

Implementasi Algoritma Naïve Bayes (Studi Kasus : Prediksi Kelulusan Mahasiswa) Menggunakan Bahasa Pemrograman C#.NET Dan LINQ

Indonesia

9/2/2016

Christian Yonathan S.

BAB 1 : ALGORITMA NAÏVE BAYES

Algoritma Naïve Bayes merupakan salah satu Algoritma untuk klasifikasi dengan pendekatan **probabilitas** dan **statistik** yang dikemukakan oleh ilmuwan asal Negara Inggris bernama **Thomas Bayes**. Algoritma ini, memprediksi peluang pada **masa depan** berdasarkan **pengalaman** pada **masa lalu** sehingga disebut **Teorema Bayes**. Teorema tersebut dikombinasikan dengan naïve, dimana terdapat asumsi kondisi antar atribut **saling bebas (*independent*)**. Klasifikasi Naïve Bayes, mengasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak memiliki hubungan ciri terhadap kelas yang lainnya.

Persamaan dari Teorema Bayes :

$$P(H|X) = \frac{P(X|H) * P(H)}{P(X)}$$

Keterangan :

X = data dengan class yang belum diketahui.

H = Hipotesis terhadap data X merupakan suatu kelas spesifik.

P (H|X) = probabilitas hipotesis H berdasarkan kondisi X.

P (X|H) = probabilitas X berdasarkan kondisi pada hipotesis H.

P (H) = probabilitas hipotesis H.

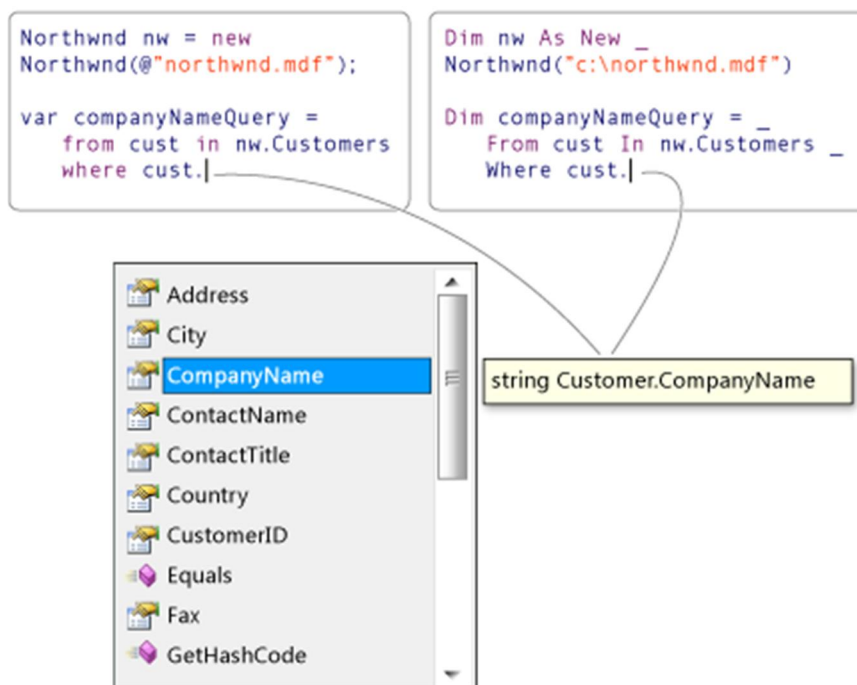
P (X) = probabilitas hipotesis X.

Berdasarkan dari persamaan di atas, persamaan tersebut dapat disesuaikan sebagai berikut :

$$P(C|F1 \dots FN) = \frac{P(C) * P(F1 \dots FN|C)}{P(F1 \dots FN)}$$

BAB 2 : LINQ

LINQ merupakan singkatan dari Language Integrated Query, yang merupakan inovasi perkembangan terbaru yang dikembangkan pada .NET Framework versi 3.5 yang berfungsi menjembatani antara "dunia" objek dengan "dunia" data. Secara tradisional, query "melawan" data diekspresikan sebagai *string* sederhana tanpa melakukan pengecekan saat *compile* atau *intelliSense*. Selanjutnya, kita juga harus mempelajari perbedaan bahasa query untuk setiap *data source* seperti : SQL Database, Dokumen XML, Web Services, dan masih banyak lagi. Anda bisa menulis LINQ queries pada bahasa C#.NET dan VB.NET, yang didukung juga dengan SQL Server Database, Dokument XML, ADO.NET Dataset, dan bentuk *Collection* dari objek salah satunya *IEnumerable* atau yang lebih umum *IEnumerable(Of T)* interface. LINQ juga mendukung untuk menyediakan *third parties* sebagai web services and implementasi pada database lain.



Gambar 1 Contoh Penggunaan LINQ

BAB 3 : IMPLEMENTASI PROGRAM

Buatlah sebuah form, seperti gambar di bawah ini :

Gambar 2 Contoh Form

Abjad	Tipe Object	Nama Object
A	Textbox	textLokasiFile
B	Button	tombolCariFile
C	ComboBox	comboJK
D	ComboBox	comboMahasiswa
E	ComboBox	comboPernikahan
F	ComboBox	comboIPK
G	ComboBox	comboKelulusan
H	Button	tombolHitung
I	DataGridView	dataGridViewTraining

Tabel 1 Keterangan Form

Source code pada **tombolCariFile** :

```
31     try
32     {
33         OpenFileDialog OFD = new OpenFileDialog();
34         OFD.Filter = "File(*.xls; *.xlsx) | *.xlsx;";
35         OFD.Title = "Pilihlah File Excel";
36
37         if (OFD.ShowDialog() == DialogResult.OK)
38         {
39             textLokasiFile.Text = OFD.FileName;
40             OleDbConnection ODBC = null;
41
42             if (System.IO.Path.GetExtension(OFD.FileName).ToLower().Trim().Equals(".xlsx"))
43             {
44                 ODBC = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data Source=" + textLokasiFile.Text + ";" +
45                 "Extended Properties=\\Excel 12.0; HDR=Yes;");
46             }
47
48             else
49             {
50                 ODBC = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" + textLokasiFile.Text + ";" +
51                 "Extended Properties=\\Excel 8.0; HDR=Yes;");
52             }
53
54             OleDbDataAdapter ODBA = new OleDbDataAdapter("select * from [Sheet1$]", ODBC);
55             DataTable DT = new DataTable(); ODBA.Fill(DT);
56             dataGridViewTraining.DataSource = DT;
57             isiComboKriteria(DT);
58         }
59     }
60
61     catch (Exception er)
62     {
63         MessageBox.Show("Error\n" + er.Message, "Informasi Kesalahan", MessageBoxButtons.OK, MessageBoxIcon.Error);
64     }
```

Gambar 3 Source Code pada tombolCariFile

Source code pada method **isiComboKriteria** :

```
67 void isiComboKriteria(DataTable pIsian)
68 {
69     var COMBO_JK = (from xIsian in pIsian.AsEnumerable()
70                     select xIsian.Field<String>("JK")).Distinct().ToList();
71     comboJK.ValueMember = "JK"; comboJK.DisplayMember = "JK"; comboJK.DataSource = COMBO_JK;
72
73     var COMBO_SM = (from xIsian in pIsian.AsEnumerable()
74                     select xIsian.Field<String>("Status_Mahasiswa")).Distinct().ToList();
75     comboMahasiswa.ValueMember = "Status_Mahasiswa"; comboMahasiswa.DisplayMember = "Status_Mahasiswa"; comboMahasiswa.DataSource = COMBO_SM;
76
77     var COMBO_PK = (from xIsian in pIsian.AsEnumerable()
78                     select xIsian.Field<String>("Status_Pernikahan")).Distinct().ToList();
79     comboPernikahan.ValueMember = "Status_Pernikahan"; comboPernikahan.DisplayMember = "Status_Pernikahan"; comboPernikahan.DataSource = COMBO_PK;
80
81     var COMBO_IP = (from xIsian in pIsian.AsEnumerable()
82                     select xIsian.Field<Double>("IPK")).Distinct().ToList();
83     comboIPK.ValueMember = "IPK"; comboIPK.DisplayMember = "IPK"; comboIPK.DataSource = COMBO_IP;
84
85     var COMBO_KL = (from xIsian in pIsian.AsEnumerable()
86                     select xIsian.Field<String>("Status_Kelulusan")).Distinct().ToList();
87     comboKelulusan.ValueMember = "Status_Kelulusan"; comboKelulusan.DisplayMember = "Status_Kelulusan"; comboKelulusan.DataSource = COMBO_KL;
88
89     MessageBox.Show("Selesai Upload File",
90                     "Inforasi Selesai Upload File",
91                     MessageBoxButtons.OK,
92                     MessageBoxIcon.Information);
93 }
```

Gambar 4 Source Code pada Method isiComboKriteria

Source code pada **tombolHitung** :

```
97     var DATATABLE_GV = dataGridViewTraining.DataSource as DataTable;
98     Console.WriteLine("Jumlah Records = " + (DATATABLE_GV.Rows.Count - 1));
99
100    double JUMLAH_TEPAT = (double)
101    (
102        from xGVLengkap in DATATABLE_GV.AsEnumerable()
103        where xGVLengkap.Field<String>("Status_Kelulusan") == "Tepat"
104        select xGVLengkap.Field<String>("Status_Kelulusan")
105    ).Count();
106    Console.WriteLine("Jumlah Tepat = " + JUMLAH_TEPAT);
107
108    double JUMLAH_TERLAMBAT = (double)
109    (
110        from xGVLengkap in DATATABLE_GV.AsEnumerable()
111        where xGVLengkap.Field<String>("Status_Kelulusan") == "Terlambat"
112        select xGVLengkap.Field<String>("Status_Kelulusan")
113    ).Count();
114    Console.WriteLine("Jumlah Terlambat = " + JUMLAH_TERLAMBAT);
115
116    double JUMLAH_JK = (double)
117    (
118        from xGVLengkap in DATATABLE_GV.AsEnumerable()
119        where xGVLengkap.Field<String>("JK") == comboJK.SelectedItem.ToString()
120        select xGVLengkap.Field<String>("JK")
121    ).Count();
122    Console.WriteLine("Jumlah JK = " + JUMLAH_JK);
```

Gambar 5 Source Code pada tombolHitung #1

```
124    double JUMLAH_STATUS_MAHASISWA = (double)
125    (
126        from xGVLengkap in DATATABLE_GV.AsEnumerable()
127        where xGVLengkap.Field<String>("Status_Mahasiswa") == comboMahasiswa.SelectedItem.ToString()
128        select xGVLengkap.Field<String>("Status_Mahasiswa")
129    ).Count();
130    Console.WriteLine("Jumlah SM = " + JUMLAH_STATUS_MAHASISWA);
131
132    double JUMLAH_STATUS_PERNIKAHAN = (double)
133    (
134        from xGVLengkap in DATATABLE_GV.AsEnumerable()
135        where xGVLengkap.Field<String>("Status_Pernikahan") == comboPernikahan.SelectedItem.ToString()
136        select xGVLengkap.Field<String>("Status_Pernikahan")
137    ).Count();
138    Console.WriteLine("Jumlah SP = " + JUMLAH_STATUS_PERNIKAHAN);
139
140    double JUMLAH_IPK = (double)
141    (
142        from xGVLengkap in DATATABLE_GV.AsEnumerable()
143        where xGVLengkap.Field<Double>("IPK") == Double.Parse(comboIPK.SelectedItem.ToString())
144        select xGVLengkap.Field<Double>("IPK")
145    ).Count();
146    Console.WriteLine("Jumlah IPK = " + JUMLAH_IPK);
147
148    double JK_TEPAT = JUMLAH_JK / JUMLAH_TEPAT;
149    Console.WriteLine("Jumlah JK Tepat = " + JK_TEPAT);
150
151    double JK_TERLAMBAT = JUMLAH_JK / JUMLAH_TERLAMBAT;
152    Console.WriteLine("Jumlah JK Terlambat = " + JK_TERLAMBAT);
153
154    double SM_TEPAT = JUMLAH_STATUS_MAHASISWA / JUMLAH_TEPAT;
155    Console.WriteLine("Jumlah SM Tepat = " + SM_TEPAT);
```

Gambar 6 Source Code pada tombolHitung #2

```

157     double SM_TERLAMBAT = JUMLAH_STATUS_MAHASISWA / JUMLAH_TERLAMBAT;
158     Console.WriteLine("Jumlah SM Terlambat = " + SM_TERLAMBAT);
159
160     double SP_TEPAT = JUMLAH_STATUS_PERNIKAHAN / JUMLAH_TEPAT;
161     Console.WriteLine("Jumlah SP Tepat = " + SP_TEPAT);
162
163     double SP_TERLAMBAT = JUMLAH_STATUS_PERNIKAHAN / JUMLAH_TERLAMBAT;
164     Console.WriteLine("Jumlah SP Terlambat = " + SP_TERLAMBAT);
165
166     double IPK_TEPAT = JUMLAH_IPK / JUMLAH_TEPAT;
167     Console.WriteLine("Jumlah IPK Tepat = " + IPK_TEPAT);
168
169     double IPK_TERLAMBAT = JUMLAH_IPK / JUMLAH_TERLAMBAT;
170     Console.WriteLine("Jumlah IPK Terlambat = " + IPK_TERLAMBAT);
171
172     double TEPAT = JK_TEPAT * SM_TEPAT * SP_TEPAT * IPK_TERLAMBAT * (JUMLAH_TEPAT / (DATATABLE_GV.Rows.Count - 1));
173     double TERLAMBAT = JK_TERLAMBAT * SM_TERLAMBAT * SP_TERLAMBAT * IPK_TERLAMBAT * (JUMLAH_TERLAMBAT / (DATATABLE_GV.Rows.Count - 1));
174
175     Console.WriteLine("Nilai Tepat = " + TEPAT + "\nNilai Terlambat = " + TERLAMBAT);
176     if (TEPAT > TERLAMBAT)
177         MessageBox.Show("Lulus Tepat");
178     else
179         MessageBox.Show("Lulus Terlambat");

```

Gambar 7 Source Code pada tombolHitung #3

Q & A

- 1) Q : Dimana letak koding, penggunaan LINQ?
A : pada saat method **isiComboKriteria** dan **tombolHitung** ditekan.
- 2) Q : dimana letak koding, Algoritma Naïve Bayes?
A : pada saat **tombolHitung** ditekan. Lihat pada baris 172 dan 173, yang merupakan inti dari Algoritma Naïve Bayes.
- 3) Q : jika pada contoh yang diberikan, hanya terdapat 2 label yaitu "Tepat" dan "Terlambat" lalu bagaimana jika mempunyai 3 atau 4 label?
A : maka tambahkan bagian koding untuk setiap atribut terhadap label baru tersebut.
- 4) Q : kenapa pada gambar 1, terdapat keyword properties atribut sedangkan saat mengkodingnya tidak ada?
A : karena LINQ yang sedang dibuat ini, merupakan LINQ tanpa terhubung SQL Server Database. Sedangkan pada gambar 1, terhubung. Selanjutnya, pada gambar 1 dibuat dengan cara (**Add > New Item > Linq to SQL Classes**) nah LINQ yang kita buat dengan cara koding secara langsung.

REFERENSI PENULISAN

Referensi Algoritma Naïve Bayes

<https://id.scribd.com/doc/218550049/Algoritma-Naive-Bayes-Data-Mining>

<https://ditut113070242.wordpress.com/2011/03/22/algoritma-naive-bayes-classification/>

http://education-programmer.blogspot.co.id/2013/01/naive-bayes-algorithm_22.html

<http://www.slideshare.net/ashrafmath/naive-bayes-15644818>

<http://www.slideshare.net/chowdhury343/naive-bayes-presentation>

<http://www.metode-algoritma.com/2013/06/conoth-perhitungan-naive-bayes.html>

Referensi LINQ

<https://msdn.microsoft.com/en-us/library/mt693042.aspx>

<https://msdn.microsoft.com/en-us/library/bb308959.aspx>

[https://msdn.microsoft.com/en-us/library/system.linq\(v=vs.111\).aspx](https://msdn.microsoft.com/en-us/library/system.linq(v=vs.111).aspx)

***NB**

Data dummy, bisa didownload

<https://drive.google.com/open?id=0Bxn6E8TrwP2AaXBmaGNRV3pRZkk>