Implementation Regression and Naïve Bayes to Predict and Classify Data Asset at Educational Institutions

Handy Noviyarto
Faculty of Computer Science, Mercu Buana University, Indonesia

Abstract— *Regional assets represent regional assets which in essence belong to the respective provincial government. These government assets can play a role as collateral for regional development. The preparation of asset documents aims to safeguard assets from the aspect of regional administration. In this study, to process prediction analysis use the regression method, and to process classification use the Naive Bayes method. The purpose of this study was to predict and classify data asset that will be used and categorized according to regional planning using Regression and Naïve Bayes Method. This research was conducted using the Python programming language and the Visual Studio code.*

Keywords— *Regression, naïve bayes, clustering, data mining.*

I. INTRODUCTION

Local assets are essentially regional wealth is owned by the provincial government each - each. One is a regional asset is an asset not move. As for which is included in the fixed assets to which such land or land, buildings, and so forth. In this aspect, it can play a role of government assets as collateral development in the region. Preparation of the document aims to secure the assets of the assets of the administrative aspects of the area.

According to the Government Accounting Standards (2016) assets are economic resources controlled or owned by the government as a result of past events and from which economic and social benefits in the future is expected can be obtained either by the government or the public, as well as can be measured, including non-financial resources needed to provide services to the general public and resources maintained for historical and cultural reasons.

Asset security aims to keep local assets do not change hands illegally and facilitate local authorities in managing further. Safeguarding assets is absolutely necessary by completing the assets referred to as legal documents. In addition, a regional asset wealth can act as a guarantee of regional development.

A common problem of the government's assets, which is not yet completed the document, even none at all. Not infrequently, the region's assets lost due to various reasons. Such as the lack of accuracy of the value of the assets being managed, the unclear status of the assets being managed, and others.

Based on the background of the issue, so in this study was taken the title "Implementation Regression and Naïve Bayes To Predict And Classify Data Asset".

II. PLATFORM THEORY

2.1 Definition of Data Mining

Data mining is the process to obtain useful information from large data base warehouse. Techniques in Data Mining: how to search for the data that is to build a model. The model was used to identify the pattern of other data that are not in the data base stored.

2.2. Regression Analysis

Regression analysis in statistics is one method for determining the causal relationship between one variable and another variable (s). "Cause" variables are referred to by various terms: explanatory variables, explanatory variables, independent variables, or independently, variable X (because it is often depicted on the graph as abscissa, or the X-axis). Variables affected as a result are known as influenced variables, dependent variables, dependent variables, or Y variables. Both of these variables can be random variables (random), but the variables affected must always be random variables.

Regression analysis is one of the most popular and widely used analyzes. Regression analysis is widely used to make predictions and forecasts, with uses that complement each other in the field of machine learning. This analysis is also used to understand which independent variables are related to the dependent variable, and to find out the forms of the relationship.

2.3 Naïve Bayes

According Thomas Bayes, The Naive Bayes algorithm is a classification method using probability and statistical methods. The Naive Bayes algorithm predicts future opportunities based on past experience so it is known as the Bayes Theorem. The main characteristic of Naïve Bayes Classifier is a very strong assumption of independence from each condition / event.

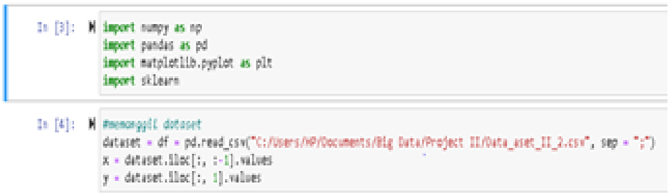
The advantage of using this method is only requires a small amount of training data to determine the estimated parameters needed in the classification process. Because it is assumed to be an independent variable, only the variance of a variable in a class is needed to determine the classification, not the whole of the covariance matrix.

III. RESULTS AND DISCUSSION

3.1 Regression Analysis Method

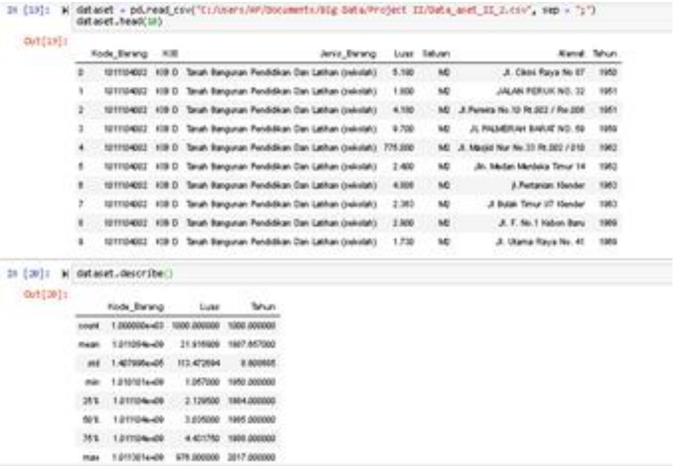
a. Data Preprocessing

Before using the code for data processing, first input library that will be used:



```
In [3]:  import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          import sklearn

In [4]:  #memanggil dataset
          dataset = df = pd.read_csv("C:/Users/HP/Documents/Big Data/Project II/Data_aset_II_2.csv", sep = ";")
          x = dataset.iloc[:, :-1].values
          y = dataset.iloc[:, 1].values
```

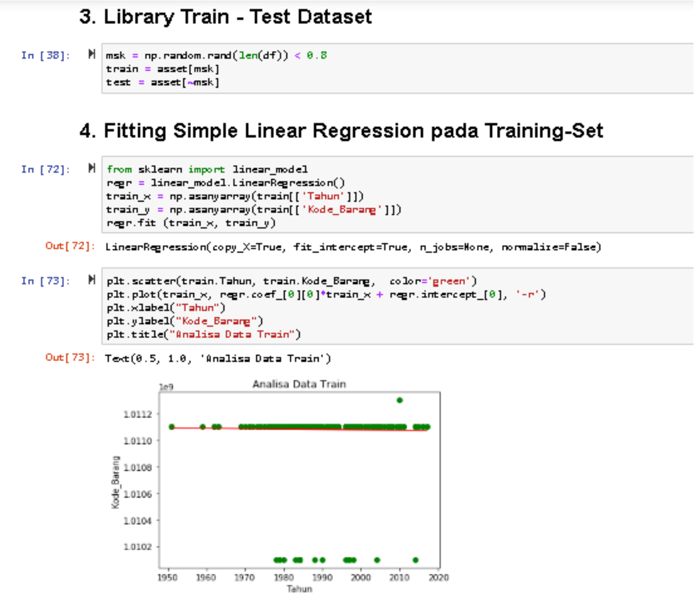
We have made, then uflood some parts, such as the name of the dataset is loaded:



```
In [19]:  dataset = pd.read_csv("C:/Users/HP/Documents/Big Data/Project II/Data_aset_II_2.csv", sep = ";")
           dataset.head(10)
Out[19]:
   Kode_Barang  KIB    Jenis_Barang                                      Luas     Satuan  Alamat                              Tahun
0  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   5.190    M2      Jl. Cikini Raya No 87               1950
1  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   1.800    M2      JALAN PERUK NO. 32                  1951
2  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   4.130    M2      Jl.Pemuda No.10 Rt.002 / Rw.008     1951
3  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   9.700    M2      JL PALMERAH BARAT NO. 59            1959
4  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   775.000  M2      Jl. Masjid Nur No.33 Rt.002 / 010   1962
5  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   2.400    M2      Jln. Medan Merdeka Timur 14         1962
6  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   4.800    M2      Jl.Pertanian Klender                1963
7  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   2.383    M2      Jl Bulak Timur I/7 Klender          1963
8  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   2.800    M2      Jl. F. No.1 Kebon Baru              1969
9  1011104002   KIB D  Tanah Bangunan Pendidikan Dan Latihan (sekolah)   1.730    M2      Jl. Utama Raya No. 41               1969

In [20]:  dataset.describe()
Out[20]:
        Kode_Barang   Luas         Tahun
count   1.000000e+03  1000.000000  1000.000000
mean    1.011094e+09  21.916909    1987.657000
std     1.407995e+05  113.472694   8.600665
min     1.010101e+09  1.057000     1950.000000
25%     1.011104e+09  2.129500     1984.000000
50%     1.011104e+09  3.035000     1985.000000
75%     1.011104e+09  4.401750     1988.000000
max     1.011301e+09  976.000000   2017.000000
```

Figure 1. Data Pre Processing Regression Analysis

b. Fitting Simple Linear Regression in Training Set



3. Library Train - Test Dataset

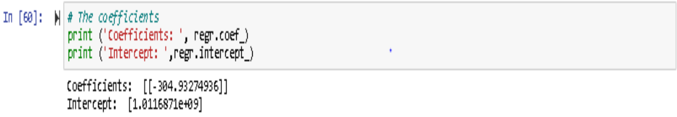
```
In [38]:  msk = np.random.rand(len(df)) < 0.8
           train = asset[msk]
           test = asset[~msk]
```

4. Fitting Simple Linear Regression pada Training-Set

```
In [72]:  from sklearn import linear_model
           regr = linear_model.LinearRegression()
           train_x = np.asanyarray(train[['Tahun']])
           train_y = np.asanyarray(train[['Kode_Barang']])
           regr.fit (train_x, train_y)

Out[72]:  LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

In [73]:  plt.scatter(train.Tahun, train.Kode_Barang,  color='green')
           plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-r')
           plt.xlabel("Tahun")
           plt.ylabel("Kode_Barang")
           plt.title("Analisa Data Train")

Out[73]:  Text(0.5, 1.0, 'Analisa Data Train')
```

Figure 2. Fitting Simple Linear Regression in Training Set

Creating value coefficient and the intercept on predictive data tabulation.



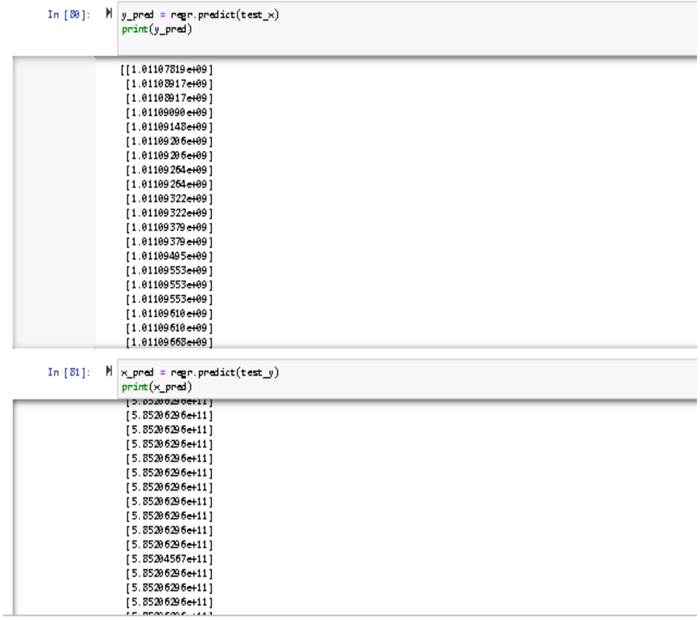
```
In [60]:  # The coefficients
           print ('Coefficients: ', regr.coef_)
           print ('Intercept: ',regr.intercept_)

           Coefficients:  [[-304.93274936]]
           Intercept:  [1.0116871e+09]
```

Figure 3. Coefficient and Intercept

Coefficients and interception in a simple linear regression fit the parameters of the line. Given that this is a simple linear regression, with only two parameters, and knowing that the parameter is the intercept and the slope of the line, can sklearn direct estimate of the data.

c. Predicting Results of Test-Set

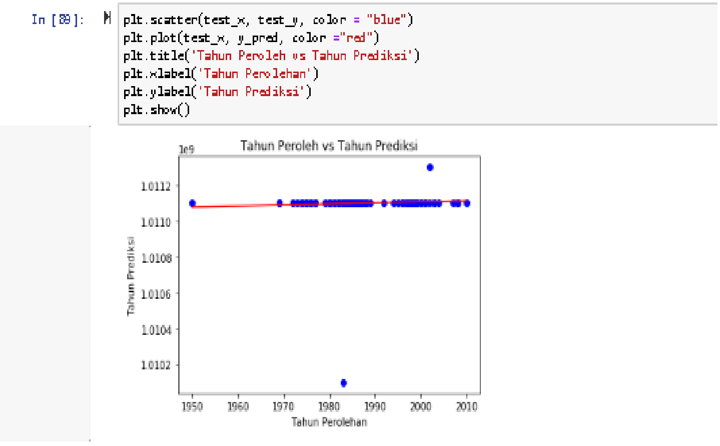
And to know the quality of the data is carried out also for testing against test data



```
In [80]:  y_pred = regr.predict(test_x)
           print(y_pred)

[[1.01107819e+09]
 [1.01108917e+09]
 [1.01108917e+09]
 [1.01109090e+09]
 [1.01109148e+09]
 [1.01109206e+09]
 [1.01109206e+09]
 [1.01109264e+09]
 [1.01109264e+09]
 [1.01109322e+09]
 [1.01109322e+09]
 [1.01109379e+09]
 [1.01109379e+09]
 [1.01109495e+09]
 [1.01109553e+09]
 [1.01109553e+09]
 [1.01109553e+09]
 [1.01109610e+09]
 [1.01109610e+09]
 [1.01109668e+09]

In [81]:  x_pred = regr.predict(test_y)
           print(x_pred)

 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85204567e+11]
 [5.85206296e+11]
 [5.85206296e+11]
 [5.85206296e+11]
```

Figure 4. Predicting Result

And included the form of the plot



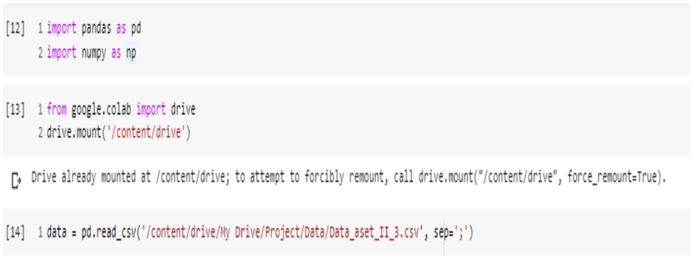
```
In [88]:  plt.scatter(test_x, test_y, color = "blue")
           plt.plot(test_x, y_pred, color ="red")
           plt.title('Tahun Peroleh vs Tahun Prediksi')
           plt.xlabel('Tahun Perolehan')
           plt.ylabel('Tahun Prediksi')
           plt.show()
```

Figure 5. Regression Analysis

3.2 Naïve Bayes Method

a. Selection Data

Before using the code for data processing, first input library that will be used:

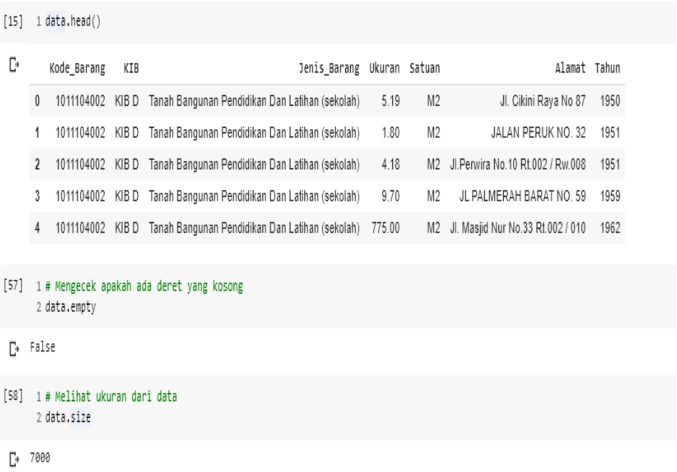


```
[12] import pandas as pd
     import numpy as np

[13] from google.colab import drive
     drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[14] data = pd.read_csv('/content/drive/My Drive/Project/Data/Data_aset_II_3.csv', sep=';')
```

Figure 6. Selection Data

b. Preprocessing/Cleaning Data

Before the data is used, cleansing data will be taken:



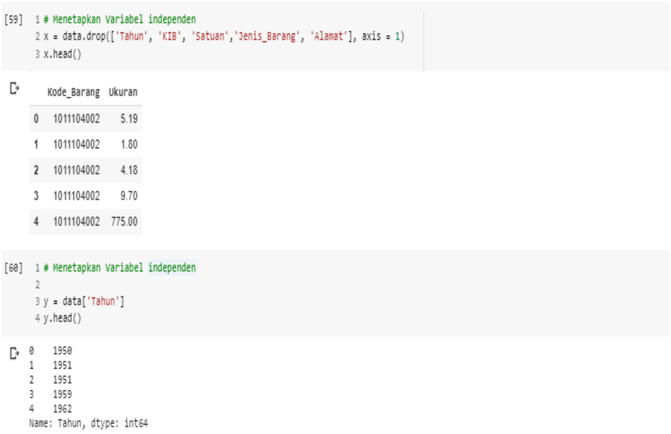
```
[15] data.head()
```

	Kode_Barang	KIB	Jenis_Barang	Ukuran	Satuan	Alamat	Tahun
0	1011104002	KIB D	Tanah Bangunan Pendidikan Dan Latihan (sekolah)	5.19	M2	Jl. Cikini Raya No 87	1950
1	1011104002	KIB D	Tanah Bangunan Pendidikan Dan Latihan (sekolah)	1.80	M2	JALAN PERUK NO. 32	1951
2	1011104002	KIB D	Tanah Bangunan Pendidikan Dan Latihan (sekolah)	4.18	M2	Jl.Perwira No.10 Rt.002 / Rw.008	1951
3	1011104002	KIB D	Tanah Bangunan Pendidikan Dan Latihan (sekolah)	9.70	M2	JL PALMERAH BARAT NO. 59	1959
4	1011104002	KIB D	Tanah Bangunan Pendidikan Dan Latihan (sekolah)	775.00	M2	Jl. Masjid Nur No.33 Rt.002 / 010	1962

```
[57] # Mengecek apakah ada deret yang kosong
     data.empty

False

[58] # Melihat ukuran dari data
     data.size

7000
```

Figure 7. Preprocessing/Cleaning Data

c. Data Transformation



```
[59] # Menetapkan Variabel independen
     x = data.drop(['Tahun', 'KIB', 'Satuan','Jenis_Barang', 'Alamat'], axis = 1)
     x.head()
```

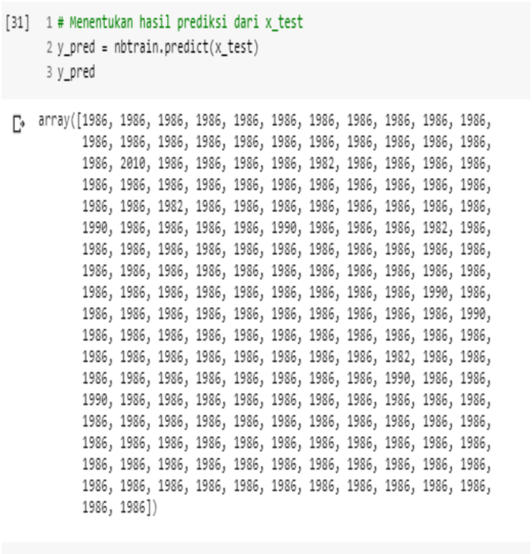
	Kode_Barang	Ukuran
0	1011104002	5.19
1	1011104002	1.80
2	1011104002	4.18
3	1011104002	9.70
4	1011104002	775.00

```
[60] # Menetapkan Variabel independen

     y = data['Tahun']
     y.head()

0    1950
1    1951
2    1951
3    1959
4    1962
Name: Tahun, dtype: int64
```

Figure 8. Data Transformation

Classification Data Naïve Bayes Method

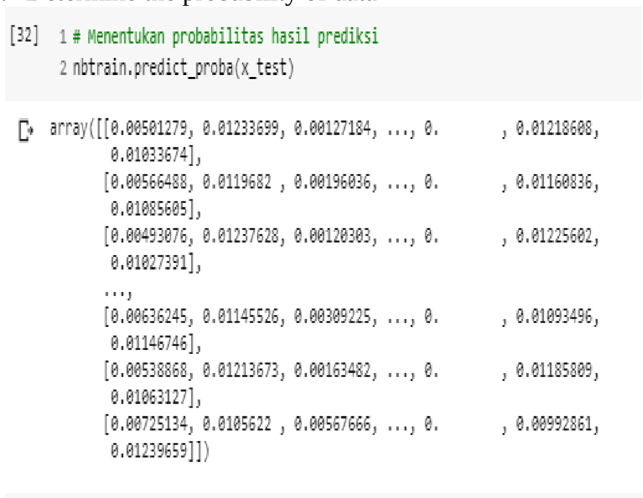
1. Determining the predictive data



```
[31] # Menentukan hasil prediksi dari x_test
     y_pred = nbtrain.predict(x_test)
     y_pred

array([1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 2010, 1986, 1986, 1986, 1986, 1982, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1982, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1990, 1986, 1986, 1986, 1986, 1990, 1986, 1986, 1986, 1982, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1990, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1990,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1982, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1990, 1986, 1986,
       1990, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986, 1986,
       1986, 1986])
```

Figure 9. Determining Predictive Data

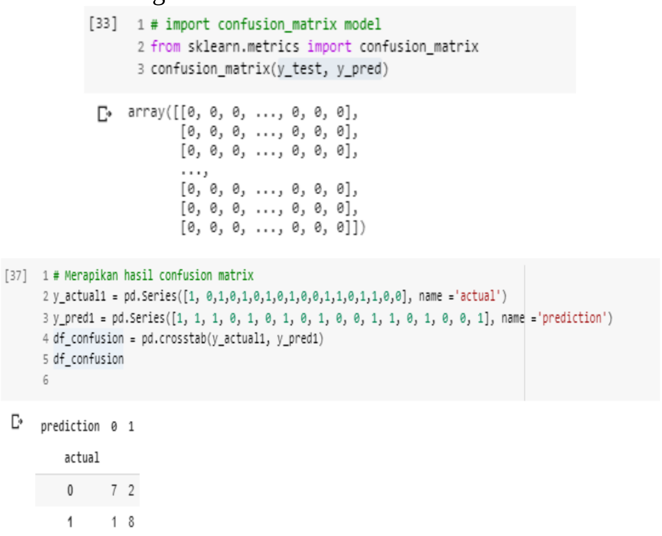
2. Determine the probability of data



```
[32] # Menentukan probabilitas hasil prediksi
     nbtrain.predict_proba(x_test)

array([[0.00501279, 0.01233699, 0.00127184, ..., 0.        , 0.01218608,
        0.01033674],
       [0.00566488, 0.0119682 , 0.00196036, ..., 0.        , 0.01160836,
        0.01085605],
       [0.00493076, 0.01237628, 0.00120303, ..., 0.        , 0.01225602,
        0.01027391],
       ...,
       [0.00636245, 0.01145526, 0.00309225, ..., 0.        , 0.01093496,
        0.01146746],
       [0.00538868, 0.01213673, 0.00163482, ..., 0.        , 0.01185809,
        0.01063127],
       [0.00725134, 0.0105622 , 0.00567666, ..., 0.        , 0.00992861,
        0.01239659]])
```

Figure 10. Determine the probability data

3. Determining Matrix Model:



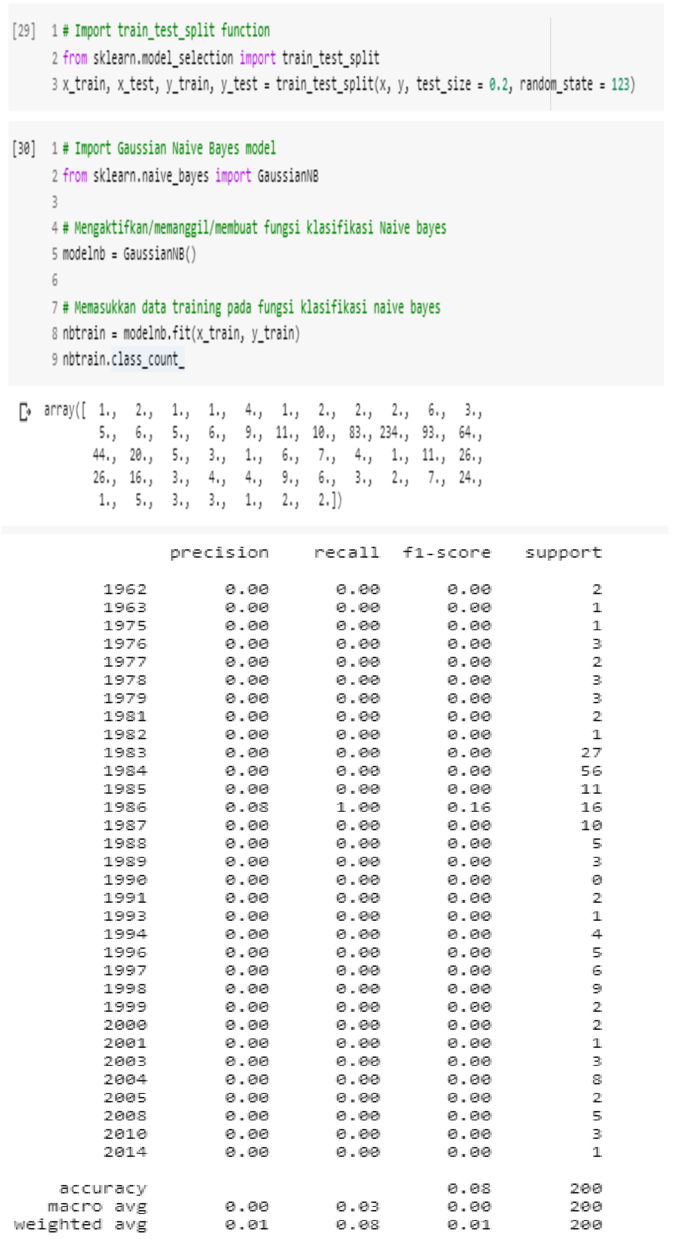
```
[33] # import confusion_matrix model
     from sklearn.metrics import confusion_matrix
     confusion_matrix(y_test, y_pred)

array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])

[37] # Merapikan hasil confusion matrix
     y_actual1 = pd.Series([1, 0,1,0,1,0,1,0,1,0,0,1,1,0,1,1,0,0], name ='actual')
     y_pred1 = pd.Series([1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1], name ='prediction')
     df_confusion = pd.crosstab(y_actual1, y_pred1)
     df_confusion
```

prediction	0	1
actual		
0	7	2
1	1	8

Figure 11. Determining Matrix Model

d. Specifying Process Data Mining

Apply an algorithm to classify the data.



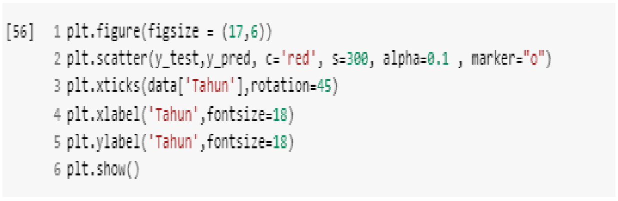
```
[29]  # Import train_test_split function
      from sklearn.model_selection import train_test_split
      x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 123)
```

```
[30]  # Import Gaussian Naive Bayes model
      from sklearn.naive_bayes import GaussianNB

      # Mengaktifkan/memanggil/membuat fungsi klasifikasi Naive bayes
      modelnb = GaussianNB()

      # Memasukkan data training pada fungsi klasifikasi naive bayes
      nbtrain = modelnb.fit(x_train, y_train)
      nbtrain.class_count_
```

```
array([  1.,   2.,   1.,   1.,   4.,   1.,   2.,   2.,   2.,   6.,   3.,
         5.,   6.,   5.,   6.,   9.,  11.,  10.,  83., 234.,  93.,  64.,
        44.,  20.,   5.,   3.,   1.,   6.,   7.,   4.,   1.,  11.,  26.,
        26.,  16.,   3.,   4.,   4.,   9.,   6.,   3.,   2.,   7.,  24.,
         1.,   5.,   3.,   3.,   1.,   2.,   2.])
```

```
              precision    recall  f1-score   support

        1962       0.00      0.00      0.00         2
        1963       0.00      0.00      0.00         1
        1975       0.00      0.00      0.00         1
        1976       0.00      0.00      0.00         3
        1977       0.00      0.00      0.00         2
        1978       0.00      0.00      0.00         3
        1979       0.00      0.00      0.00         3
        1981       0.00      0.00      0.00         2
        1982       0.00      0.00      0.00         1
        1983       0.00      0.00      0.00        27
        1984       0.00      0.00      0.00        56
        1985       0.00      0.00      0.00        11
        1986       0.08      1.00      0.16        16
        1987       0.00      0.00      0.00        10
        1988       0.00      0.00      0.00         5
        1989       0.00      0.00      0.00         3
        1990       0.00      0.00      0.00         0
        1991       0.00      0.00      0.00         2
        1993       0.00      0.00      0.00         1
        1994       0.00      0.00      0.00         4
        1996       0.00      0.00      0.00         5
        1997       0.00      0.00      0.00         6
        1998       0.00      0.00      0.00         9
        1999       0.00      0.00      0.00         2
        2000       0.00      0.00      0.00         2
        2001       0.00      0.00      0.00         1
        2003       0.00      0.00      0.00         3
        2004       0.00      0.00      0.00         8
        2005       0.00      0.00      0.00         2
        2008       0.00      0.00      0.00         5
        2010       0.00      0.00      0.00         3
        2014       0.00      0.00      0.00         1

    accuracy                           0.08       200
   macro avg       0.00      0.03      0.00       200
weighted avg       0.01      0.08      0.01       200
```

Figure 12. Specifying Process Data Mining



```
[56]  plt.figure(figsize = (17,6))
      plt.scatter(y_test,y_pred, c='red', s=300, alpha=0.1 , marker="o")
      plt.xticks(data['Tahun'],rotation=45)
      plt.xlabel('Tahun',fontsize=18)
      plt.ylabel('Tahun',fontsize=18)
      plt.show()
```

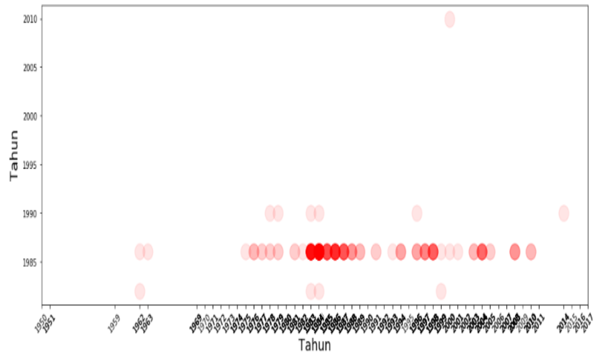
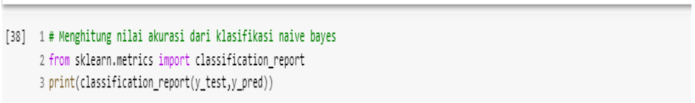


Figure 13. K-Means

e. Application of Interpretation / Evaluation



```
[38]  # Menghitung nilai akurasi dari klasifikasi naive bayes
      from sklearn.metrics import classification_report
      print(classification_report(y_test,y_pred))
```

Figure 14. Interpretation/Evaluation

IV. CONCLUSION

Based on the discussion above, it can be concluded that the asset data influences the utilization of the performance program. Naïve Bayes method successfully classifies 900 data from 1000 data tested. So the Naïve Bayes method succeeded in making the accuracy of the data percentage of accuracy 71.42%.

REFERENCES

[1] B.Liu. Sentiment Analysis and Opinion Mining. San Rafael : Morgan and Claypool Publishers. 2012

[2] Han, J., Kamber, M., & Pei, J. Data Mining Concepts and Techniques (Third Edition)Elsevier Inc. 2012

[3] M.W.Berry and J.Kogan. Text Mining Analysis and Theory. Wiley:United Kingdom.2010

[4] O. Maimon and L. Rokach, Data Mining and Knowledge Discovery Handbo- ok. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[5] S. Russell and P. Norvig, Artificial Intelligence A Modern Approach. Upper Saddle River, New Jersey 07458: Pearson Education, Inc., 3 ed., 2010

[6] Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, (First Edition). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[7] X. Wu and V. Kumar, eds., The Top Ten Algorithms in Data Mining.Chapman and Hall, 2009

[8] Nia Rahma Kurnianda & Yunita Sartika Sari. *Analysis and Design of Information System for Self-Journal on Food Based Dietary Assessment Record for Diabetes Patients*. International Research Journal of Computer Science (IRJCS). Volume 06 Issue 5. 2018

[9] Suhendra & Ranggadara, Indra. Naïve Bayes Algorithm with Chi Square and NGram Feature for Reviewing Laptop Product on Amazon Site. International Research Journal of Computer Science, Vol 4, issue 12.2017

[10] Triana, Yaya Sudarya, and Astari Retnowardhani. "Enhance interval width of crime forecasting with ARIMA model-fuzzy alpha cut." Telkomnika 17.3 (2019): 1193-1201.