Nama : Azzahra Fahriza

NIM : 2102296

Saya memilih dataset kepadatan penduduk karena sepertinya lebih mudah di pahami oleh saya

rencana:

pertama, saya akan mengguabungkan dulu dataset pmi dan kepadatan penduduk

kedua, saya akan mengubah type data provinsi menjadi category, dan tipe data kepadatan penduduk menjadi int

ketiga, saya akan mengubah yang yang kepadatan penduduk menjadi 4 kelas (< 150 rendah, 200-300 sedang, 300-500 tinggi, >500 sangat tinggi)

keempat, membuat EDA

kemudian di kisarbasi mengambil 70% data untuk datatraining, dan 10% untuk data validasi

hasilnya ternyata lebih sesuai dengan rencana

In [1]:
```
# Pertama-tama import data yang ada di excel
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.tree
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
import xgboost as xgb
from sklearn.metrics import confusion_matrix
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.metrics import mean_score
from sklearn.decomposition import PCA
import numpy
import pickle
# Free sklearn-external import joblib
```

```
C:\Users\Azzahra Fahriza\anaconda3\lib\site-packages\xgboost\compat.py:36: FutureWarning: pandas.Int64Index is deprecated and will be removed from pandas in a future version. Use p
d.Index(arr, dtype=np.int64)
  from pandas import MultiIndex, Int64Index
```

In [2]:
```
df = pd.read_csv("C:/Users/Azzahra Fahriza/Documents/SEMESTER 3/Data Mining/uas/dataset/kepadatan_penduduk.csv")
df1 = pd.read_csv("C:/Users/Azzahra Fahriza/Documents/SEMESTER 3/Data Mining/uas/dataset/pmi.csv")
```

In [3]:
```
# menggabungkan 2 file csv menjadi 1 pada df2
df2=pd.merge(df1,df, how='left')
```

In [4]:
```
df2.describe()
```

Out[4]:

	tahun
count	442.000000
mean	2016.000000
std	3.162837
min	2010.000000
25%	2013.000000
50%	2016.000000
75%	2019.000000
max	2022.000000

In [5]:
```
df2.head(10)
```

Out[5]:

	prov	tahun	kategori_pmi	kepadatan_penduduk
0	ACEH	2022	TINGGI	NaN
1	SUMATERA UTARA	2022	TINGGI	NaN
2	SUMATERA BARAT	2022	TINGGI	NaN
3	RIAU	2022	TINGGI	NaN
4	JAMBI	2022	TINGGI	NaN
5	SUMATERA SELATAN	2022	TINGGI	NaN
6	BENGKULU	2022	TINGGI	NaN
7	LAMPUNG	2022	TINGGI	NaN
8	KEP. BANGKA BELITUNG	2022	TINGGI	NaN
9	KEP. RIAU	2022	TINGGI	NaN

In [6]:
```
df.isnull().sum()
print("****************************************")
df1.isnull().sum()
print("****************************************")
df2.isnull().sum()
```

```
****************************************
prov            0
tahun           0
kategori_pmi    0
kepadatan_penduduk    170
dtype: int64
```

In [7]:
```
df1["prov"]=df1["prov"].astype('category')
df1["tahun"]=df1["tahun"].astype('int')
df1["kategori_pmi"]=df1["kategori_pmi"].astype('category')
# df1["kepadatan_penduduk"] = df1["kepadatan_penduduk"].astype('int')
# df1["kepadatan_penduduk"].astype('category')
# df1["kepadatan_penduduk"] = df1["kepadatan_penduduk"].astype(object).astype(float)
```

In [8]:
```
df2["prov"]=df2["prov"].astype('category')
df2["kategori_pmi"]=df2["kategori_pmi"].astype('category')
df2["kepadatan_penduduk"]=df2["kepadatan_penduduk"].astype('category')
```

In [9]:
```
df.info()
print("****************************************")
df1.info()
print("****************************************")
df2.info()
```

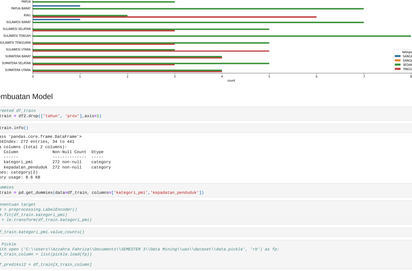
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 270
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   prov              200 non-null    category
 1   tahun             200 non-null    int64
 2   kepadatan_penduduk  200 non-null    object
dtypes: category(1), int64(1), object(1)
memory usage: 6.1+ KB
****************************************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 442 entries, 0 to 441
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   prov              442 non-null    category
 1   tahun             442 non-null    int64
 2   kategori_pmi      442 non-null    object
dtypes: category(1), int64(1), object(1)
memory usage: 8.6+ KB
****************************************
<class 'pandas.core.frame.DataFrame'>
Int64Index: 442 entries, 0 to 441
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   prov              442 non-null    category
 1   tahun             442 non-null    int64
 2   kategori_pmi      442 non-null    category
 3   kepadatan_penduduk  272 non-null    category
dtypes: category(3), int64(1)
memory usage: 15.6 KB
```

In [10]:
```
df2 = df2.dropna()
```

In [11]:
```
# mengubah angka dari kepadatan penduduk menjadi kategori (sedang, padat, sangat padat)????
```

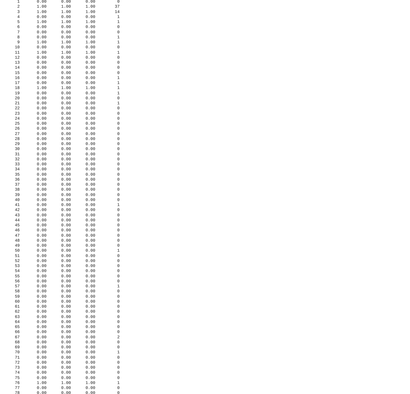
In [12]:
```
fig, ax = plt.subplots(figsize=(20, 5))# ketur ukuran chart
sns.countplot(axis, x="tahun", hue="kategori_pmi", data=df2)
```

Out[12]:
```
<AxesSubplot:xlabel='tahun', ylabel='count'>
```



In [13]:
```
fig, ax = plt.subplots(figsize=(20, 15))# ketur ukuran chart
sns.countplot(axis, y="prov",hue="kategori_pmi", data=df2)
```

Out[13]:
```
<AxesSubplot:xlabel='count', ylabel='prov'>
```



Pembuatan Model

In [14]:
```
# Created df_train
df_train = df2.drop(['tahun', 'prov'],axis=1)
```

In [15]:
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 272 entries, 30 to 441
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   kategori_pmi      272 non-null    category
 1   kepadatan_penduduk  272 non-null    category
dtypes: category(2)
memory usage: 8.6 KB
```

In [16]:
```
df_train = pd.get_dummies(data=df_train, columns=['kategori_pmi', 'kepadatan_penduduk'])
```

In [17]:
```
# penentuan target
# ip = df_train['kepadatan_labelEncoder']
# ip_fitd=df_train.kategori_pmi
# x = ip.reshape(-1,1).astype(int)
```

In [18]:
```
# df_train.kategori_pmi.value_counts()
```

In [19]:
```
# x Pickle
# pickle_out = open("C:\Users\Azzahra Fahriza\Documents\SEMESTER 3\Data Mining\uas\dataset\data.pickle", 'rb') as fp:
# x_train_column = list(pickle.load(fp))

# df_predikti2 = df_train[x_train_column]
```

In [20]:
```
# Split dataset
X_train,X_test,y_train,y_test=train_test_split(df_train,df_train_test_size=0.2,random_state=123)
```

In [21]:
```
# m_scaler = preprocessing.MinMaxScaler()
# X_train_minmax = mm_scaler.fit_transform(X_train)
# X_test_minmax = mm_scaler.transform(X_test)
```

In [22]:
```
# Modeling
clf_fit = RandomForestClassifier(n_estimators=50, random_state=123)
clf_fit.fit(X_train, Y_train)
Y_pred = clf_fit.predict(X_test)
acc = accuracy_score(Y_test, Y_pred)
print("Akurasi {}".format(acc))
print(classification_report(Y_test, Y_pred))
```

```
Akurasi 0.3454545454545455

              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           1       0.00      0.00      0.00         1
           2       1.00      1.00      1.00        17
           3       1.00      1.00      1.00        14
           4       0.00      0.00      0.00         1
           5       1.00      1.00      1.00         1
           6       0.00      0.00      0.00         0
           7       0.00      0.00      0.00         0
           8       0.00      0.00      0.00         1
           9       1.00      1.00      1.00         1
          10       0.00      0.00      0.00         0
          11       1.00      1.00      1.00         1
          12       0.00      0.00      0.00         0
          13       0.00      0.00      0.00         0
          14       0.00      0.00      0.00         0
          15       0.00      0.00      0.00         0
          16       0.00      0.00      0.00         0
          17       0.00      0.00      0.00         1
          18       1.00      1.00      1.00         1
          19       0.00      0.00      0.00         1
          20       0.00      0.00      0.00         0
          21       0.00      0.00      0.00         1
          22       0.00      0.00      0.00         0
          23       0.00      0.00      0.00         0
          24       0.00      0.00      0.00         0
          25       0.00      0.00      0.00         0
          26       0.00      0.00      0.00         0
          27       0.00      0.00      0.00         0
          28       0.00      0.00      0.00         0
          29       0.00      0.00      0.00         0
          30       0.00      0.00      0.00         0
          31       0.00      0.00      0.00         0
          32       0.00      0.00      0.00         0
          33       0.00      0.00      0.00         0
          34       0.00      0.00      0.00         0
          35       0.00      0.00      0.00         0
          36       0.00      0.00      0.00         0
          37       0.00      0.00      0.00         0
          38       0.00      0.00      0.00         0
          39       0.00      0.00      0.00         0
          40       0.00      0.00      0.00         0
          41       0.00      0.00      0.00         0
          42       0.00      0.00      0.00         0
          43       0.00      0.00      0.00         0
          44       0.00      0.00      0.00         0
          45       0.00      0.00      0.00         0
          46       0.00      0.00      0.00         0
          47       0.00      0.00      0.00         0
          48       0.00      0.00      0.00         0
          49       0.00      0.00      0.00         0
          50       0.00      0.00      0.00         1
          51       0.00      0.00      0.00         0
          52       0.00      0.00      0.00         0
          53       0.00      0.00      0.00         0
          54       0.00      0.00      0.00         0
          55       0.00      0.00      0.00         0
          56       0.00      0.00      0.00         0
          57       0.00      0.00      0.00         1
          58       0.00      0.00      0.00         0
          59       0.00      0.00      0.00         0
          60       0.00      0.00      0.00         0
          61       0.00      0.00      0.00         0
          62       0.00      0.00      0.00         0
          63       0.00      0.00      0.00         0
          64       0.00      0.00      0.00         0
          65       0.00      0.00      0.00         0
          66       0.00      0.00      0.00         0
          67       0.00      0.00      0.00         0
          68       0.00      0.00      0.00         0
          69       0.00      0.00      0.00         0
          70       0.00      0.00      0.00         0
          71       0.00      0.00      0.00         0
          72       0.00      0.00      0.00         0
          73       0.00      0.00      0.00         0
          74       0.00      0.00      0.00         0
          75       0.00      0.00      0.00         0
          76       1.00      1.00      1.00         1
          77       0.00      0.00      0.00         0
          78       0.00      0.00      0.00         0
          79       0.00      0.00      0.00         0
          80       0.00      0.00      0.00         1
          81       0.00      0.00      0.00         0
          82       0.00      0.00      0.00         0
          83       0.00      0.00      0.00         0
          84       0.00      0.00      0.00         0
          85       0.00      0.00      0.00         1
          86       0.00      0.00      0.00         0
          87       0.00      0.00      0.00         0
          88       0.00      0.00      0.00         0
          89       0.00      0.00      0.00         0
          90       0.00      0.00      0.00         0
          91       0.00      0.00      0.00         0
          92       0.00      0.00      0.00         0
          93       0.00      0.00      0.00         1
          94       0.00      0.00      0.00         0
          95       0.00      0.00      0.00         0
          96       0.00      0.00      0.00         0
          97       0.00      0.00      0.00         1
          98       0.00      0.00      0.00         0
          99       0.00      0.00      0.00         0
         100       0.00      0.00      0.00         0
         101       0.00      0.00      0.00         0
         102       0.00      0.00      0.00         0
         103       0.00      0.00      0.00         0
         104       0.00      0.00      0.00         1
         105       0.00      0.00      0.00         0
         106       0.00      0.00      0.00         0
         107       0.00      0.00      0.00         0
         108       0.00      0.00      0.00         0
         109       0.00      0.00      0.00         0
         110       0.00      0.00      0.00         0
         111       0.00      0.00      0.00         0
         112       0.00      0.00      0.00         0
         113       0.00      0.00      0.00         1
         114       0.00      0.00      0.00         0
         115       0.00      0.00      0.00         0
         116       0.00      0.00      0.00         0
         117       0.00      0.00      0.00         0
         118       0.00      0.00      0.00         1
         119       0.00      0.00      0.00         0
         120       0.00      0.00      0.00         0
         121       1.00      1.00      1.00         1
         122       0.00      0.00      0.00         0
         123       1.00      1.00      1.00         2
         124       0.00      0.00      0.00         0
         125       0.00      0.00      0.00         0
         126       1.00      1.00      1.00         1
         127       0.00      0.00      0.00         1
         128       0.00      0.00      0.00         0
         129       0.00      0.00      0.00         0
         130       0.00      0.00      0.00         0
         131       0.00      0.00      0.00         2
         132       0.00      0.00      0.00         0
         133       0.00      0.00      0.00         0
         134       0.00      0.00      0.00         0
         135       0.00      0.00      0.00         1
         136       0.00      0.00      0.00         0
         137       1.00      1.00      1.00         1
         138       1.00      1.00      1.00         1
         139       0.00      0.00      0.00         0
         140       0.00      0.00      0.00         0
         141       0.00      0.00      0.00         0
         142       0.00      0.00      0.00         0
         143       0.00      0.00      0.00         0
         144       0.00      0.00      0.00         0
         145       1.00      1.00      1.00         1
         146       0.00      0.00      0.00         0
         147       0.00      0.00      0.00         0
         148       1.00      1.00      1.00         1
         149       0.00      0.00      0.00         0
         150       0.00      0.00      0.00         0
         151       0.00      0.00      0.00         1
         152       0.00      0.00      0.00         0
         153       1.00      1.00      1.00         1
         154       1.00      1.00      1.00         1
         155       0.00      0.00      0.00         0
         156       0.00      0.00      0.00         0
         157       1.00      1.00      1.00         1
         158       0.00      0.00      0.00         0
         159       0.00      0.00      0.00         1
         160       0.00      0.00      0.00         1
         161       0.00      0.00      0.00         0
         162       0.00      0.00      0.00         1
         163       0.00      0.00      0.00         0
         164       0.00      0.00      0.00         0
         165       0.00      0.00      0.00         0
         166       0.00      0.00      0.00         0
         167       0.00      0.00      0.00         1
         168       0.00      0.00      0.00         0
         169       0.00      0.00      0.00         0
         170       0.00      0.00      0.00         0
         171       0.00      0.00      0.00         0
         172       0.00      0.00      0.00         0
         173       0.00      0.00      0.00         1
         174       0.00      0.00      0.00         0
         175       0.00      0.00      0.00         2
         176       0.00      0.00      0.00         0
         177       0.00      0.00      0.00         0
         178       0.00      0.00      0.00         0
         179       0.00      0.00      0.00         0
         180       0.00      0.00      0.00         0
         181       1.00      1.00      1.00         1
         182       0.00      0.00      0.00         0
         183       0.00      0.00      0.00         0
         184       0.00      0.00      0.00         0
         185       0.00      0.00      0.00         1
         186       0.00      0.00      0.00         0

    micro avg       0.34      0.34      0.34       110
    macro avg       0.11      0.11      0.11       110
 weighted avg       0.67      0.67      0.67       110
  samples avg       0.34      0.34      0.34       110

C:\Users\Azzahra Fahriza\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in
labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Azzahra Fahriza\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in lab
els with no true samples. Use `zero_division` parameter to control this behavior
  _warn_prf(average, modifier, msg_start, len(result))
```

In [23]:
```
```