Computer Science

E-ISSN No : 2581-6195 | Volume : 3 | Issue : 1 | Jan-Feb 2020

IMPLEMENTATION REGRESSION ANALYSIS AND K-MEANS TO PREDICT AND CLASSIFY FIXED DATA ASSET

Handy Noviyarto

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, the method used are Regression Analysis and K-Means Method. The purpose of this study was to predict and classify fixed data asset using Regression Analysis and K-Means Method. This research was conducted using the Python programming language and the Visual Studio code.

KEYWORDS: Analysis Regression, K-means, Clustering, Data mining

1. 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 (SAP) (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 dependiharapkan can be obtained either by the government or the public, as well as dapatdiukur in units of money, including non-financial resources needed to provide services to the public and the sources of power in 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. Absolute asset security is done by completing the assets in question to a legal document. 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. As yet completed documents are Letter or Certificate of Land Ownership History certain land owned by the provincial government, incomplete documents such as letters leasing, handing over others. Table 1.1 lists the data that show the problem in asset Jakarta Education Agency

Based on the background of the issue and the importance of asset security systems in every activity of the company - the company, so in this study was taken the title "Implementation Regression Analysis And K-Means To Predict And Classify Fixed Asset Data In Education Authorities"

2. 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

Faculty of Computer Science, Mercu Buana University, Indonesia

HOW TO CITE THIS ARTICLE:
Handy Noviyarto (2020). Implementation Regression Analysis And K-Means To Predict And Classify Fixed Data Assetin Education Authorities, International Educational Journal of Science and Engineering (IEJSE), Vol: 3, Issue: 1, 01-05

Research Paper

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.

2.2 K Means

K-means is a clustering algorithm. The purpose of this algorithm is to divide data into groups. This algorithm accepts input in the form of data without class labels. This is different from supervised learning which accepts input in the form of vectors (x1, y1), (x2, y2), ..., (xi, yi), where xi is the data from a training data and yi is the class label for xi.

In this learning algorithm, the computer groups its own data into input without first knowing the target class. This learning is included in unsupervised learning. The input received is the data or object and the desired group (cluster). This algorithm will group data or objects into these groups. In each cluster there is a center point (centroid) that represents the cluster.

3. RESULTS AND DISCUSSION

3.1 Predictive Analysis (Regression Method)

a. Using 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)	4.180	M2	Jl.Perwira 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)	4.808	M2	jl.Pertanian Klender	1963
7	1011104002	KIB D	Tanah Bangunan Pendidikan Dan Latihan (sekolah)	2.363	M2	Jl Bulak Timur I/7 Klender	1963
8	1011104002	KIB D	Tanah Bangunan Pendidikan Dan Latihan (sekolah)	2.900	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	1.011084e+09	21.916909	1987.657000
std	1.407996e+05	113.472694	8.608685
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	4.431750	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')
```

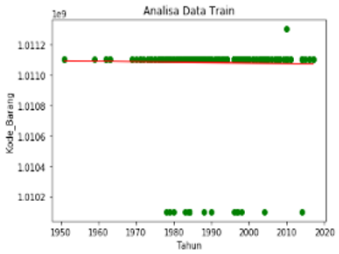

Figure 2: Fitting Simple Linear Regression in Training Set

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

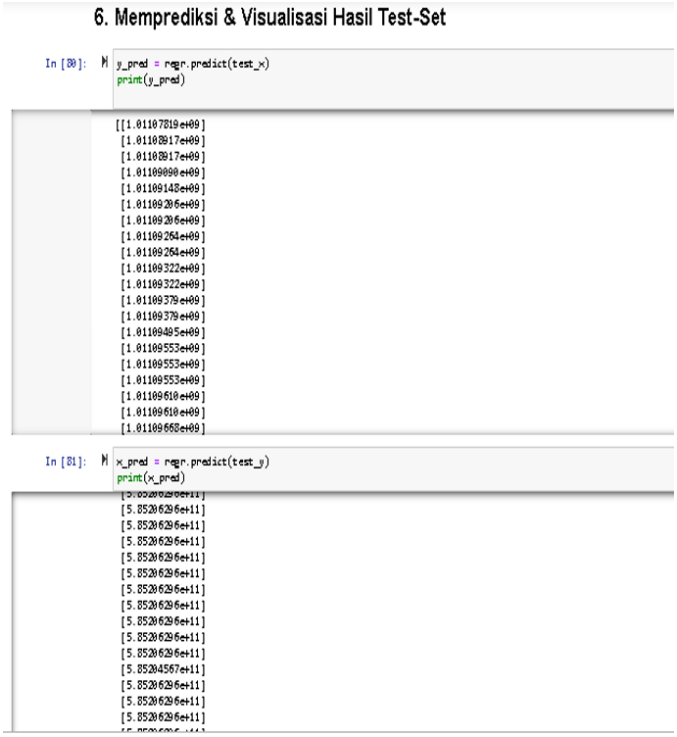

Figure 4: Predicting Result

And included the form of the plot

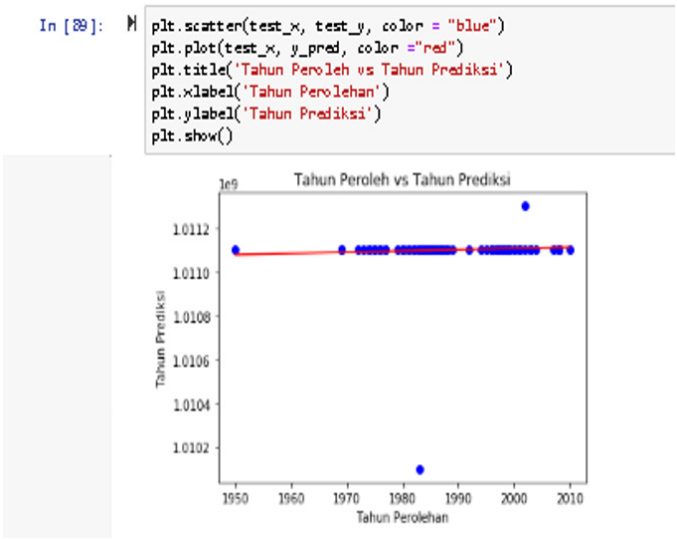

Figure 5: Regression Analysis

3.2 K-Means Method:

a. Screening Data: Before using the code for Data Processing, first input library that will be used:

Figure 6: Screening Data

b. Selecting Data

Before the data were used to do the cleansing / disaggregation that will grab

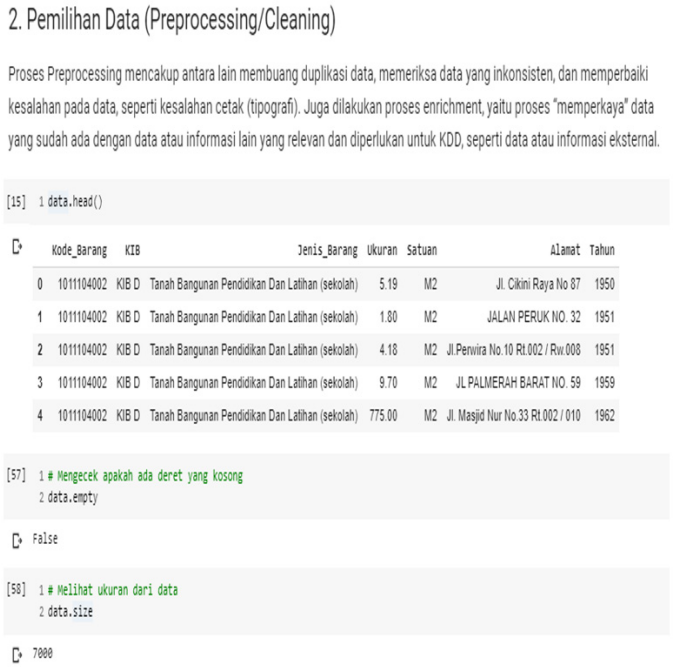

Figure 7: Selecting Data

c.Creating a Data Transformation

```
[59] 1 # Menetapkan Variabel independen
     2 X = data.drop(['Tahun', 'KIB', 'Satuan','Jenis_Barang', 'Alamat'], axis = 1)
     3 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] 1 # Menetapkan Variabel independen
     2
     3 y = data['Tahun']
     4 y.head()
```

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

Figure 8: Creating Data Transformation

Classification Data K-Means Method

1. Determining the predictive data

```
[31] 1 # Menentukan hasil prediksi dari x_test
     2 y_pred = nbtrain.predict(x_test)
     3 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] 1 # Menentukan probabilitas hasil prediksi
     2 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] 1 # import confusion_matrix model
     2 from sklearn.metrics import confusion_matrix
     3 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] 1 # Merapikan hasil confusion matrix
     2 y_actual1 = pd.Series([1, 0,1,0,1,0,1,0,1,0,0,1,1,0,1,1,0,0], name ='actual')
     3 y_pred1 = pd.Series([1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1], name ='prediction')
     4 df_confusion = pd.crosstab(y_actual1, y_pred1)
     5 df_confusion
     6
```

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

```
[30] 1 # Import Gaussian Naive Bayes model
     2 from sklearn.naive_bayes import GaussianNB
     3
     4 # Mengaktifkan/memanggil/membuat fungsi klasifikasi Naive bayes
     5 modelnb = GaussianNB()
     6
     7 # Memasukkan data training pada fungsi klasifikasi naive bayes
     8 nbtrain = modelnb.fit(x_train, y_train)
     9 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] 1 plt.figure(figsize = (17,6))
     2 plt.scatter(y_test,y_pred, c='red', s=300, alpha=0.1 , marker="o")
     3 plt.xticks(data['Tahun'],rotation=45)
     4 plt.xlabel('Tahun',fontsize=18)
     5 plt.ylabel('Tahun',fontsize=18)
     6 plt.show()
```

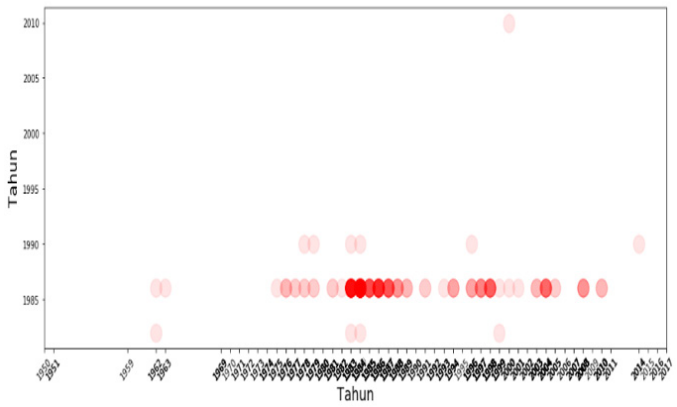

Figure 13: K-Means

e. Application of Interpretation / Evaluation

```
[38] 1 # Menghitung nilai akurasi dari klasifikasi naive bayes
     2 from sklearn.metrics import classification_report
     3 print(classification_report(y_test,y_pred))
```

Figure 14: Interpretation/Evaluation

4. CONCLUSION

The k-means method succeeded in making the accuracy of the data classification of DKI Jakarta Education authorities asset beneficiary percentage accuracy of 71.42%.

REFERENCE

1. X. Wu and V. Kumar, eds., The Top Ten Algorithms in Data Mining.Chapman and Hall, 2009.
2. S. Russell and P. Norvig, Artificial Intelligence A Modern Approach. Upper Saddle River, New Jersey 07458: Pearson Education, Inc., 3 ed., 2010.
3. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, (First Edition). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
4. O. Maimon and L. Rokach, Data Mining and Knowledge Discovery Handbo- ok. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.
5. 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
6. Ranggadara, Indra & Suhendra. Zachman Framework Approach for Designing Recruitment System Modules in HRIS Application (Case Study in PT. Karya Impian Teknologi Abadi).IJCSMC, Vol 7, issue 2.2018