# UTS PEMBELAJARAN MESIN

Mujahid Ansori Majid 1197050093 D

## Topik Mini Riset

Studi kasus yang saya bawakan merupakan Campus recrutment dalam mini riset kali ini akan mengetahui faktor apa saja yang menyebabkan students itu dapat ditempatkan

## Data Understanding

Campus recruitment merupakan sebuah strategi untuk menempatkan young talent baik untuk magang atau entry-level position

The dataset includes:

1.sl_no : Serial Number

2.ssc_p: Secondary Education Percentage

3.ssc_b: Board of Secondary Education

4.hsc_p: Higher Secondary Education Percentage

5.hsc_b: Board of Higher Secondary Education

6.hsc_s: Specialisation in Higher Secondary Education

7.degree_p: Degree Percentage

8.degree_t: Field of Degree Education

9.workex: Work Experience

10.etest_p: Employability Test Percentage

11.specialisation: MBA Specialisation

12.mba_p: MBA Percentage

13.status: Status of Placement

## Data Preparation

```
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
```

```
%matplotlib inline
import seaborn as sns
```

```
df=pd.read_csv("/content/drive/MyDrive/dataset/Placement_Data_Full_Class.csv")
```

```
df.head()
```

| | sl_no | gender | ssc_p | ssc_b | hsc_p | hsc_b | hsc_s | degree_p | degree_t | wor |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | M | 67.00 | Others | 91.00 | Others | Commerce | 58.00 | Sci&Tech | |
| **1** | 2 | M | 79.33 | Central | 78.33 | Others | Science | 77.48 | Sci&Tech | |
| **2** | 3 | M | 65.00 | Central | 68.00 | Central | Arts | 64.00 | Comm&Mgmt | |
| **3** | 4 | M | 56.00 | Central | 52.00 | Central | Science | 52.00 | Sci&Tech | |
| **4** | 5 | M | 85.80 | Central | 73.60 | Central | Commerce | 73.30 | Comm&Mgmt | |

```
df.shape
```

```
(215, 15)
```

```
df.info()
# hanya terdapat 67 data yang null pada kolom salary
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sl_no           215 non-null    int64  
 1   gender          215 non-null    object 
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object 
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object 
 6   hsc_s           215 non-null    object 
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object 
 9   workex          215 non-null    object 
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object 
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object 
 14  salary          148 non-null    float64
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

```
# untuk memperjelas
```

```
df.isnull().sum()
```

```
sl_no              0
gender             0
ssc_p              0
ssc_b              0
hsc_p              0
hsc_b              0
hsc_s              0
degree_p           0
degree_t           0
workex             0
etest_p            0
specialisation     0
mba_p              0
status             0
salary            67
dtype: int64
```

kita memiliki 215 kandidat dengan kolom yang beragam. kita kehilangan beberapa value dari salary, kenapa karena angka demikian merupakan student yang tidak ditempatkan atau gagal.

```
df1=df.drop(['sl_no','ssc_b','hsc_b','salary','hsc_s','ssc_p','hsc_p'],axis='columns')
df1
```

| | gender | degree_p | degree_t | workex | etest_p | specialisation | mba_p | status |
|---|---|---|---|---|---|---|---|---|
| **0** | M | 58.00 | Sci&Tech | No | 55.0 | Mkt&HR | 58.80 | Placed |
| **1** | M | 77.48 | Sci&Tech | Yes | 86.5 | Mkt&Fin | 66.28 | Placed |
| **2** | M | 64.00 | Comm&Mgmt | No | 75.0 | Mkt&Fin | 57.80 | Placed |
| **3** | M | 52.00 | Sci&Tech | No | 66.0 | Mkt&HR | 59.43 | Not Placed |
| **4** | M | 73.30 | Comm&Mgmt | No | 96.8 | Mkt&Fin | 55.50 | Placed |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **210** | M | 77.60 | Comm&Mgmt | No | 91.0 | Mkt&Fin | 74.49 | Placed |
| **211** | M | 72.00 | Sci&Tech | No | 74.0 | Mkt&Fin | 53.62 | Placed |
| **212** | M | 73.00 | Comm&Mgmt | Yes | 59.0 | Mkt&Fin | 69.72 | Placed |
| **213** | F | 58.00 | Comm&Mgmt | No | 70.0 | Mkt&HR | 60.23 | Placed |
| **214** | M | 53.00 | Comm&Mgmt | No | 89.0 | Mkt&HR | 60.22 | Not Placed |

```
df1['degree_t'].unique()
```

```
array(['Sci&Tech', 'Comm&Mgmt', 'Others'], dtype=object)
```

```
df1['specialisation'].unique()
```

```
array(['Mkt&HR', 'Mkt&Fin'], dtype=object)
```

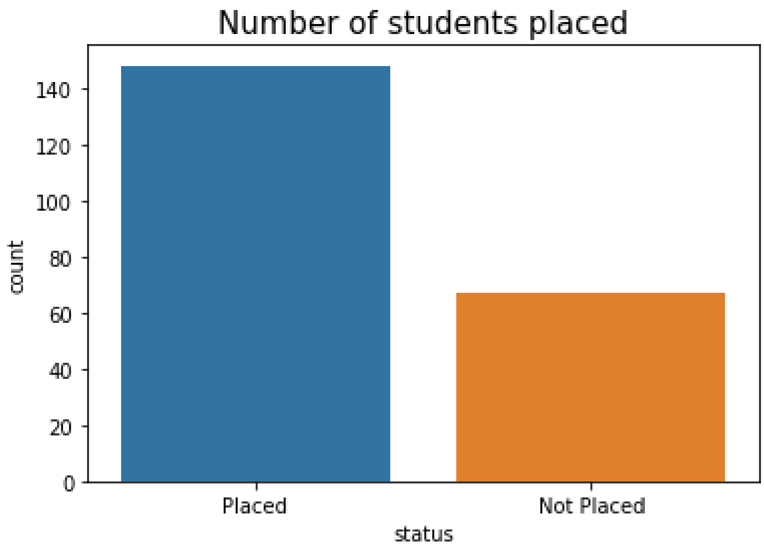
```
df1['status'].value_counts()
```

```
Placed        148
Not Placed     67
Name: status, dtype: int64
```

Terbukti bahwa 148 orang tersalurkan atau dapat ditempatkan, sedangkan sisanya tidak ditempatkan

```
sns.countplot(df1['status'])
plt.title('Number of students placed',fontsize=15)
plt.show()
```

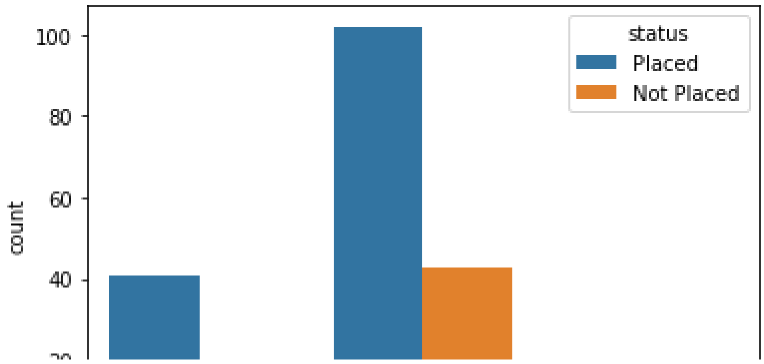
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
```



```
df1.groupby(['degree_t'])['status'].value_counts()
```

```
degree_t   status
Comm&Mgmt  Placed        102
           Not Placed     43
Others     Not Placed      6
           Placed          5
Sci&Tech   Placed         41
           Not Placed     18
Name: status, dtype: int64
```

```
sns.countplot(x=df1['degree_t'],hue=df1['status'])
plt.show()
```

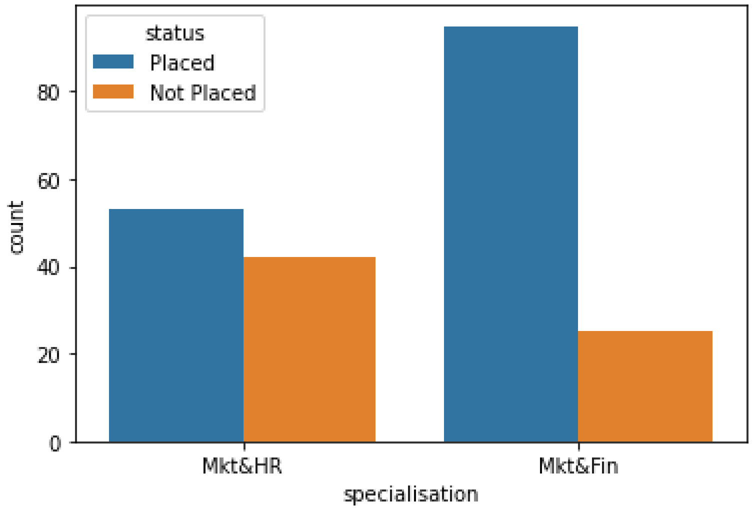


melihat dari jurusannya commerce dan management, pun science dan teknologi. menduduki kursi yang signifikan untuk tersalurkan

```
df1.groupby(['specialisation'])['status'].value_counts()
```

```
specialisation  status    
Mkt&Fin         Placed        95
                Not Placed    25
Mkt&HR          Placed        53
                Not Placed    42
Name: status, dtype: int64
```

```
sns.countplot(x=df1['specialisation'],hue=df1['status'])
plt.show()
```



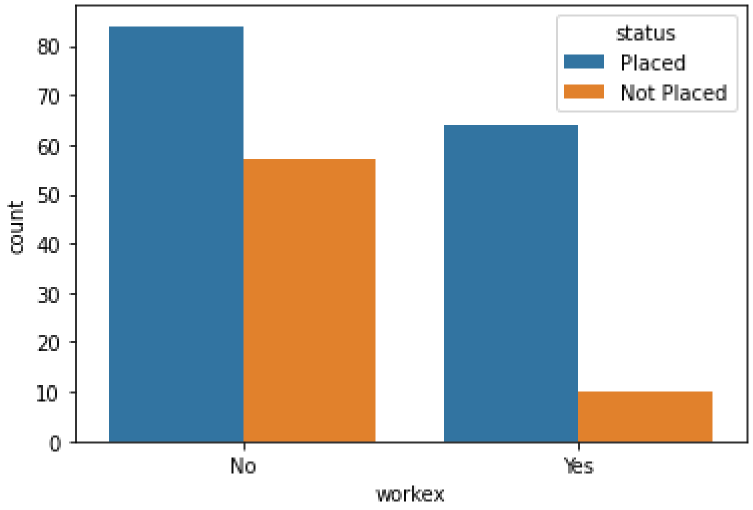
Sekitar 79,16% dan 55,78% siswa masing-masing ditempatkan dari aliran pasar & keuangan dan pasar & SDM. Dari sini kita dapat menyimpulkan bahwa siswa yang memiliki pasar & keuangan sebagai spesialisasi mereka memiliki lebih banyak kesempatan untuk ditempatkan.

```
df1.groupby(['workex'])['status'].value_counts()
```

```
workex  status    
No      Placed        84
        Not Placed    57
Yes     Placed        64
```

```
          Not Placed    10
     Name: status, dtype: int64
```

```
sns.countplot(x=df1['workex'],hue=df1['status'])
plt.show()
```

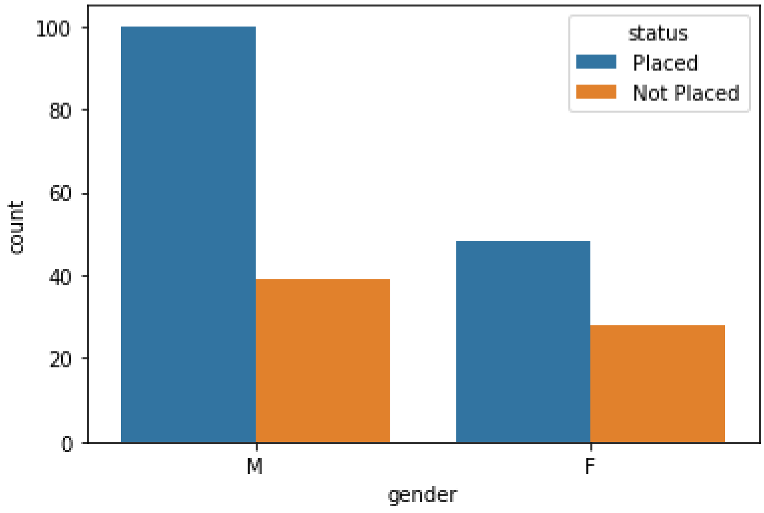


86,48% siswa yang memiliki pengalaman kerja yang diterima bekerja sementara hanya 59,57% siswa yang tidak memiliki pengalaman kerja yang diterima. Oleh karena itu pengalaman kerja mempengaruhi proses perekrutan.

```
df1.groupby(['gender'])['status'].value_counts()
```

```
     gender  status
     F       Placed         48
             Not Placed     28
     M       Placed        100
             Not Placed     39
     Name: status, dtype: int64
```

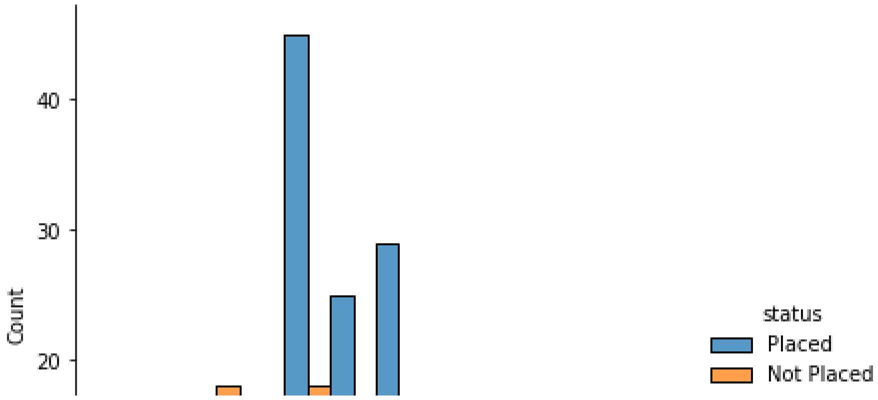
```
sns.countplot(x=df1['gender'],hue=df1['status'])
plt.show()
```



63,16% perempuan dan 71,94% laki-laki mendapat tempat. Laki-laki memiliki lebih banyak kesempatan untuk ditempatkan.

```
sns.displot(x=df1['degree_p'],hue=df1['status'],multiple='dodge')
sns.displot(x=df1['mba_p'],hue=df1['status'],multiple='dodge')
sns.displot(x=df1['etest_p'],hue=df1['status'],multiple='dodge')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8b37667050>
```



Dari representasi di atas dapat dipahami bahw
persentase gelar meningkat, peluang untuk dit
meningkat tetapi ketika mempertimbangkan pers
persentase tes kerja, siswa dengan persentase
ditempatkan

Dari representasi di atas dapat dipahami bahwa ketika persentase gelar meningkat, peluang untuk diterima juga meningkat tetapi ketika mempertimbangkan persentase MBA dan persentase tes kerja, siswa dengan persentase rata-rata ditempatkan

# Building Model

```
le_degree_t=LabelEncoder()
le_specialisation=LabelEncoder()
le_workex=LabelEncoder()
le_status=LabelEncoder()
le_gender=LabelEncoder()
```

```
df1['degree_t_n']=le_degree_t.fit_transform(df1['degree_t'])
df1['specialisation_n']=le_specialisation.fit_transform(df1['specialisation'])
df1['workex_n']=le_workex.fit_transform(df1['workex'])
df1['status_n']=le_status.fit_transform(df1['status'])
df1['gender_n']=le_gender.fit_transform(df1['gender'])
```

```
df1.head()
```

| | gender | degree_p | degree_t | workex | etest_p | specialisation | mba_p | status | deg |
|---|---|---|---|---|---|---|---|---|---|
| **0** | M | 58.00 | Sci&Tech | No | 55.0 | Mkt&HR | 58.80 | Placed | |
| **1** | M | 77.48 | Sci&Tech | Yes | 86.5 | Mkt&Fin | 66.28 | Placed | |

```
df2=df1.drop(['degree_t','specialisation','workex','status','gender'],axis='columns')
df2.head()
```

| | degree_p | etest_p | mba_p | degree_t_n | specialisation_n | workex_n | status_n | gende |
|---|---|---|---|---|---|---|---|---|
| **0** | 58.00 | 55.0 | 58.80 | 2 | 1 | 0 | 1 | |
| **1** | 77.48 | 86.5 | 66.28 | 2 | 0 | 1 | 1 | |
| **2** | 64.00 | 75.0 | 57.80 | 0 | 0 | 0 | 1 | |
| **3** | 52.00 | 66.0 | 59.43 | 2 | 1 | 0 | 0 | |
| **4** | 73.30 | 96.8 | 55.50 | 0 | 0 | 0 | 1 | |

```
df2.astype(int)
```

| | degree_p | etest_p | mba_p | degree_t_n | specialisation_n | workex_n | status_n | gen |
|---|---|---|---|---|---|---|---|---|
| **0** | 58 | 55 | 58 | 2 | 1 | 0 | 1 | |
| **1** | 77 | 86 | 66 | 2 | 0 | 1 | 1 | |
| **2** | 64 | 75 | 57 | 0 | 0 | 0 | 1 | |
| **3** | 52 | 66 | 59 | 2 | 1 | 0 | 0 | |
| **4** | 73 | 96 | 55 | 0 | 0 | 0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **210** | 77 | 91 | 74 | 0 | 0 | 0 | 1 | |
| **211** | 72 | 74 | 53 | 2 | 0 | 0 | 1 | |
| **212** | 73 | 59 | 69 | 0 | 0 | 1 | 1 | |
| **213** | 58 | 70 | 60 | 0 | 1 | 0 | 1 | |
| **214** | 53 | 89 | 60 | 0 | 1 | 0 | 0 | |

215 rows × 8 columns

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   degree_p          215 non-null    float64
 1   etest_p           215 non-null    float64
 2   mba_p             215 non-null    float64
 3   degree_t_n        215 non-null    int64  
```

```
  4   specialisation_n  215 non-null    int64
  5   workex_n          215 non-null    int64
  6   status_n          215 non-null    int64
  7   gender_n          215 non-null    int64
 dtypes: float64(3), int64(5)
 memory usage: 13.6 KB
```

```
X=df2.drop(['status_n'],axis='columns')
y=df2['status_n']
```

# Logistic Regression

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

```
from sklearn.linear_model import LogisticRegression
clf=LogisticRegression()
clf.fit(X_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: Converg
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
LogisticRegression()
```

```
X_test
```

```
     degree_p  etest_p  mba_p  degree_t_n  specialisation_n  workex_n
 96     76.00    66.00  64.44           0                 0         1
197     91.00    59.32  69.71           2                 1         0
 37     65.60    58.00  55.47           2                 1         0
 64     75.00    61.00  58.78           0                 0         0
101     68.00    78.00  60.44           0                 1         0
  4     73.30    96.80  55.50           0                 0         0
107     83.00    80.00  73.52           0                 1         0
192     64.80    93.40  57.34           0                 0         1
214     53.00    89.00  60.22           0                 1         0
 82     74.00    82.00  60.44           0                 0         0
109     65.00    86.00  56.09           2                 1         1
136     64.00    78.00  61.58           0                 0         0
 51     56.20    67.00  62.65           0                 1         0
211     72.00    74.00  53.62           2                 0         0
142     73.43    60.00  61.29           2                 0         1
182     57.00    75.00  59.81           1                 0         1
145     71.25    72.00  63.23           2                 1         0
131     62.00    80.00  60.78           1                 0         1
122     69.30    80.40  71.00           0                 0         1
121     69.60    55.67  71.49           2                 1         1
173     55.00    67.00  59.32           2                 1         0
133     77.00    65.00  60.98           0                 1         1
206     60.00    97.00  53.39           0                 0         0
111     61.00    60.00  60.64           2                 1         0
198     65.00    88.00  71.96           1                 1         0
 39     64.00    93.00  62.56           2                 0         0
 30     73.00    52.00  56.70           0                 1         0
167     67.00    58.10  75.71           2                 0         1
 22     72.23    55.53  68.81           2                 1         0
 23     64.74    92.00  63.62           2                 0         1
166     60.00    63.00  52.38           0                 1         1
 71     71.00    95.00  66.94           0                 0         0
```

```
clf.predict(X_test)
```

```
array([1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

```
y_test
```

```
96     1
197    1
37     1
64     1
101    1
4      1
107    1
192    1
214    0
82     0
109    0
136    0
51     0
211    1
142    1
182    0
145    1
131    1
122    1
121    1
173    0
133    1
206    0
111    0
198    0
39     1
30     1
167    0
22     1
23     1
166    1
71     1
137    1
103    1
178    1
88     1
44     1
11     1
17     0
85     1
185    1
204    1
110    1
Name: status_n, dtype: int64
```

```
clf.score(X_test,y_test)
```

```
0.813953488372093
```

# Decision Tree Algorithm

```
from sklearn import tree
model_2 = tree.DecisionTreeClassifier(criterion='entropy')
model_2.fit(X_train, y_train)
```

```
     DecisionTreeClassifier(criterion='entropy')
```

```
model_2.score(X_train,y_train)
```

```
     1.0
```

# Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
model_3 = RandomForestClassifier(n_estimators=100)
model_3.fit(X_train, y_train)
```

```
     RandomForestClassifier()
```

```
model_3.score(X_test, y_test)
```

```
     0.7209302325581395
```

✓ 0s completed at 10:18 PM ● ✕