PREDICTING COLLEGE PLACEMENTS

Data Mining Project

- Smrati Sharma
- Soumya Upadhyaya
- Rohit Singh
- Pratham Arya

Introduction

ALL Educational Institutes aims at helping their students to get a well-paid job through their placement cell. One of the biggest challenges that higher learning institutes face these days is to uplift the placement performance of scholars.

The goal
student wi
on varia

Introduction

The goal of this system is to predict whether the student will get a campus placement or not based on various parameters such as Gender, SSC percentage, HSC percentage, HSC stream, degree percentage, degree type, work experience & e-test percentage.

helping their
rough their
hallenges that
se days is to

This res
machin
Decision

Introduction

This research focuses on various algorithms of machine learning such as Logistic Regression, Decision Tree, K-Nearest Neighbours in order to produce economical and correct results for campus placement prediction. This system follows a Supervised Machine Learning Approach as it uses class labelled data for training the classification algorithm.

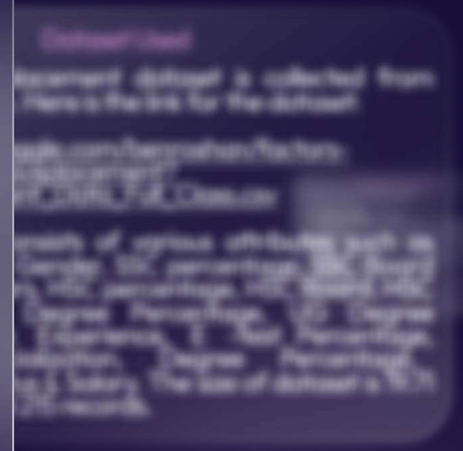
hether the
r not based
der. SSC

General Information

Technique Used:

Classification

Classification in data mining is a common technique that separates data points into different classes. It allows you to organize data sets of all sorts, including complex and large datasets as well as small and simple ones.



General Information

Dataset Used

The campus placement dataset is collected from Kaggle website. Here is the link for the dataset:

https://www.kaggle.com/benroshan/factors-affecting-campusplacement?select=Placement_Data_Full_Class.csv

The dataset consists of various attributes such as Serial Number, Gender, SSC percentage, SSC Board - Central/ Others, HSC percentage, HSC Board, HSC Specialization, Degree Percentage, UG Degree Stream, Work Experience, E -test Percentage, Degree Specialization, Degree Percentage, Placement Status & Salary. The size of dataset is 19.71 KB & it has total 215 records.

General Information

Description Of Dataset

Some Attributes of Data set are:

1. Serial Number
2. Gender: Male or Female
3. SSC Percentage: Percentage of 9th and 10th standard
4. HSC Percentage: Percentage of 11th and 12th standard
5. Degree Percentage: Percentage of Under-graduation.
6. Degree Type
7. Work Experience
8. Employability-test percentage

Data Preprocessing

We have used 4 Techniques to process our data.

The Techniques used are:

- Dimensionality reduction
- Handling NaN values
- Outlier Handling
- Handling Categorical Data

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Categorical Data

Data Preprocessing

Dimensionality Reduction

Dimensionality reduction is defined as a method of reducing variables in a training dataset.

In our DataSet, we have dropped Three Variables which include 'sl_no' (serial number) ,'ssc_b' (secondary School Certificate Board),'hsc_b' (high School Certificate Board).

These Variables contain Redundant Values which do not contribute to our model.

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Categorical Data

Data Preprocessing

Dimensionality Reduction

Dimensionality reduction is defined as a method of reducing variables in a training dataset.

In our DataSet, we have dropped Three Variables which include 'sl_no' (serial number) ,'ssc_b' (secondary School Certificate Board),'hsc_b' (secondary School Certificate Board).

These Variables contain Redundant Values which do not contribute to our model.

Dataset Before Dimensionality Reduction:

```
In [4]: 1 data.head(10)
```

Out[4]:

	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	55.00	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.50	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.00	Mkt&Fin	57.80	Placed	250000.0

Code:

```
data.drop(['sl_no','ssc_b','hsc_b'], axis = 1 , inplace = True)
```

Dataset After Dimensionality Reduction:

```
1 data.head()
```

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	M	67.00	91.00	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	M	79.33	78.33	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	M	65.00	68.00	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Categorical Data

Data Preprocessing

Handling NaN Values

In our dataset NaN values are present only in the Salary column as these values correspond to the students who didn't get placed in any placement drive. So it is assumed that the missing values in Salary Column are Zero & replaced them by zero using fillna(0,inplace=True) function in Python.

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Categorical Data

Data Preprocessing

Outlier Handling

Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is Vastly Larger Or Smaller than the remaining values in the set. In our DataSet, the variables ‘hsc_p’ and ‘degree_p’ contains Outliers. We used BoxPlot to detect outliers and IQR method to remove them from our DataSet.

Dimensionality Reduction

Handling NaN Values

Outlier Handling

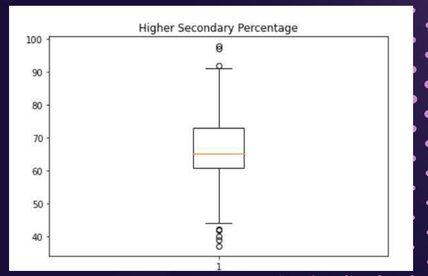
Handling Catergorical Data

Data Preprocessing

Outlier Handling

Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is Vastly Larger Or Smaller than the remaining values in the set. In our DataSet, the variables ‘hsc_p’ and ‘degree_p’ contains Outliers. We used BoxPlot to detect outliers and IQR method to remove them from our DataSet.

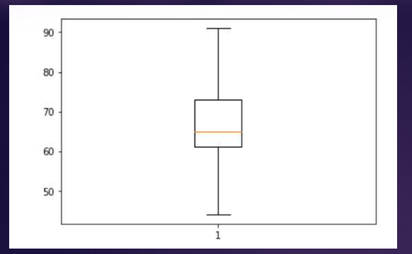
Box Plot Graph Before Outllier Handling



Code:

```
Q1 = placement['hsc_p'].quantile(0.25)
Q3 = placement['hsc_p'].quantile(0.75)
IQR = Q3-Q1
filter = (placement['hsc_p']>= Q1 - 1.5 * IQR) & (placement['hsc_p']<= Q3+ 1.5*IQR)
placement_filtered = placement.loc[filter]
plt.boxplot(placement_filtered['hsc_p'])
```

Box Plot Graph After Outllier Handling



Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Catergorical Data

Data Preprocessing

Handling Categorical Data

Since we cannot deal with Categorical Values directly, mapping is done for attributes having categorical values. Gender attribute has values M (Male) & Female (M). Here, M is replaced by 0 & F is replaced by 1. Work Experience attribute has values ‘Yes’ & ‘No’. Here, ‘Yes’ is replaced by 1 and ‘No’ is replaced by 0. Degree specialization attribute has values ‘Marketing & Finance’ & ‘Marketing & HR’. Here, ‘Marketing & Finance’ is replaced by 1 and ‘Marketing & HR’ is replaced by 0. Status attribute has values ‘Placed’ and ‘Not Placed’. Here, ‘Placed’ is replaced by 1 and ‘Not Placed’ is replaced by 0. This is achieved through Map Function in Python.

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Catergorical Data

Data Preprocessing

Handling Categorical Data

Since we cannot deal with Categorical Values directly, mapping is done for attributes having categorical values. Gender attribute has values M (Male) & Female (M). Here, M is replaced by 0 & F is replaced by 1. Work Experience attribute has values ‘Yes’ & ‘No’. Here, ‘Yes’ is replaced by 1 and ‘No’ is replaced by 0. Degree specialization attribute has values ‘Marketing & Finance’ & ‘Marketing & HR’. Here, ‘Marketing & Finance’ is replaced by 1 and ‘Marketing & HR’ is replaced by 0. Status attribute has values ‘Placed’ and ‘Not Placed’. Here, ‘Placed’ is replaced by 1 and ‘Not Placed’ is replaced by 0. This is achieved through Map Function in Python.

Label Encoding

We used label encoding in the columns 'gender', 'workex', 'specialisation', 'status'

One Hot Encoding

We used One Hot encoding in the columns 'hsc_s', 'degree_t'.

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Catergorical Data

Data Preprocessing

Label Encoding

```
from sklearn.preprocessing import LabelEncoder
object_cols= ['gender','workex','specialisation','status']
label_encoder = LabelEncoder()
for col in object_cols:
placement_filtered[col]= label_encoder.fit_transform(placement_filtered[col])
placement_filtered.head(10)
```

DataSet After Label Encoding:

7]:

	gender	ssc_p	hsc_p	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	67.00	91.00	Commerce	58.00	Sci&Tech	0	55.00	1	58.80	1	270000.0
1	1	79.33	78.33	Science	77.48	Sci&Tech	1	86.50	0	66.28	1	200000.0
2	1	65.00	68.00	Arts	64.00	Comm&Mgmt	0	75.00	0	57.80	1	250000.0
3	1	56.00	52.00	Science	52.00	Sci&Tech	0	66.00	1	59.43	0	0.0
4	1	85.80	73.60	Commerce	73.30	Comm&Mgmt	0	96.80	0	55.50	1	425000.0
5	1	55.00	49.80	Science	67.25	Sci&Tech	1	55.00	0	51.58	0	0.0
6	0	46.00	49.20	Commerce	79.00	Comm&Mgmt	0	74.28	0	53.29	0	0.0
7	1	82.00	64.00	Science	66.00	Sci&Tech	1	67.00	0	62.14	1	252000.0
8	1	73.00	79.00	Commerce	72.00	Comm&Mgmt	0	91.34	0	61.29	1	231000.0
9	1	58.00	70.00	Commerce	61.00	Comm&Mgmt	0	54.00	0	52.21	0	0.0

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Catergorical Data

Data Preprocessing

One Hot Encoding

```
dummy_hsc_s=pd.get_dummies(placement_filtered['hsc_s'], prefix = 'dummy')

dummy_degree_t=pd.get_dummies(placement_filtered['degree_t'], prefix = 'dummy')

placement_coded = pd.concat([placement_filtered , dummy_hsc_s , dummy_degree_t],axis = 1)

placement_coded.drop(['hsc_s','degree_t','salary'],axis = 1 , inplace = True)

placement_coded.head()
```

DataSet After One Hot Encoding:

e_p	workex	etest_p	specialisation	mba_p	status	dummy_Arts	dummy_Commerce	dummy_Science	dummy_Comm&Mgmt	dummy_Others	dummy_Sci&Tech
8.00	0	55.0	1	58.80	1	0	1	0	0	0	1
7.48	1	86.5	0	66.28	1	0	0	1	0	0	1
4.00	0	75.0	0	57.80	1	1	0	0	1	0	0
2.00	0	66.0	1	59.43	0	0	0	1	0	0	1
3.30	0	96.8	0	55.50	1	0	1	0	1	0	0

Dimensionality Reduction

Handling NaN Values

Outlier Handling

Handling Catergorical Data

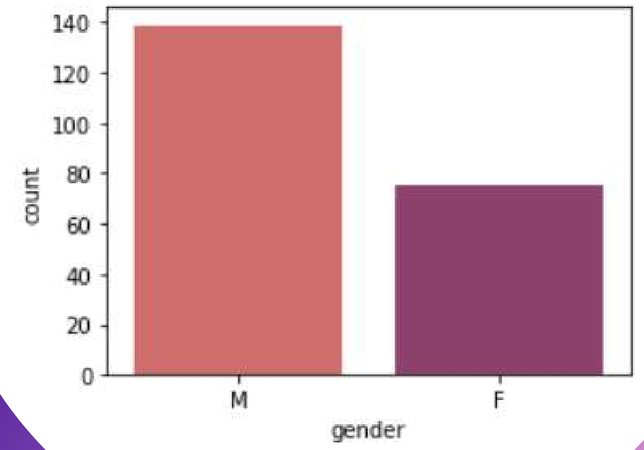
Data Visualization

Data visualization is the Graphical Representation of information and data.

We have used various Data Visualization techniques to analyze the data effectively.

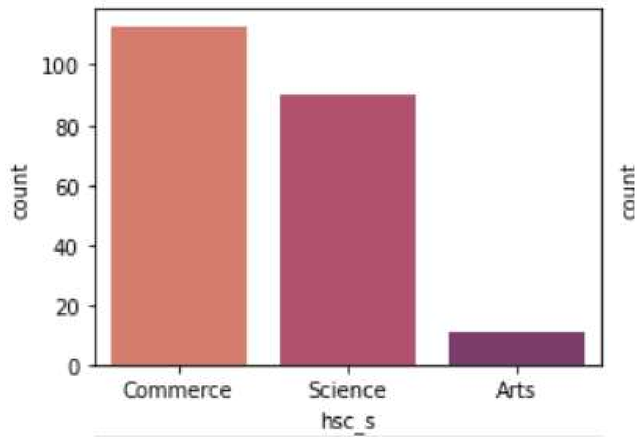
Data Visualization

Visualizing each column



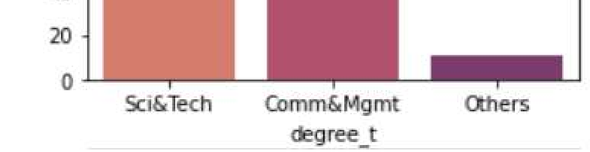
Data Visualization

Visualizing each column



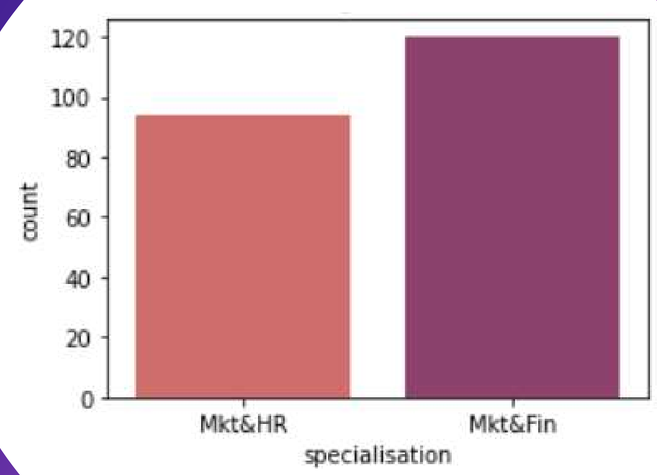
Data Visualization

Visualizing each column



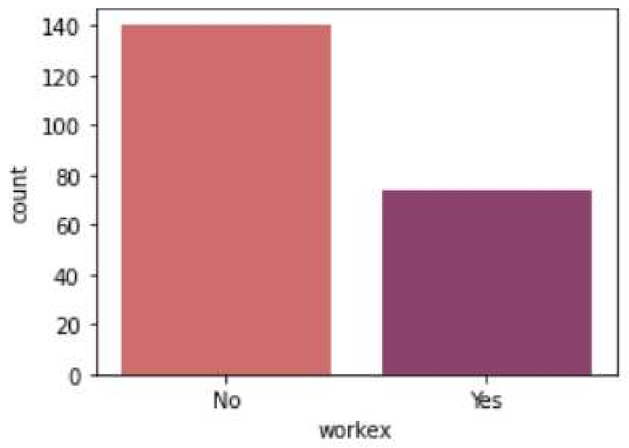
Data Visualization

Visualizing each column



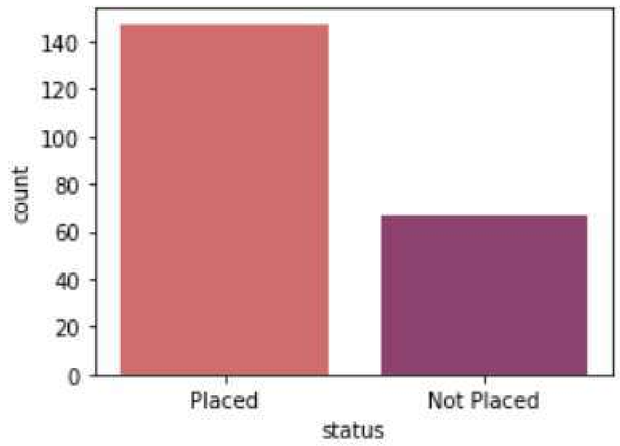
Data Visualization

Visualizing each column



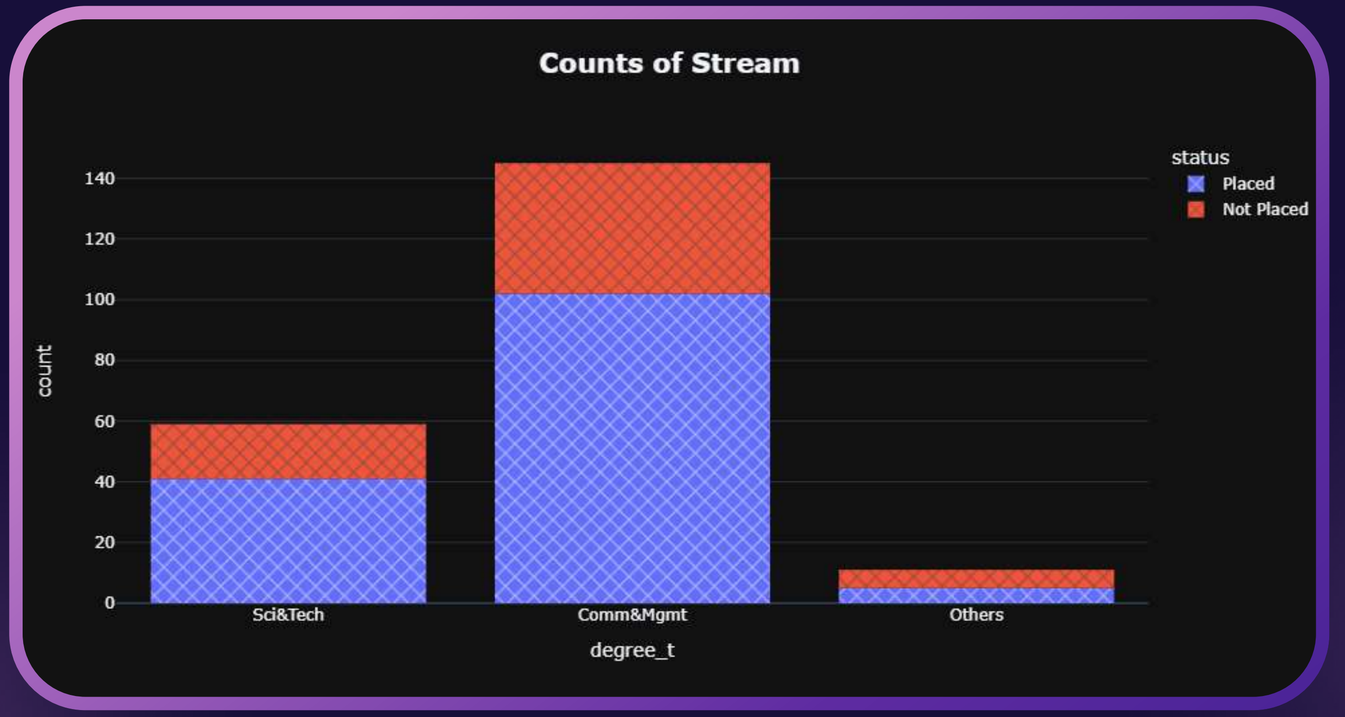
Data Visualization

Visualizing each column



Data Visualization

Visualizing Which Stream Placed The Higher Number Of Students.

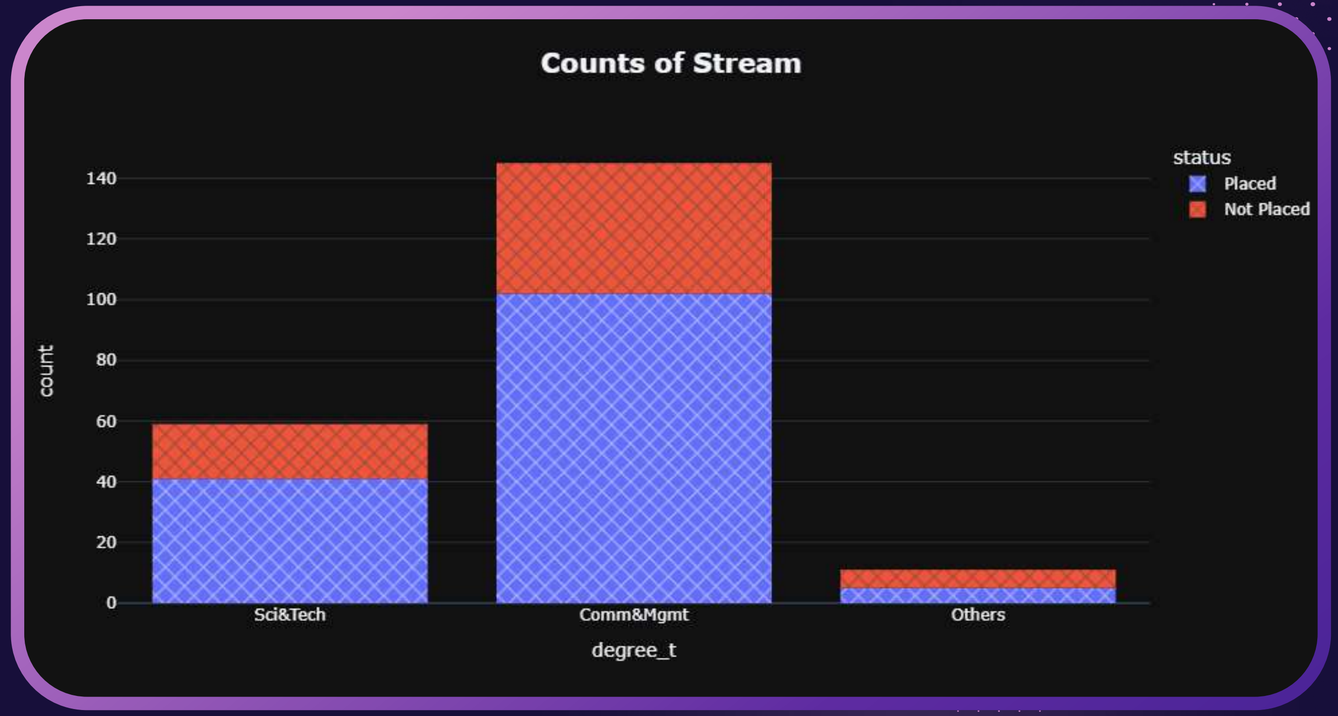


Data Visualization

Visualizing Which Stream Placed The Higher Number Of Students.

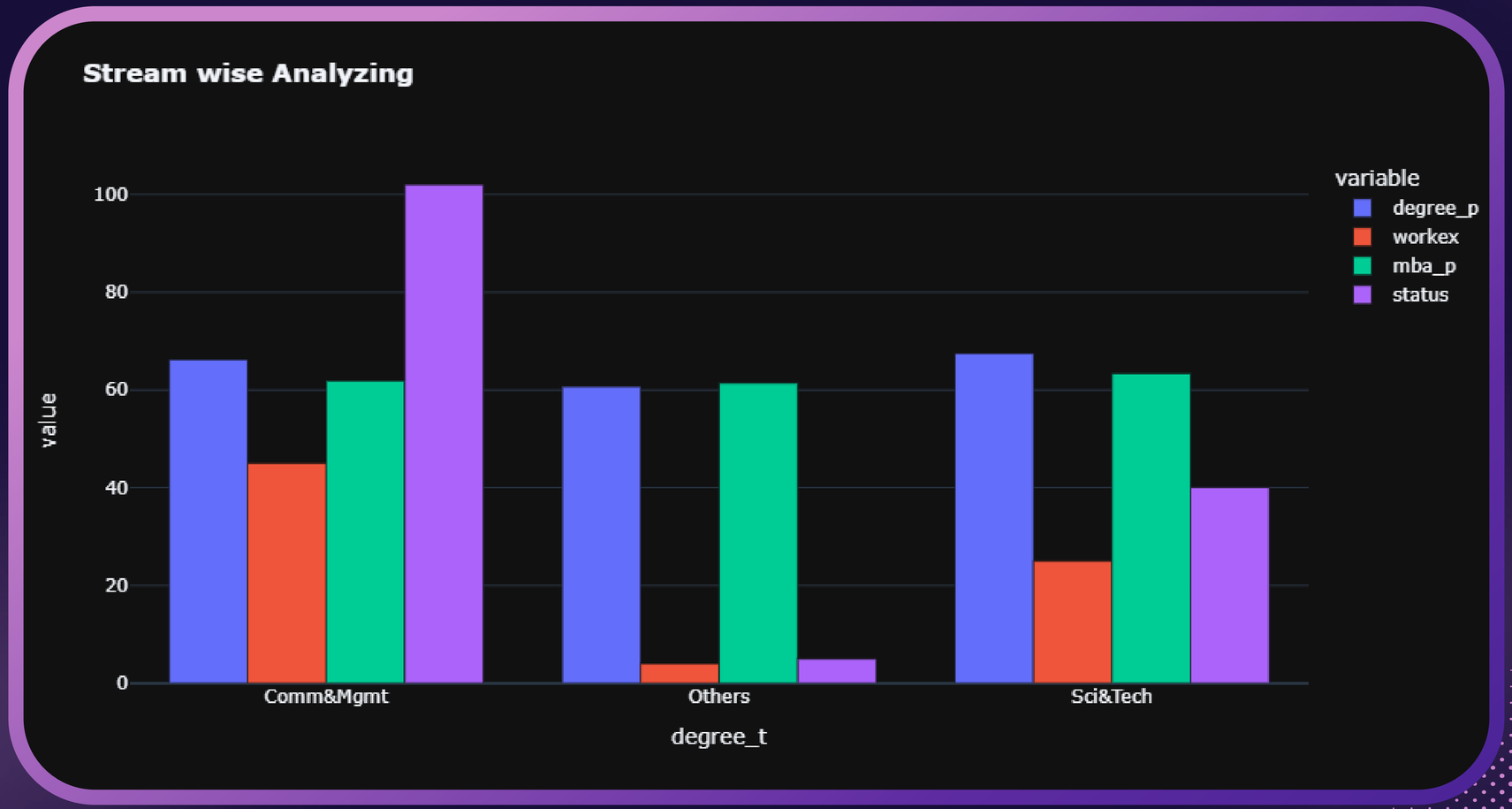
Code :

```
import plotly.express as px
fig = px.histogram(data_frame = data,
x = "degree_f",
        color="status", title="<b>Counts
of Stream</b>",
        pattern_shape_sequence=['x'],
        template='plotly_dark')


fig.update_layout(title_x = 0.5,
title_font = dict(size = 20),
uniformtext_minsize = 15)
fig.show()
```



Data Visualization

Stream Wise Analyzing

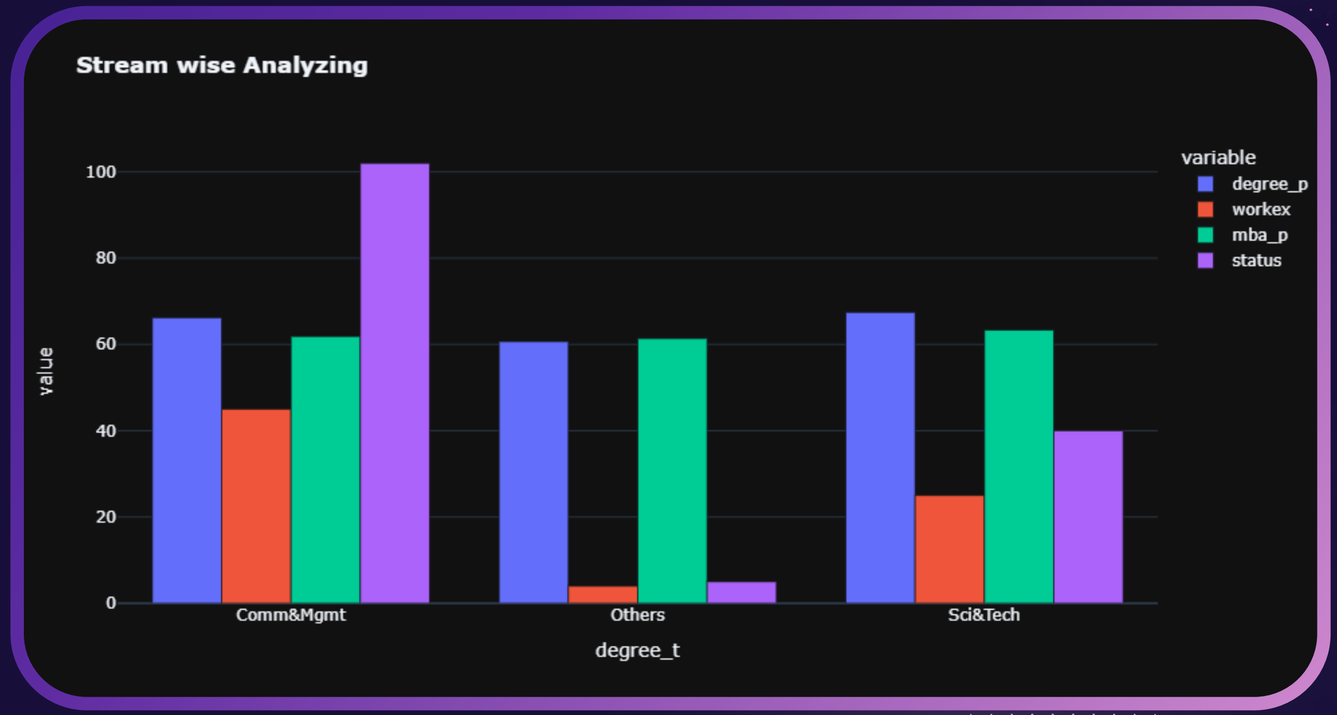


Data Visualization

Stream Wise Analyzing

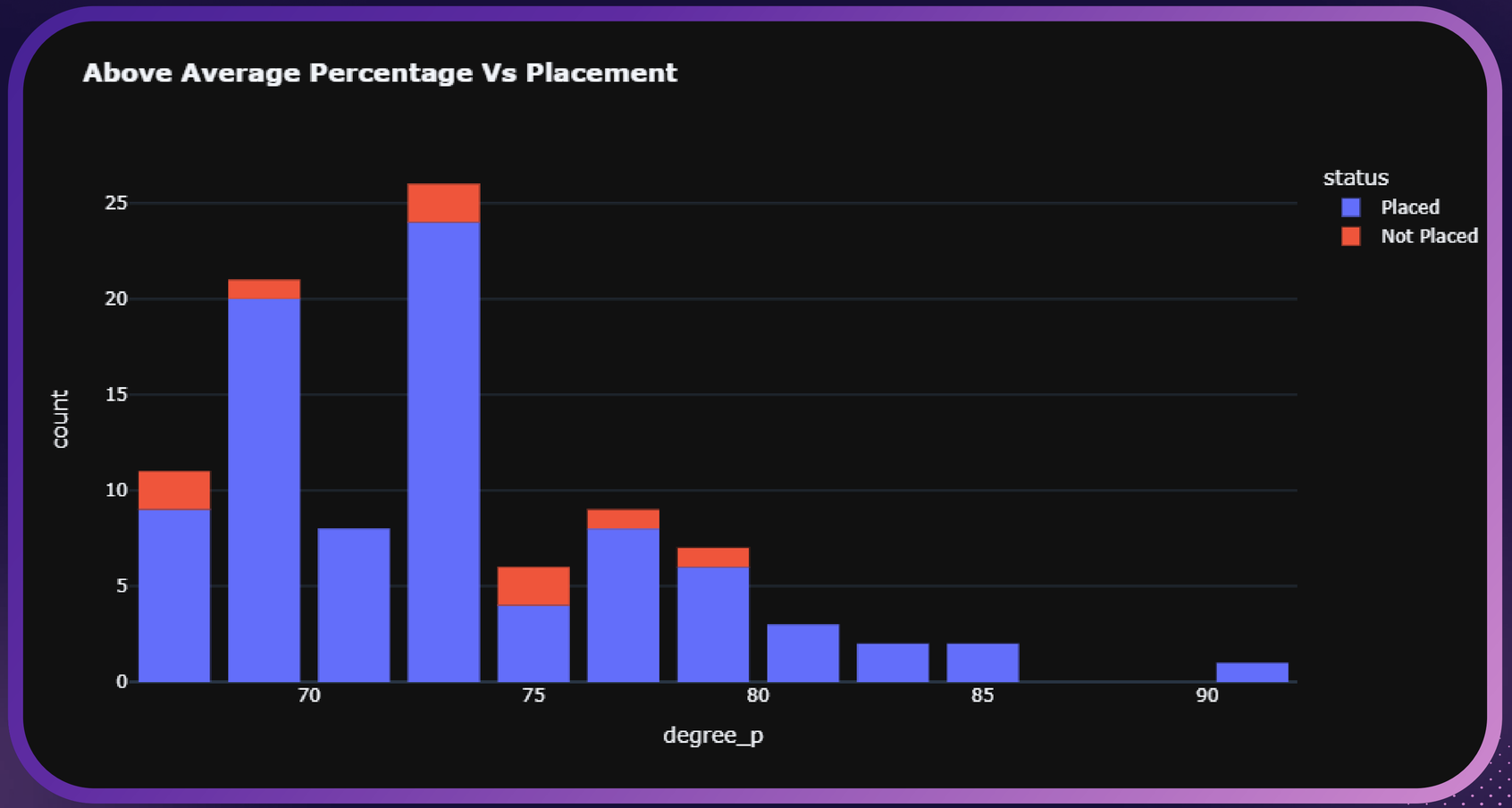
Code :

```
stream_wise =
placement_filtered.groupby('d
egree_t').agg({'degree_p'
:'mean', 'workex' : 'sum',
"mba_p":'mean','status':'sum'})


stream_wise.style.highlight_m
ax()
```



Data Visualization

Above Average Percentage Vs. Placements

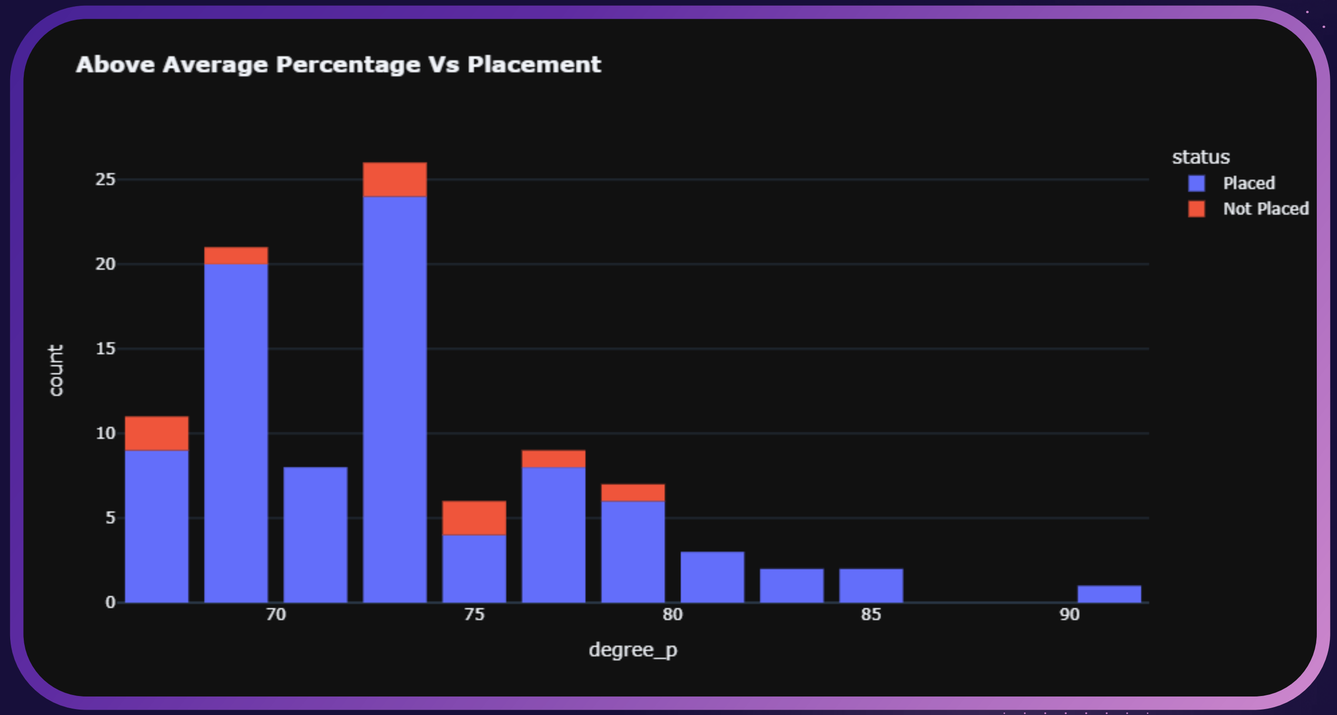


Data Visualization

Above Average Percentage Vs. Placements

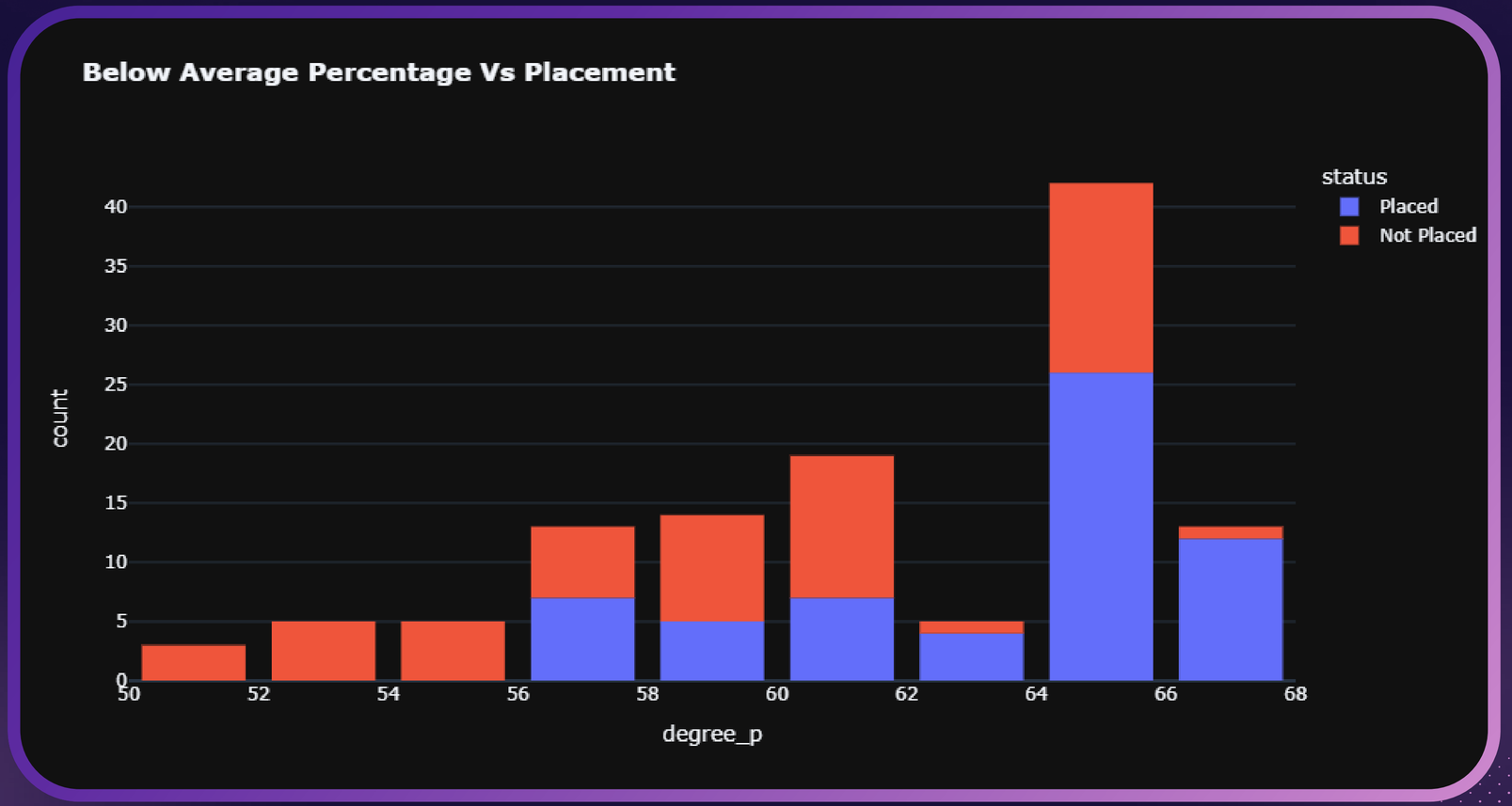
Code :

```
degree_p_above_avg=placement[placement['degree_p'] > placement['degree_p'].mean()]
degree_p_above_avg

fig = px.histogram(data_frame=degree_p_above_avg, x = 'degree_p', color='status', title = "<b>Above Average Percentage Vs Placement</b>", template='plotly')

fig.update_layout(bargap=0.2)

fig.show()
```



Data Visualization

Below Average Percentage Vs. Placements

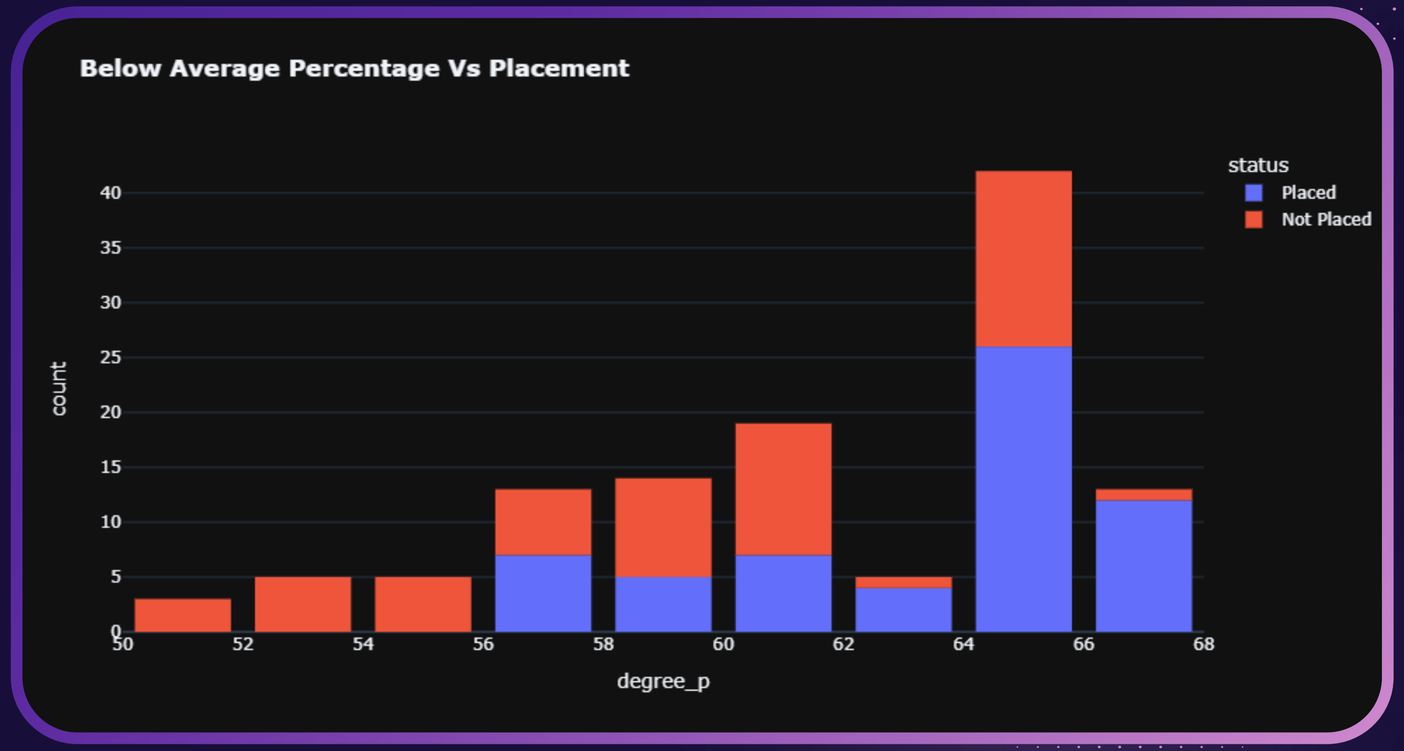


Data Visualization

Below Average Percentage Vs. Placements

Code :

```
degree_p_below_avg=placement[placement['
degree_p'] < placement['degree_p'].mean()]
degree_p_below_avg

fig =
px.histogram(data_frame=degree_p_below_
avg,  x = 'degree_p',  color='status', title =
"<b>Below Average Percentage Vs
Placement</b>",  template='plotly')

fig.update_layout(bargap=0.2)

fig.show()
```



Splitting Data

Data splitting is when data is divided into Two Or More Subsets. Typically, with a two-part split, one part is used to evaluate or Test the data and the other to Train The Model.

Here, data is divided into two parts i.e. training data & testing data. Where 80 % Data Is Taken For Training our machine learning algorithm and remaining 20 % data is used for testing whether our trained machine learning model is working correctly or not.

Classification Techniques

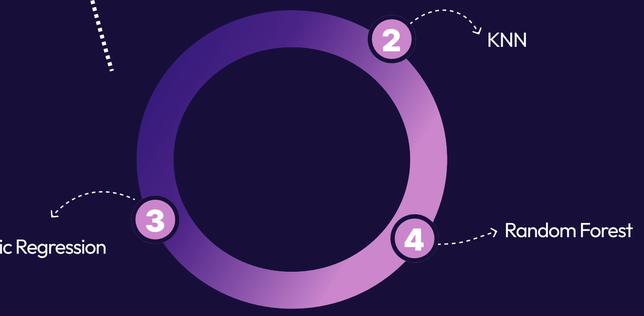
1. Decision Tree
2. KNN
3. Logistic Regression
4. Random Forest

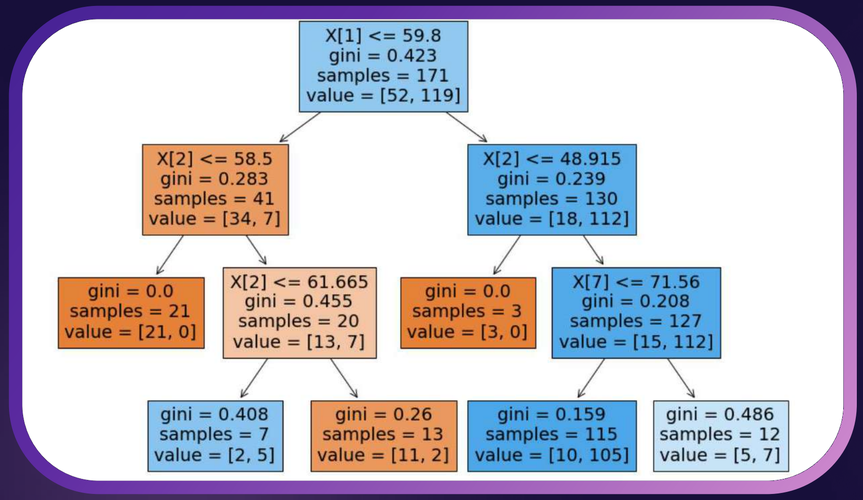
Classification Techniques

1 DECISION TREE

- A decision tree is a Graph like a tree where nodes represent the position where we select the feature and ask a question, Edges Represent The Answers of the question; and the Leaves Represent The Final Output Or Label of the class.

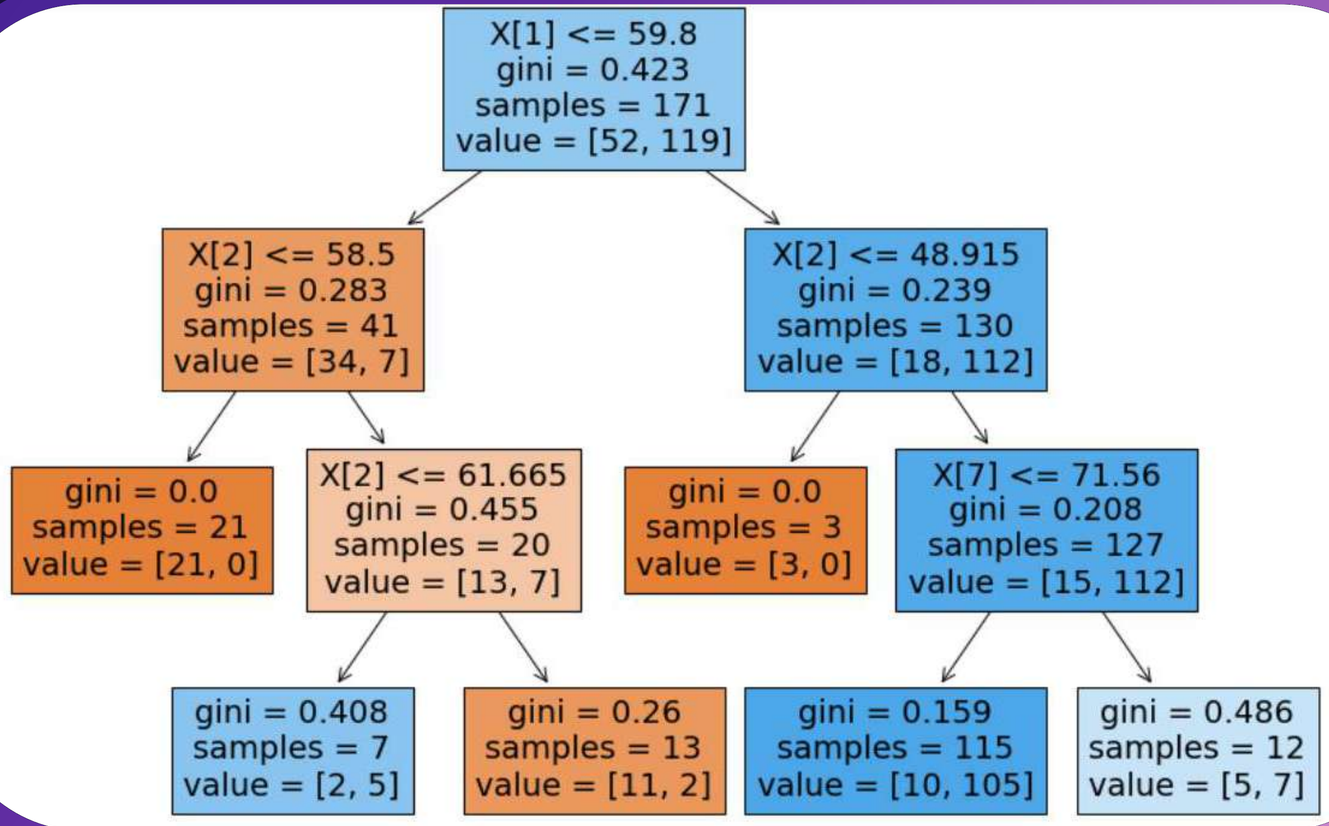
- We have achieved an Accuracy of 72.0930%





Classification Techniques

1 DECISION TREE



2 KNN

3 ic Regression

4 Random Forest

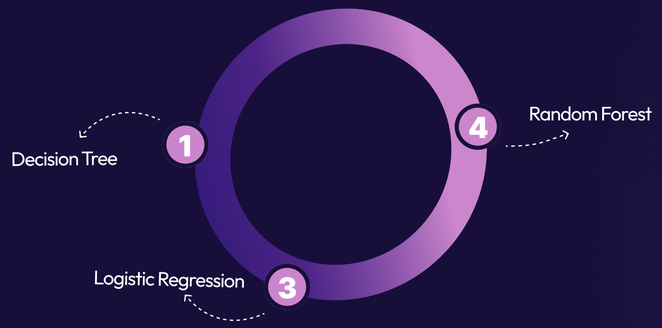
Classification Techniques

2 KNN

- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

- K-NN is a Non-Parametric Algorithm, which means it does not make any assumption on underlying data.

- We have achieved an Accuracy of 81.395%



Classification Techniques

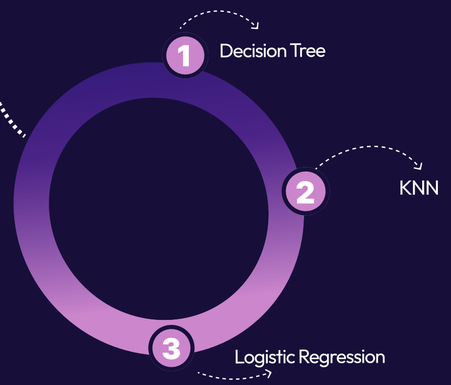
3 LOGISTIC REGRESSION

- This type of statistical model (also known as Logit Model) is often used for classification and predictive analytics. It uses Logistic function to model a binary dependent variable.
- Logistic regression estimates the Probability Of An Event Occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.
- We have achieved an Accuracy of 90.697%

Classification Techniques

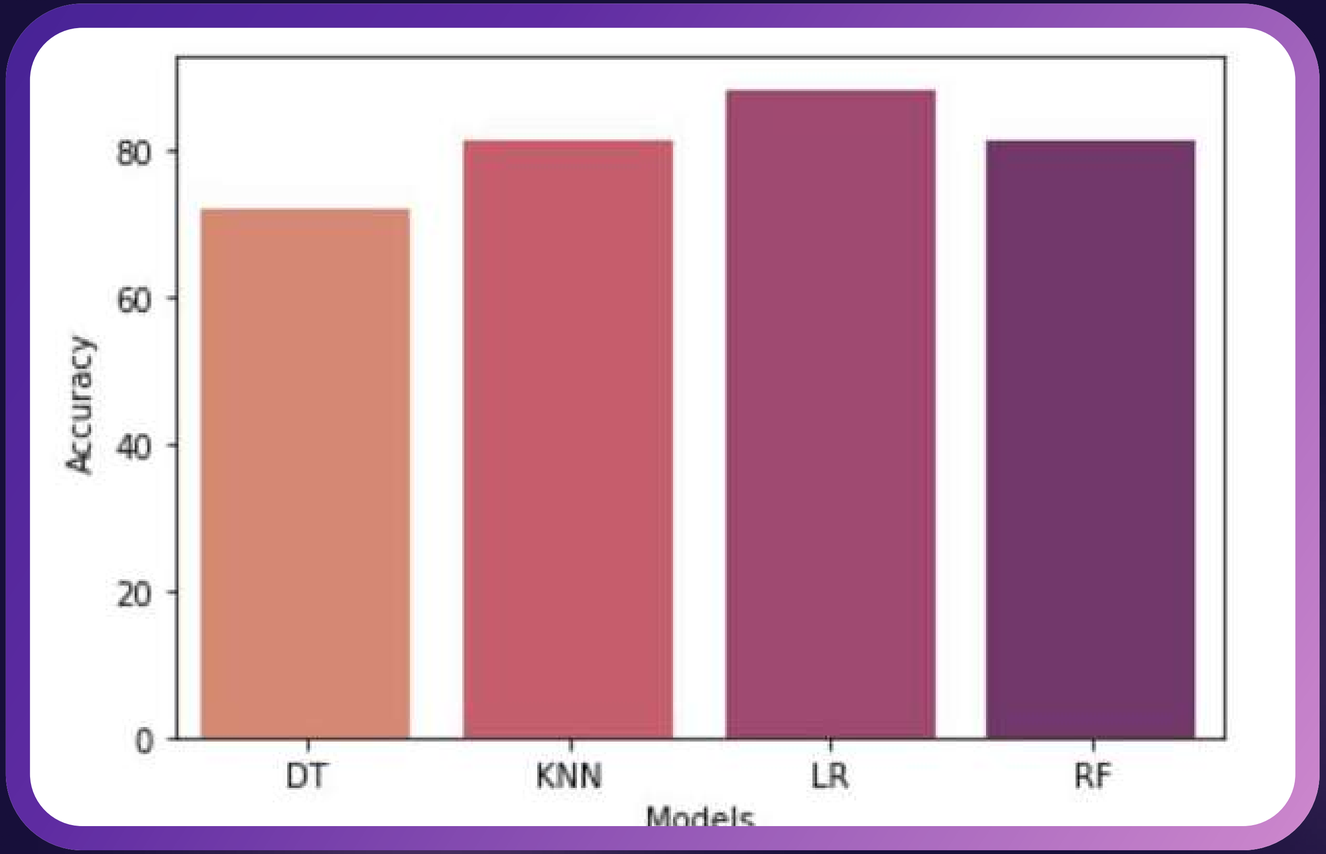
4 RANDOM FOREST

- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and Takes The Average to improve the predictive accuracy of that dataset
- It is based on the Concept Of Ensemble Learning.
- The Greater Number Of Trees in the forest leads to Higher Accuracy and prevents the problem of overfitting.
- We have achieved an Accuracy of 83.725%



Classification Techniques

A Bar Plot Comparing Accuracy All The Techniques Provided



Conclusion

- The problem of campus placement prediction can be solved with the help of different machine learning algorithms such as Logistic regression, Decision Tree, KNN & Random Forest.
- Here, the Logistic Regression algorithm gave the highest accuracy of 90.697% for campus placements prediction.
- The selected features i.e. Gender, SSC percentage, HSC percentage, HSC Specialization, Degree Percentage, UG Degree Stream, Work Experience, E -test Percentage, Degree Specialization & Degree Percentage lead to higher classification accuracy.

THANK YOU

svg images used from: People illustrations by Storyset