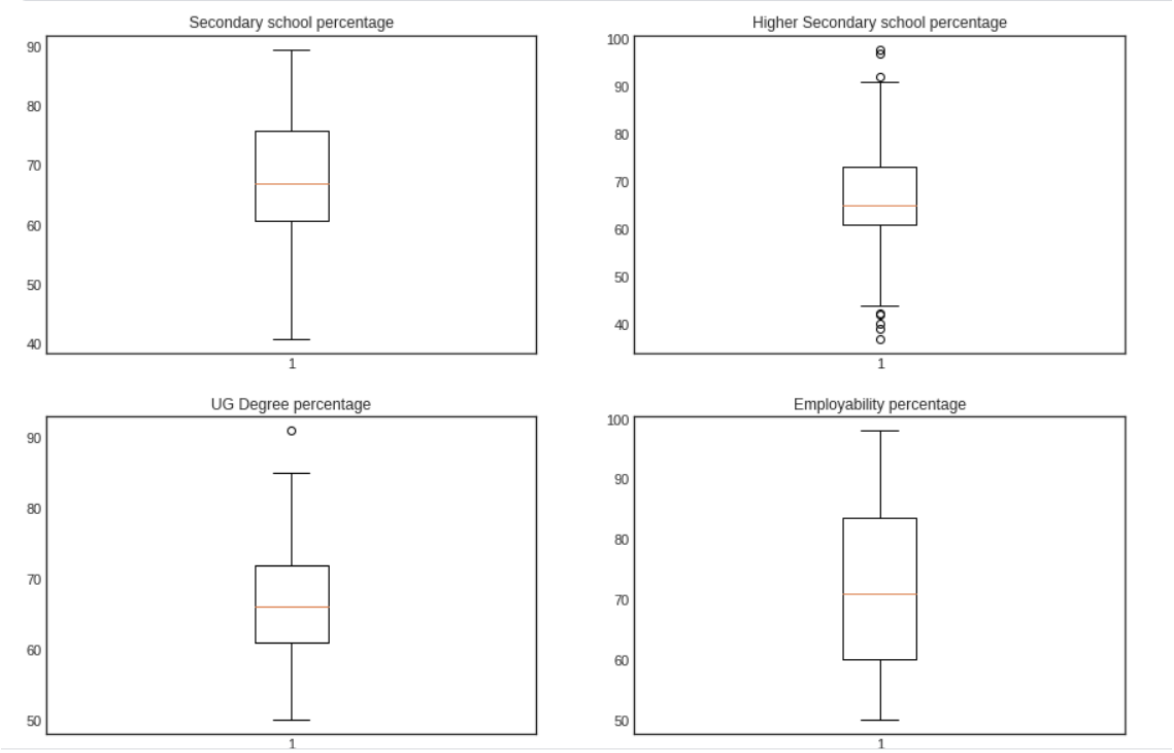
- We have **Gender and Educational qualification** data
- We have all the **educational performance (score)** data
- We have the **status** of placement and salary details
- We can expect **null values in salary** as candidates who weren't placed would have no salary
- **Status** of placement is the target variable rest of them are independent variables except salary

2. Missing Data:

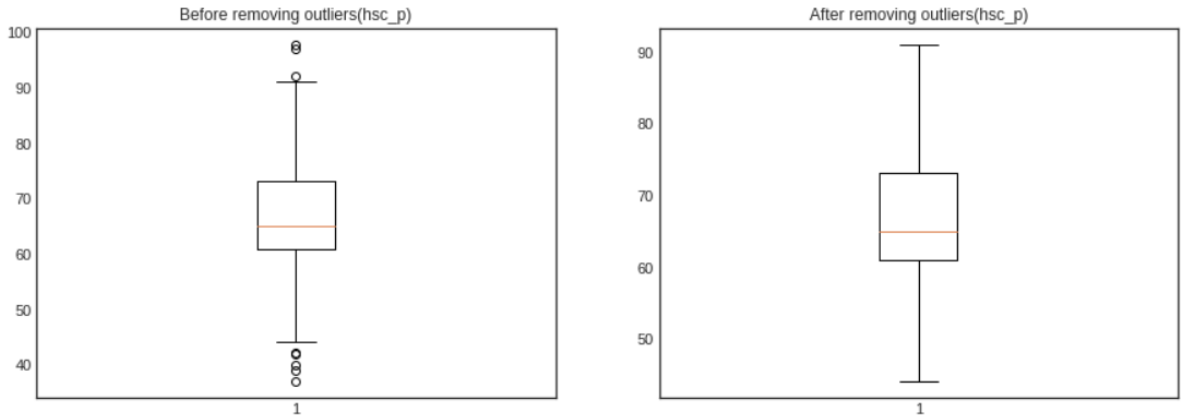


- The only null values are present in **salary** as expected with the count equal to **67 which** means 67 **unhired candidates.**
- We can't drop these values as this will provide valuable information on why candidates failed to get hired.
- We can't impute it with mean/median values and it will go against the context of this dataset and it will show unhired candidates got a salary.
- Our best way to deal with these null values is to **impute it with '0'** which shows they don't have any income

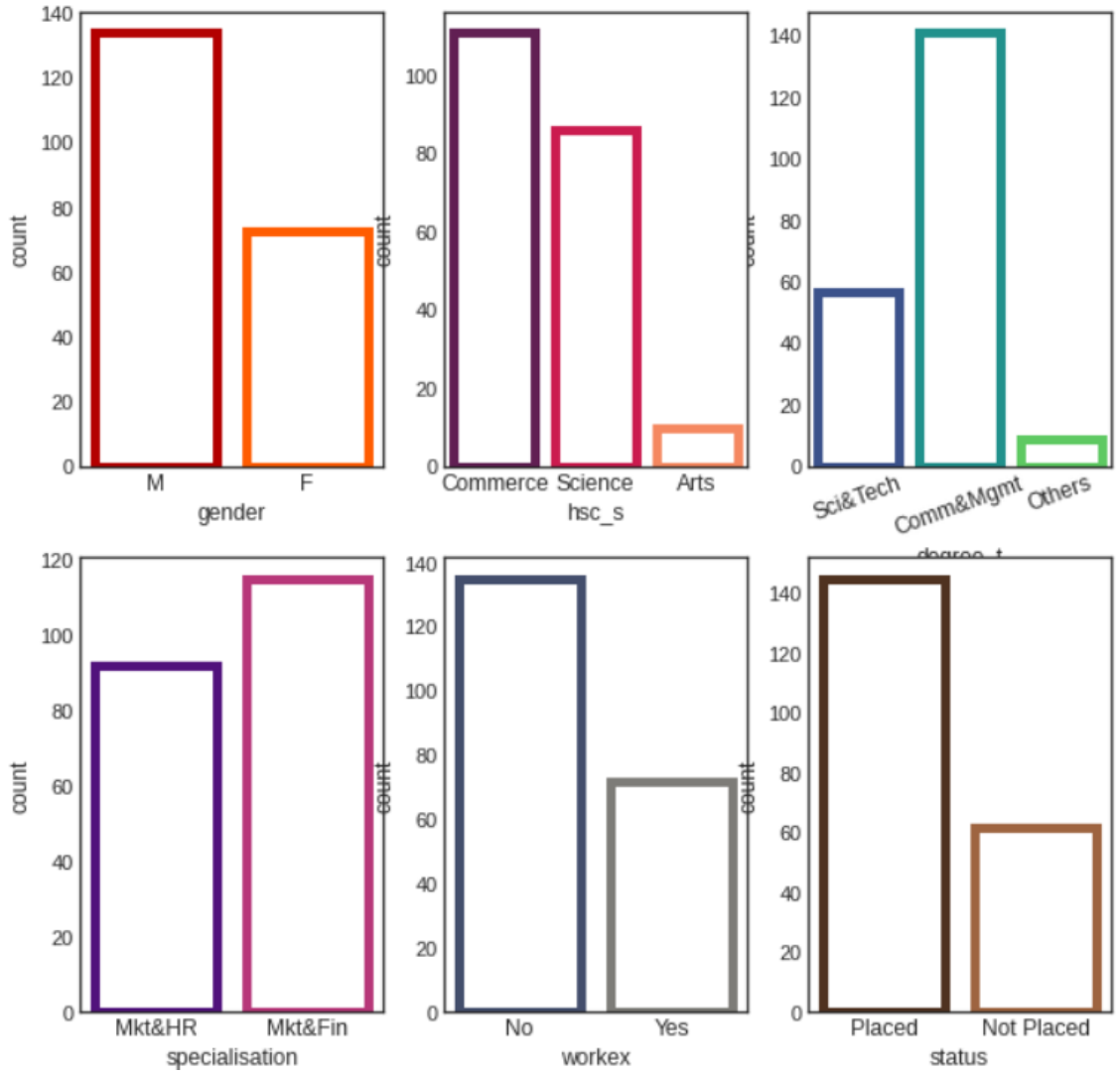
3. Outliers:



- We have a very less number of outliers in our features. Especially we have the majority of the outliers in **hsc percentage**

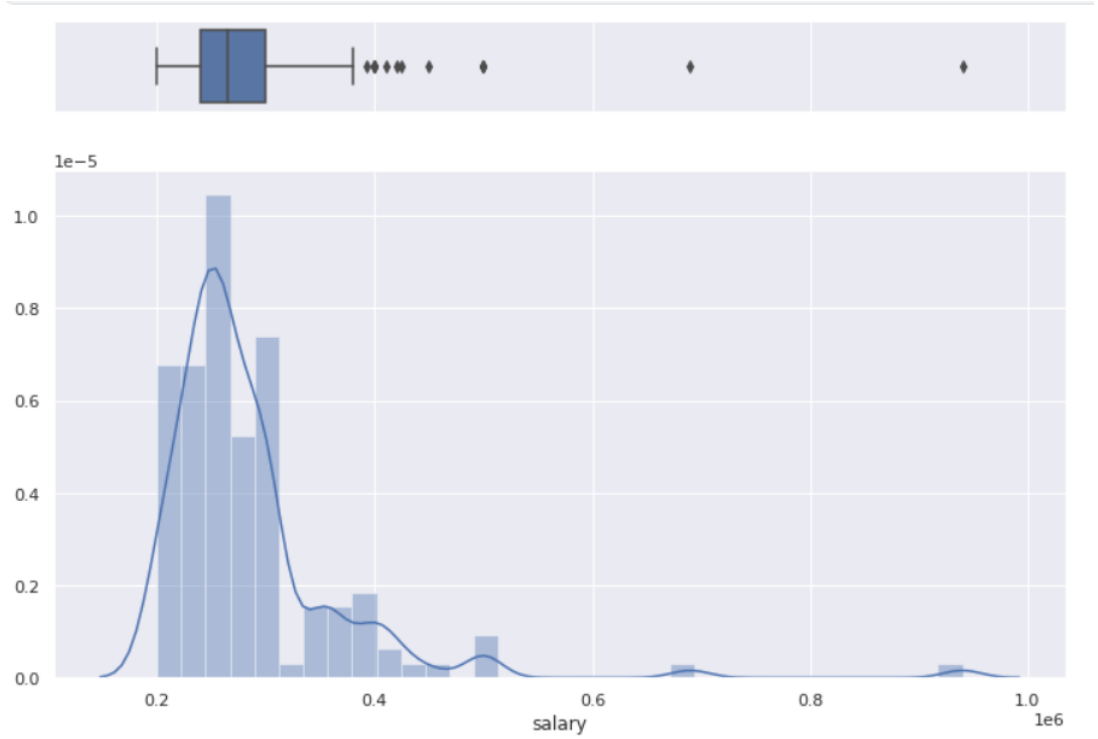


4. Count of categorical features - Countplot:



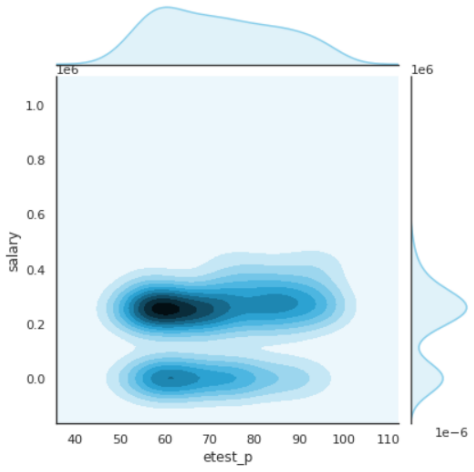
- We have **more male candidates** than female
- We have candidates who did **commerce** as their hsc course and as well as undergrad
- **Science background** candidates are the second highest in both cases
- Candidates from **Marketing and Finance** dual specialization are high
- Most of the candidates from our dataset **don't have any work experience**
- Most of the candidates from our dataset **got placed** in a company

5. Distribution Salary - Placed Students



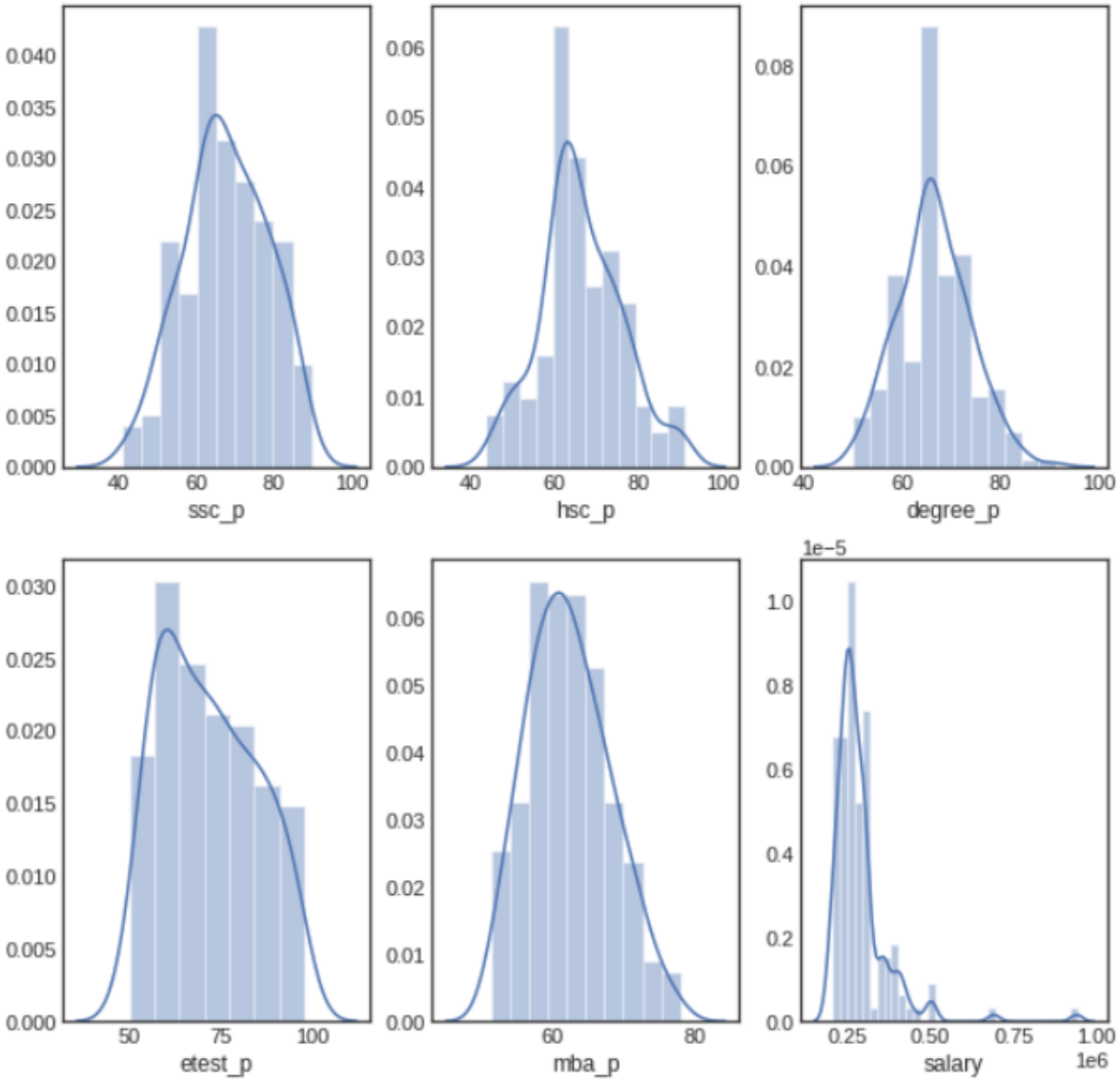
- Many candidates who got placed received package between **2L-4L PA**
- Only **one** candidate got around **10L PA**
- The **average** salary is a little **more than 2LPA**

6. Employability score vs Salary - Joint plot



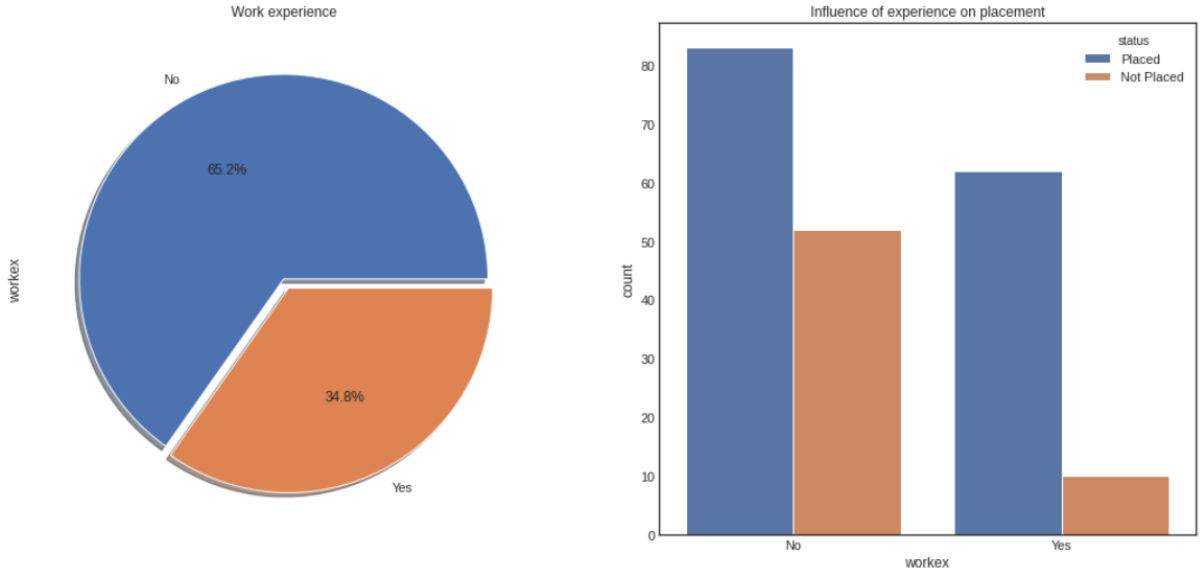
- Most of the candidates who scored around **60 per cent** got a decent package of b **3 lakhs PA**
- **Not** many candidates received a salary of **more than 4 lakhs PA**
- The bottom dense part shows the candidates who were **not placed**

7. Distribution of all percentages



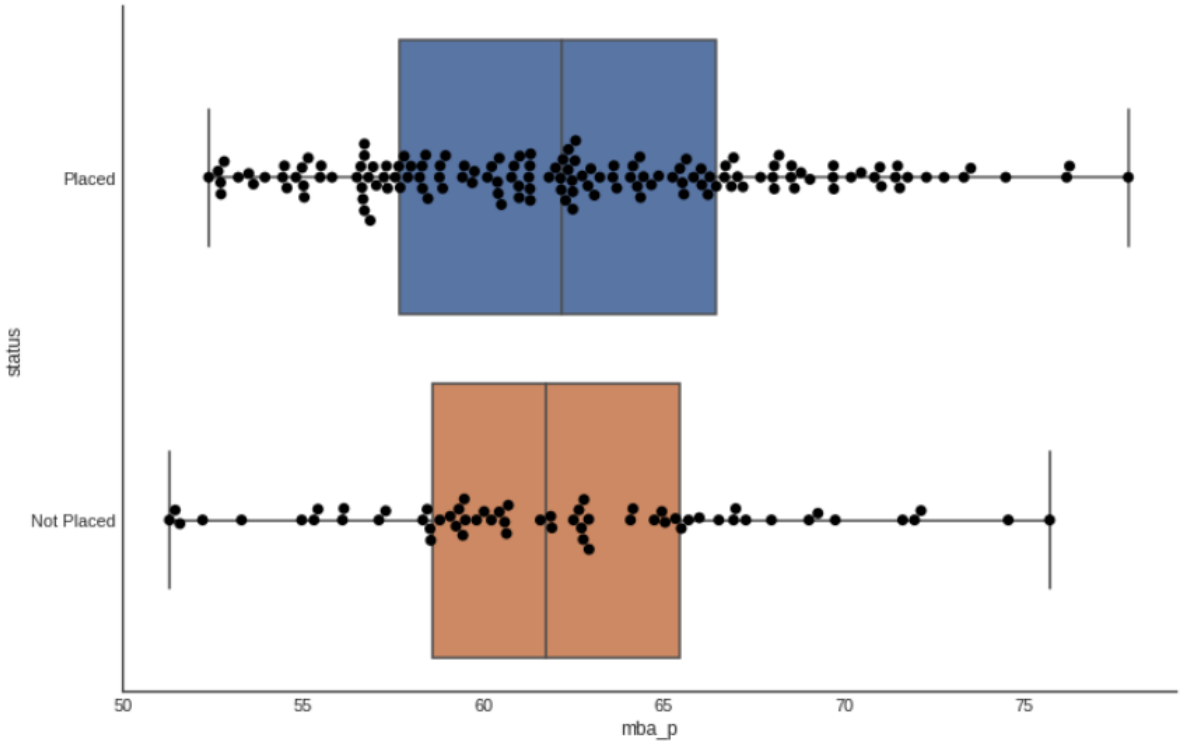
- All the distributions follow **normal distribution** except the salary feature
- Most of the candidate's **educational performances are between 60-80%**
- **Salary distribution got outliers** where few have got a salary of 7.5L and 10L PA

8. Work experience Vs Placement Status



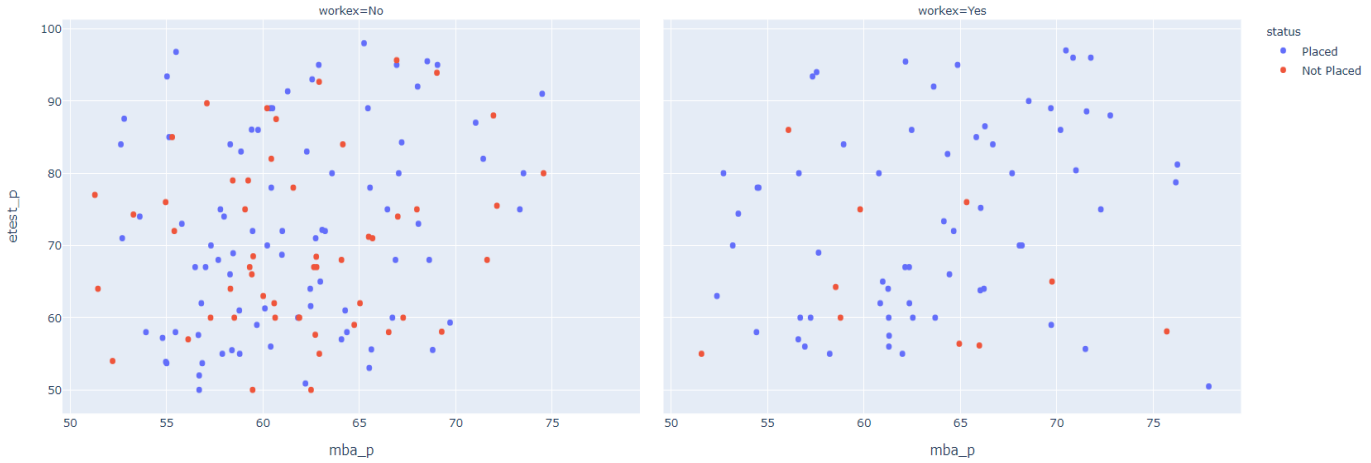
- We have nearly **66.2%** of candidates who never had any work experience
- Candidates who **never had work experience** have **got hired** more than the ones who had experience
- We can conclude that **work experience doesn't influence** a candidate in the recruitment process

9. MBA marks vs Placement Status



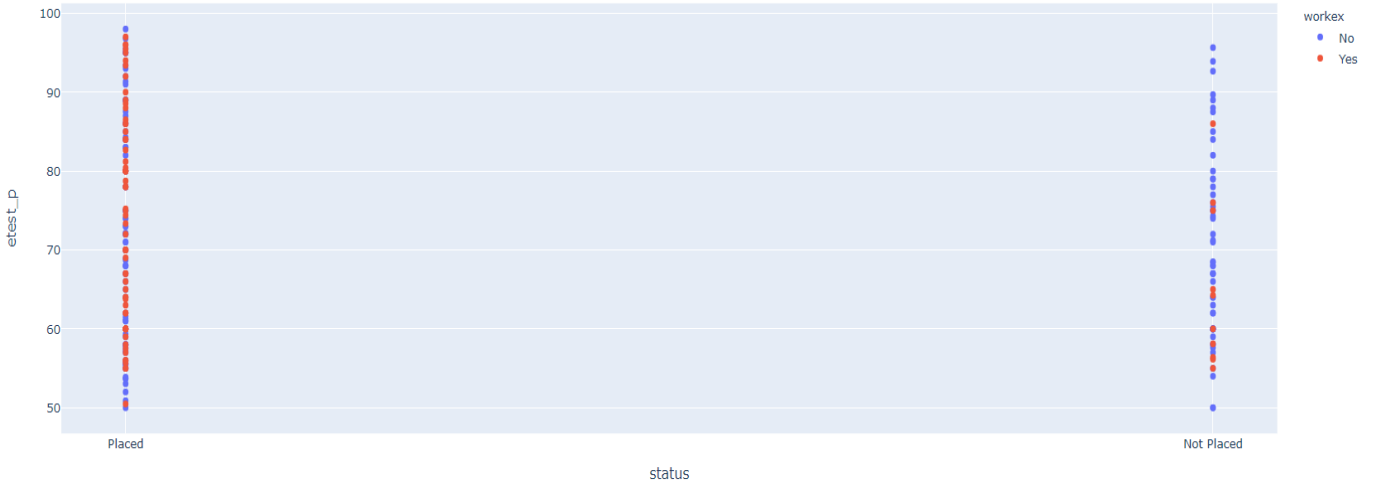
- Comparatively, there's a slight difference between the percentage scores between both the groups, But still placed candidates still have an upper hand when it comes to numbers as you can see in the swarm. So as per the plot, the **percentage does influence** the placement status

10. Does MBA percentage and Employability score correlate



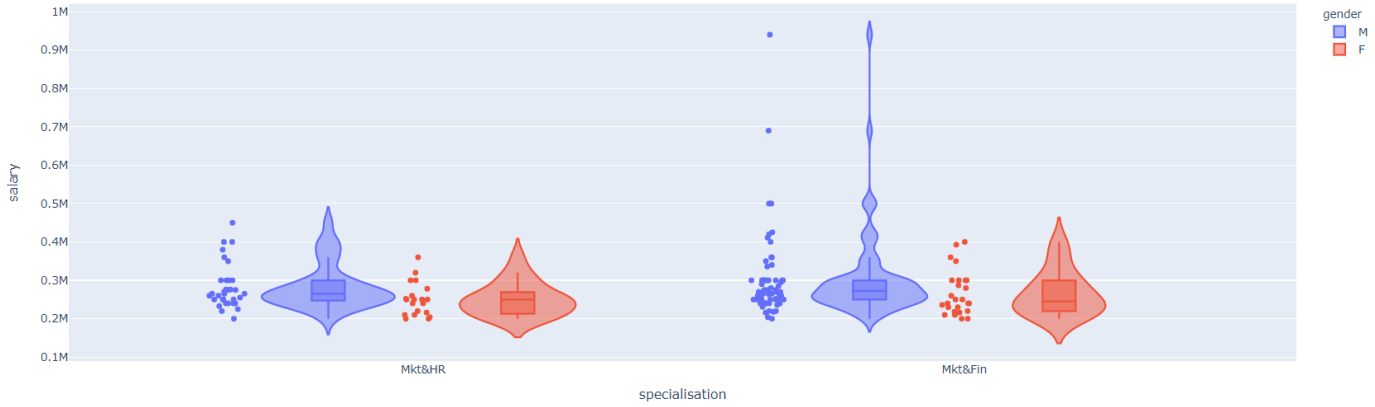
- There is **no relation** between mba percentage and employability test
- There are many candidates who **haven't got place**d when they don't have work experience
- Most of the candidates who performed better in both tests **have got placed**

11. Does Work experience and Status correlate



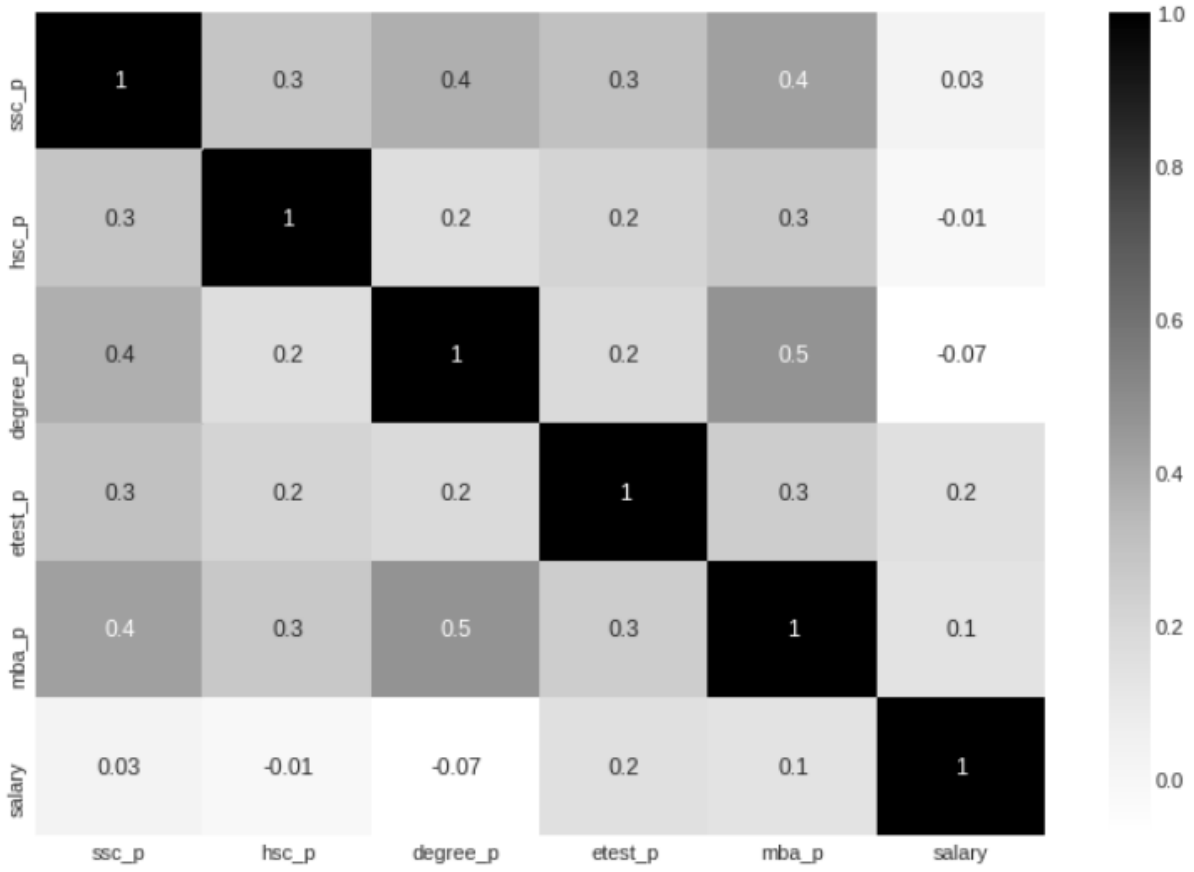
- There is a **relation** between workex and status.
- There are many candidates who **had similar e_test scores** but did not get placed when they don't have work experience
- Most of the candidates who had work experience **have got placed**

12. Is there any gender bias while offering remuneration



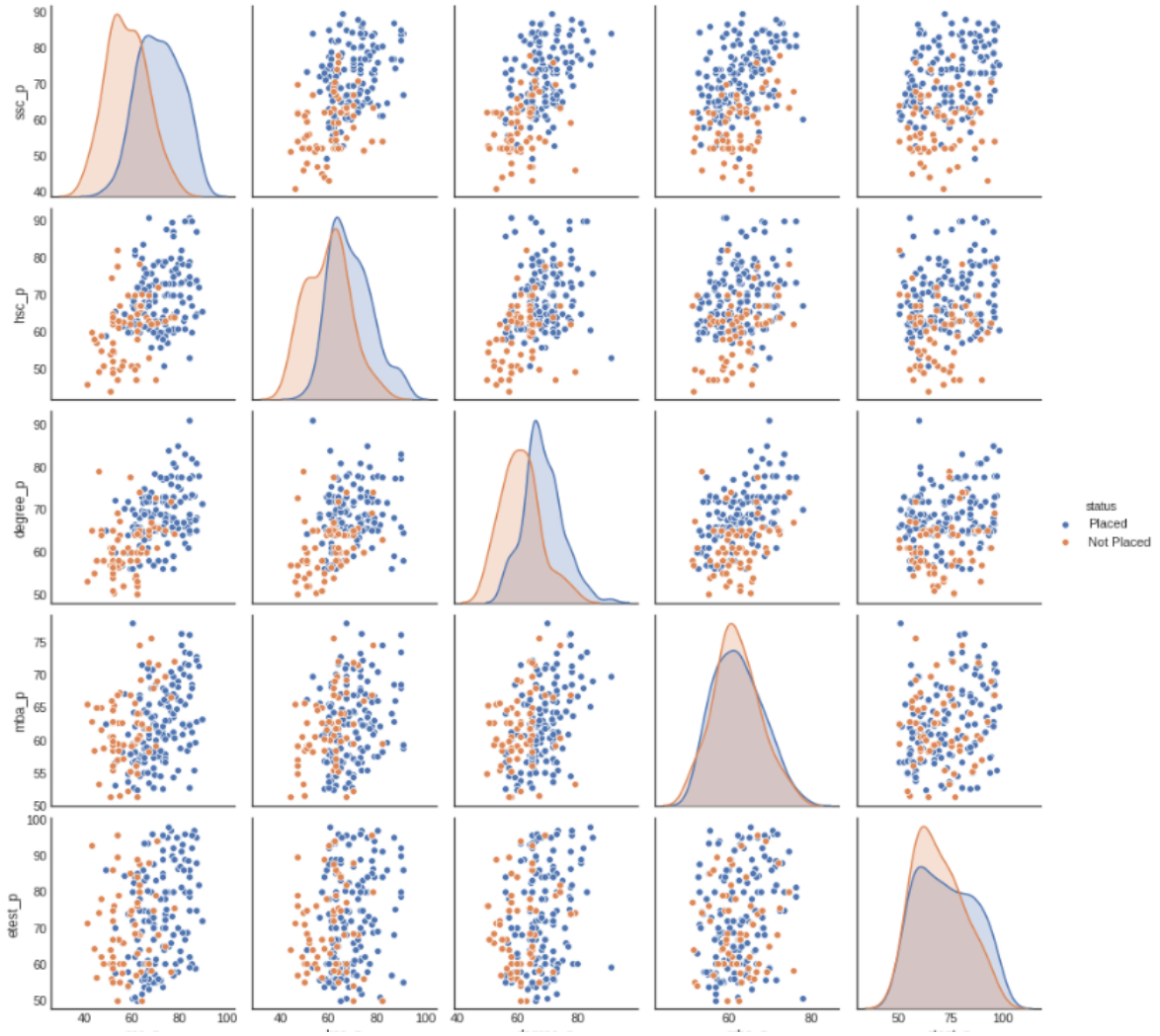
- The **top salaries were given to male**
- The **average salary** offered was also **higher for male**
- **More male candidates were placed** compared to female candidates

13. Correlation between academic percentages



- Candidates who were good in their academics performed well throughout school, undergrad, MBA and even employability test
- These percentages **don't have any influence over their salary**

14. Distribution of our data



- Candidates who have a **high scores in higher secondary and undergrad got placed**
- Whoever got **high scores in their schools got placed**
- Comparing the number of students who got placed candidates who got **good mba percentages and employability percentage**

15. Overall Result

- **Educational percentages** are highly influential for a candidate to get placed
- **Past work experience** doesn't influence much on your master final placements
- There is **no gender discrimination** while hiring, but higher packages were given to male
- Academic percentages have **no relation** towards salary package.

Problem Statement 2: Detecting Emotional Sentiment in Cartoons

Challenge Description:

Social media platforms are widely used by individuals and organizations to express emotions, opinions, and ideas. These platforms generate vast amounts of data, which can be analyzed to gain insights into user behaviour, preferences, and sentiment. Accurately classifying the sentiment of social media posts can provide valuable insights for businesses, individuals, and organizations to make informed decisions.

To accomplish this task, a customized private cartoon dataset (original images) of social media posts has been provided, which contains labels for each post's emotion category, such as:

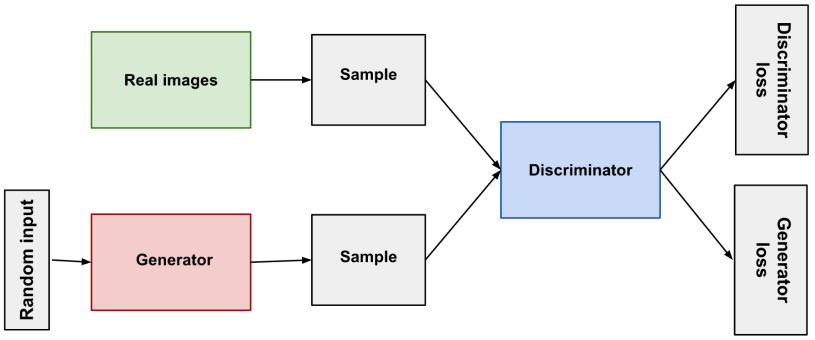
1. Happy
2. Angry
3. Sad
4. Neutral

The task is to build and fine-tune a machine-learning model that accurately classifies social media posts into their corresponding emotion categories, using synthetic images.

A. Synthetic Data Generation: Code Link: Problem 2 (Generate Data)

The synthetic images were created using a **GAN (Generative Adversarial Network)**. A GAN is a machine learning (ML) model in which two neural networks compete with each other by using deep learning methods to become more accurate in their predictions.

The two neural networks that make up a GAN are referred to as the **generator** and the **discriminator**. The **generator is a convolutional neural network**, and the **discriminator is a deconvolutional neural network**. The goal of the generator is to artificially manufacture outputs that could easily be mistaken for real data. The goal of the discriminator is to identify which of the outputs it receives have been artificially created.



1. Architecture of the Generator:

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, 100)	0
dense_3 (Dense)	(None, 4096)	413696
leaky_re_lu_2 (LeakyReLU)	(None, 4096)	0
reshape_1 (Reshape)	(None, 16, 16, 16)	0
conv2d_2 (Conv2D)	(None, 16, 16, 16)	6416
leaky_re_lu_3 (LeakyReLU)	(None, 16, 16, 16)	0
conv2d_transpose_1 (Conv2DTr	(None, 32, 32, 16)	4112
leaky_re_lu_4 (LeakyReLU)	(None, 32, 32, 16)	0
conv2d_3 (Conv2D)	(None, 32, 32, 16)	6416
leaky_re_lu_5 (LeakyReLU)	(None, 32, 32, 16)	0
conv2d_4 (Conv2D)	(None, 32, 32, 3)	2355
leaky_re_lu_6 (LeakyReLU)	(None, 32, 32, 3)	0

Total params: 432,995
Trainable params: 432,995
Non-trainable params: 0

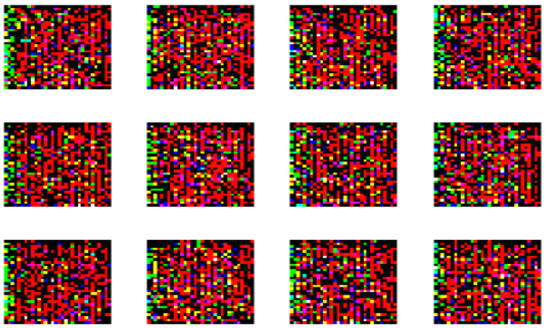
2. Architecture of the Discriminator:
The weights are set to be non-trainable.
Optimizer: RMSprop
Learning Rate: 0.0005

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32, 32, 3)	0
conv2d_1 (Conv2D)	(None, 30, 30, 16)	448
leaky_re_lu_1 (LeakyReLU)	(None, 30, 30, 16)	0
flatten_1 (Flatten)	(None, 14400)	0
dropout_1 (Dropout)	(None, 14400)	0
dense_1 (Dense)	(None, 10)	144010
dense_2 (Dense)	(None, 1)	11

Total params: 144,469
Trainable params: 144,469
Non-trainable params: 0

3. GAN model:
Combining the Generator and the Discriminator
Optimizer: RMSprop
Learning Rate: 0.0005

4. Initial random generated Image:



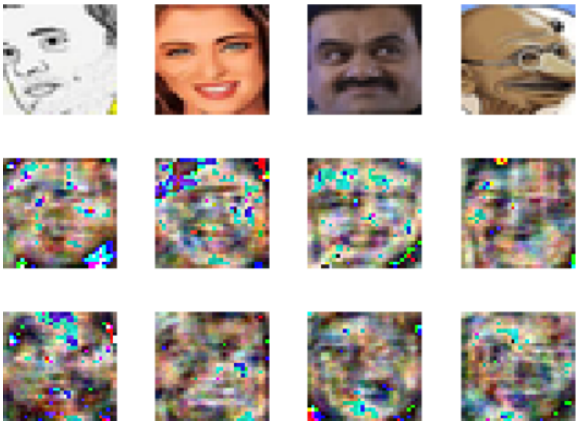
Latent Dimension: 100
Batch Size: 16
Height: 32
Width: 32

5. Model Training and Result:

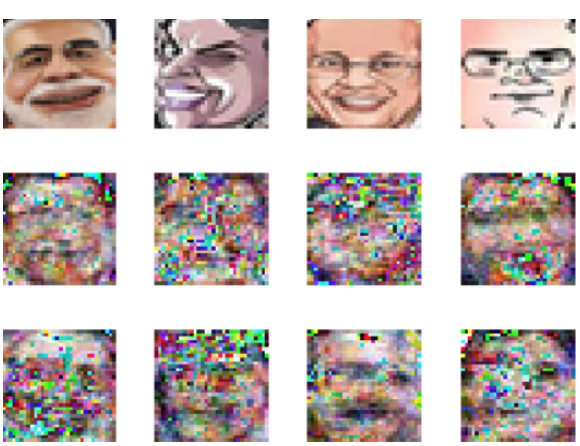
Total Iterations: 100000

Iteration: 500
Discriminator loss: 0.80
Adversarial loss: 5.37

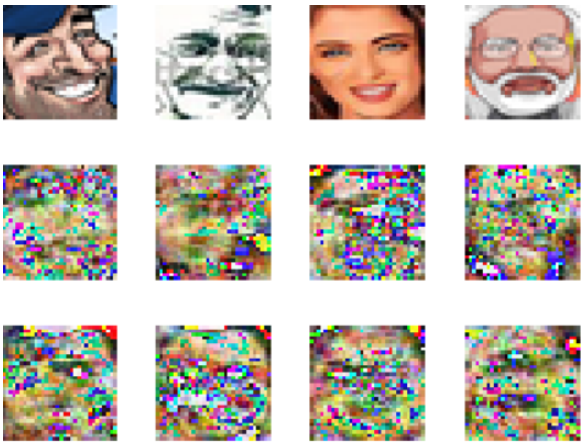
Iteration: 2000
Discriminator loss: 0.80
Adversarial loss: 1.39



Iteration: 8000
Discriminator loss: 1.05
Adversarial loss: 1.07

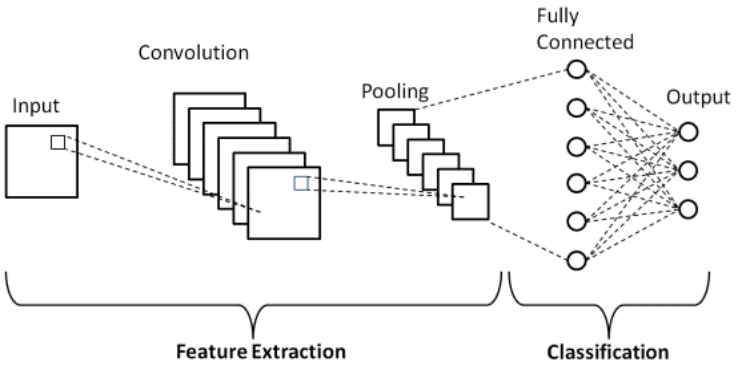


Iteration: 10000
Discriminator loss: 1.11
Adversarial loss: 1.02



B. Classification Model: [Code Link: Problem 2 (Classification)]

The objective of performing multiclass classification was done using a **CNN (Convolutional Neural Network)** model. A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data.



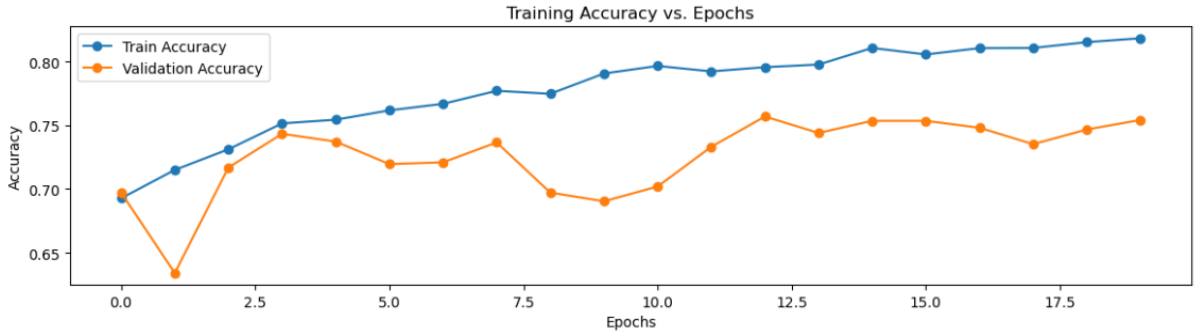
1. Image Augmentation:
The image augmentation technique is a great way to expand the size of the dataset.

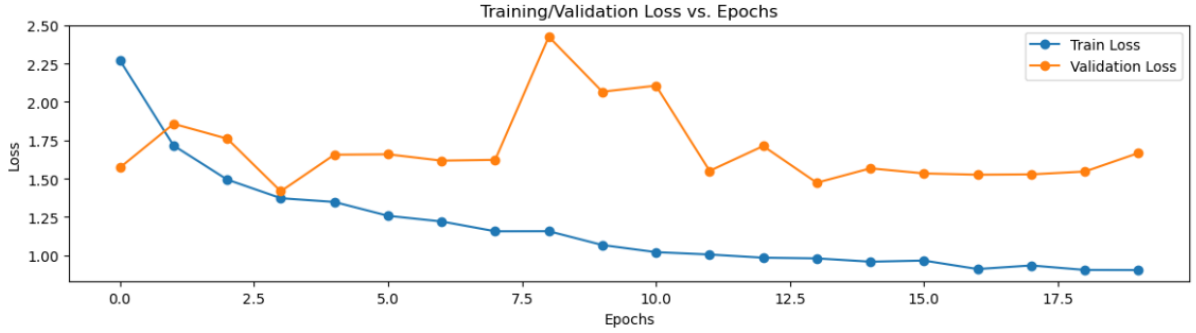
Original count of samples: 354
Augmented count of samples: 1810

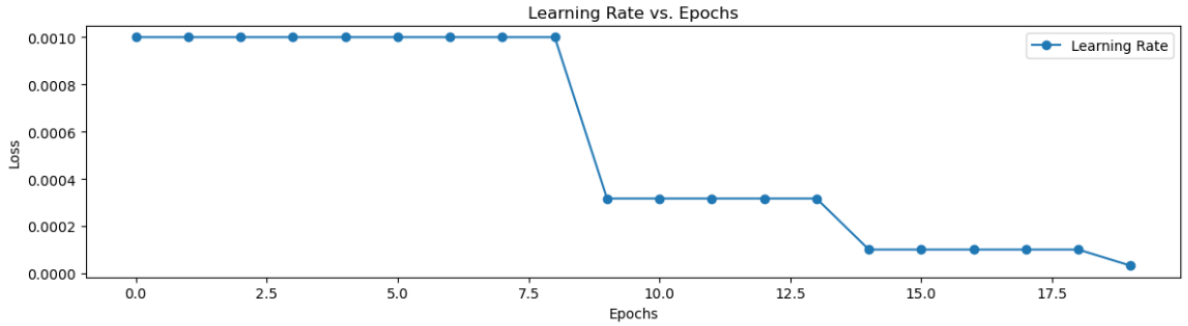
2. Architecture of the CNN model:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 146, 146, 32)      2432

 activation (Activation)     (None, 146, 146, 32)      0

 max_pooling2d (MaxPooling2D  (None, 73, 73, 32)       0
 )

 batch_normalization (BatchN  (None, 73, 73, 32)       128
 ormalization)

 conv2d_1 (Conv2D)           (None, 71, 71, 64)        18496

 activation_1 (Activation)   (None, 71, 71, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 35, 35, 64)       0
 2D)

 batch_normalization_1 (Batc  (None, 35, 35, 64)       256
 hNormalization)

 conv2d_2 (Conv2D)           (None, 33, 33, 32)        18464

 activation_2 (Activation)   (None, 33, 33, 32)        0

 max_pooling2d_2 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 batch_normalization_2 (Batc  (None, 16, 16, 32)       128
 hNormalization)

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 64)                524352

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 4)                 260
```

3. Training Results:





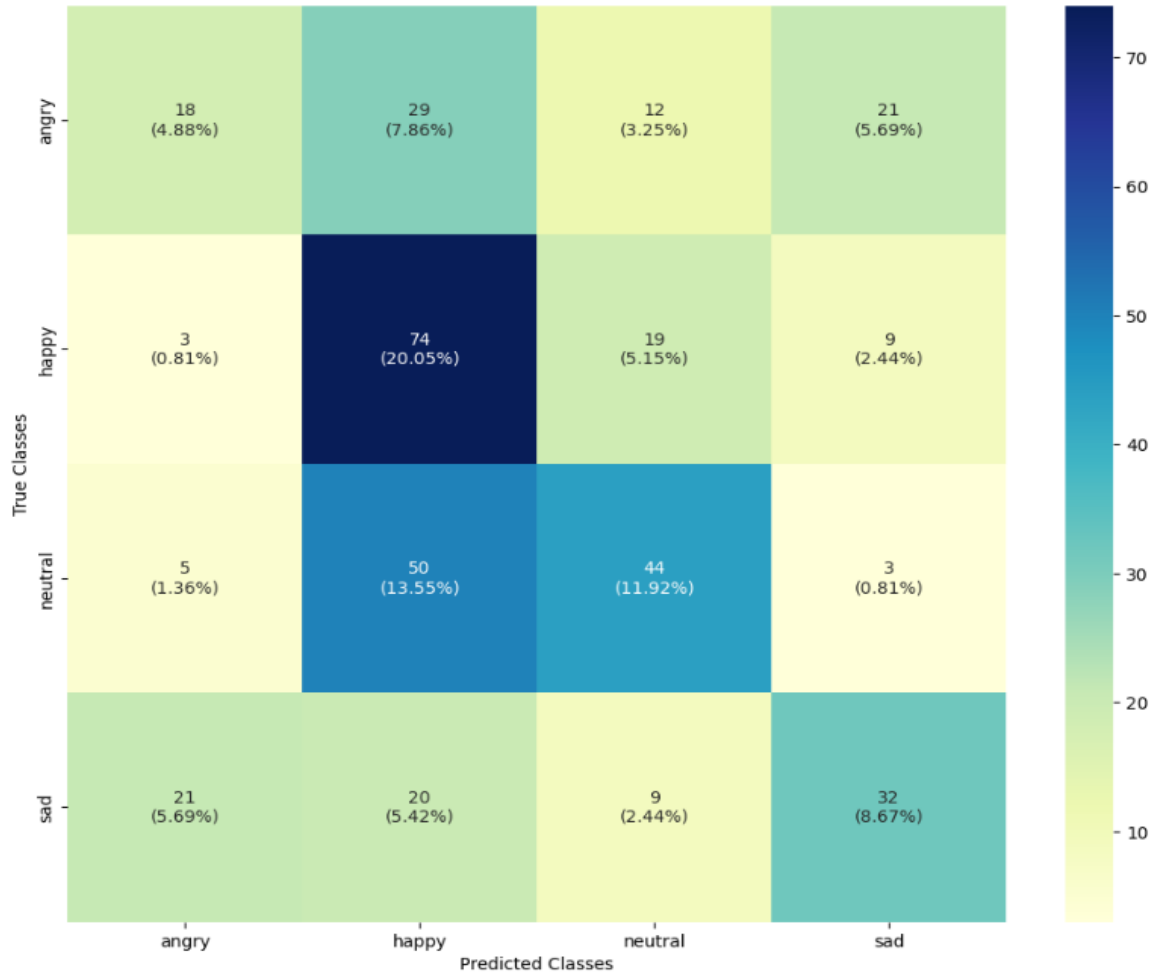


4. Sample Predictions:

5. Test Results:

Test Loss:	1.66
Test Accuracy:	0.75
Test Precision:	0.52
Test Recall:	0.35
Test AUC:	0.73
Test F1 score:	0.39

6. Confusion Matrix:



7. Classification Report:

	precision	recall	f1-score	support
angry	0.38	0.23	0.28	80
happy	0.43	0.70	0.53	105
neutral	0.52	0.43	0.47	102
sad	0.49	0.39	0.44	82
accuracy			0.46	369
macro avg	0.46	0.44	0.43	369
weighted avg	0.46	0.46	0.44	369