Final Project_Week 1_Morgenstern

June 25, 2019

1 Final project

Using the Breast Cancer Wisconsin (Diagnostic) Data Set, I am aiming to predict whether the cancer is benign or malignant.

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. n the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

The data set was downloaded from Kaggle: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data#data.csv

Attribute Information:

1) ID number 2) Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter) b) texture (standard deviation of gray-scale values) c) perimeter d) area e) smoothness (local variation in radius lengths) f) compactness (perimeter^2 / area - 1.0) g) concavity (severity of concave portions of the contour) h) concave points (number of concave portions of the contour) i) symmetry j) fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

All feature values are recoded with four significant digits.

Class distribution: 357 benign, 212 malignant

1.1 Step 1 - Loading and cleaning of data

```
In [41]: # import dataset downloaded from Kaggle and display first rows
         import pandas as pd
         df = pd.read_csv('data.csv')
         df.head()
```

```
Out[41]:          id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
         0    842302         M        17.99         10.38          122.80     1001.0
         1    842517         M        20.57         17.77          132.90     1326.0
```

```
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   ...  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0  ...          17.33           184.60      2019.0            0.1622
1  ...          23.41           158.80      1956.0            0.1238
2  ...          25.53           152.50      1709.0            0.1444
3  ...          26.50            98.87       567.7            0.2098
4  ...          16.67           152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0             0.6656           0.7119                0.2654          0.4601
1             0.1866           0.2416                0.1860          0.2750
2             0.4245           0.4504                0.2430          0.3613
3             0.8663           0.6869                0.2575          0.6638
4             0.2050           0.4000                0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32
0                  0.11890          NaN
1                  0.08902          NaN
2                  0.08758          NaN
3                  0.17300          NaN
4                  0.07678          NaN

[5 rows x 33 columns]
```

In [42]:
```
# display type of data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
id                         569 non-null int64
diagnosis                  569 non-null object
radius_mean                569 non-null float64
texture_mean               569 non-null float64
perimeter_mean             569 non-null float64
area_mean                  569 non-null float64
smoothness_mean            569 non-null float64
compactness_mean           569 non-null float64
```

```
concavity_mean             569 non-null float64
concave points_mean        569 non-null float64
symmetry_mean              569 non-null float64
fractal_dimension_mean     569 non-null float64
radius_se                  569 non-null float64
texture_se                 569 non-null float64
perimeter_se               569 non-null float64
area_se                    569 non-null float64
smoothness_se              569 non-null float64
compactness_se             569 non-null float64
concavity_se               569 non-null float64
concave points_se          569 non-null float64
symmetry_se                569 non-null float64
fractal_dimension_se       569 non-null float64
radius_worst               569 non-null float64
texture_worst              569 non-null float64
perimeter_worst            569 non-null float64
area_worst                 569 non-null float64
smoothness_worst           569 non-null float64
compactness_worst          569 non-null float64
concavity_worst            569 non-null float64
concave points_worst       569 non-null float64
symmetry_worst             569 non-null float64
fractal_dimension_worst    569 non-null float64
Unnamed: 32                0 non-null float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

```
In [43]: # check for missing values in data
         print df.isnull().sum()
```

```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
```

```
smoothness_se                0
compactness_se               0
concavity_se                 0
concave points_se            0
symmetry_se                  0
fractal_dimension_se         0
radius_worst                 0
texture_worst                0
perimeter_worst              0
area_worst                   0
smoothness_worst             0
compactness_worst            0
concavity_worst              0
concave points_worst         0
symmetry_worst               0
fractal_dimension_worst      0
Unnamed: 32                569
dtype: int64
```

In [44]:
```
# drop column 32 because of missing values
df1 = df.drop('Unnamed: 32', axis=1)
df1
```

Out[44]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	842302	M	17.990	10.38	122.80	1001.0	
1	842517	M	20.570	17.77	132.90	1326.0	
2	84300903	M	19.690	21.25	130.00	1203.0	
3	84348301	M	11.420	20.38	77.58	386.1	
4	84358402	M	20.290	14.34	135.10	1297.0	
5	843786	M	12.450	15.70	82.57	477.1	
6	844359	M	18.250	19.98	119.60	1040.0	
7	84458202	M	13.710	20.83	90.20	577.9	
8	844981	M	13.000	21.82	87.50	519.8	
9	84501001	M	12.460	24.04	83.97	475.9	
10	845636	M	16.020	23.24	102.70	797.8	
11	84610002	M	15.780	17.89	103.60	781.0	
12	846226	M	19.170	24.80	132.40	1123.0	
13	846381	M	15.850	23.95	103.70	782.7	
14	84667401	M	13.730	22.61	93.60	578.3	
15	84799002	M	14.540	27.54	96.73	658.8	
16	848406	M	14.680	20.13	94.74	684.5	
17	84862001	M	16.130	20.68	108.10	798.8	
18	849014	M	19.810	22.15	130.00	1260.0	
19	8510426	B	13.540	14.36	87.46	566.3	
20	8510653	B	13.080	15.71	85.63	520.0	
21	8510824	B	9.504	12.44	60.34	273.9	
22	8511133	M	15.340	14.26	102.50	704.4	

23	851509	M	21.160	23.04	137.20	1404.0
24	852552	M	16.650	21.38	110.00	904.6
25	852631	M	17.140	16.40	116.00	912.7
26	852763	M	14.580	21.53	97.41	644.8
27	852781	M	18.610	20.25	122.10	1094.0
28	852973	M	15.300	25.27	102.40	732.4
29	853201	M	17.570	15.05	115.00	955.1
..	...	...	...	...	...	...
539	921362	B	7.691	25.44	48.34	170.4
540	921385	B	11.540	14.44	74.65	402.9
541	921386	B	14.470	24.99	95.81	656.4
542	921644	B	14.740	25.42	94.70	668.6
543	922296	B	13.210	28.06	84.88	538.4
544	922297	B	13.870	20.70	89.77	584.8
545	922576	B	13.620	23.23	87.19	573.2
546	922577	B	10.320	16.35	65.31	324.9
547	922840	B	10.260	16.58	65.85	320.8
548	923169	B	9.683	19.34	61.05	285.7
549	923465	B	10.820	24.21	68.89	361.6
550	923748	B	10.860	21.48	68.51	360.5
551	923780	B	11.130	22.44	71.49	378.4
552	924084	B	12.770	29.43	81.35	507.9
553	924342	B	9.333	21.94	59.01	264.0
554	924632	B	12.880	28.92	82.50	514.3
555	924934	B	10.290	27.61	65.67	321.4
556	924964	B	10.160	19.59	64.73	311.7
557	925236	B	9.423	27.88	59.26	271.3
558	925277	B	14.590	22.68	96.39	657.1
559	925291	B	11.510	23.93	74.52	403.5
560	925292	B	14.050	27.15	91.38	600.4
561	925311	B	11.200	29.37	70.67	386.0
562	925622	M	15.220	30.62	103.40	716.9
563	926125	M	20.920	25.09	143.00	1347.0
564	926424	M	21.560	22.39	142.00	1479.0
565	926682	M	20.130	28.25	131.20	1261.0
566	926954	M	16.600	28.08	108.30	858.1
567	927241	M	20.600	29.33	140.10	1265.0
568	92751	B	7.760	24.54	47.92	181.0

	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	\
0	0.11840	0.27760	0.300100	0.147100	
1	0.08474	0.07864	0.086900	0.070170	
2	0.10960	0.15990	0.197400	0.127900	
3	0.14250	0.28390	0.241400	0.105200	
4	0.10030	0.13280	0.198000	0.104300	
5	0.12780	0.17000	0.157800	0.080890	
6	0.09463	0.10900	0.112700	0.074000	
7	0.11890	0.16450	0.093660	0.059850	

8	0.12730	0.19320	0.185900	0.093530
9	0.11860	0.23960	0.227300	0.085430
10	0.08206	0.06669	0.032990	0.033230
11	0.09710	0.12920	0.099540	0.066060
12	0.09740	0.24580	0.206500	0.111800
13	0.08401	0.10020	0.099380	0.053640
14	0.11310	0.22930	0.212800	0.080250
15	0.11390	0.15950	0.163900	0.073640
16	0.09867	0.07200	0.073950	0.052590
17	0.11700	0.20220	0.172200	0.102800
18	0.09831	0.10270	0.147900	0.094980
19	0.09779	0.08129	0.066640	0.047810
20	0.10750	0.12700	0.045680	0.031100
21	0.10240	0.06492	0.029560	0.020760
22	0.10730	0.21350	0.207700	0.097560
23	0.09428	0.10220	0.109700	0.086320
24	0.11210	0.14570	0.152500	0.091700
25	0.11860	0.22760	0.222900	0.140100
26	0.10540	0.18680	0.142500	0.087830
27	0.09440	0.10660	0.149000	0.077310
28	0.10820	0.16970	0.168300	0.087510
29	0.09847	0.11570	0.098750	0.079530
..	...	...	...	...
539	0.08668	0.11990	0.092520	0.013640
540	0.09984	0.11200	0.067370	0.025940
541	0.08837	0.12300	0.100900	0.038900
542	0.08275	0.07214	0.041050	0.030270
543	0.08671	0.06877	0.029870	0.032750
544	0.09578	0.10180	0.036880	0.023690
545	0.09246	0.06747	0.029740	0.024430
546	0.09434	0.04994	0.010120	0.005495
547	0.08877	0.08066	0.043580	0.024380
548	0.08491	0.05030	0.023370	0.009615
549	0.08192	0.06602	0.015480	0.008160
550	0.07431	0.04227	0.000000	0.000000
551	0.09566	0.08194	0.048240	0.022570
552	0.08276	0.04234	0.019970	0.014990
553	0.09240	0.05605	0.039960	0.012820
554	0.08123	0.05824	0.061950	0.023430
555	0.09030	0.07658	0.059990	0.027380
556	0.10030	0.07504	0.005025	0.011160
557	0.08123	0.04971	0.000000	0.000000
558	0.08473	0.13300	0.102900	0.037360
559	0.09261	0.10210	0.111200	0.041050
560	0.09929	0.11260	0.044620	0.043040
561	0.07449	0.03558	0.000000	0.000000
562	0.10480	0.20870	0.255000	0.094290
563	0.10990	0.22360	0.317400	0.147400

```
564          0.11100           0.11590         0.243900              0.138900
565          0.09780           0.10340         0.144000              0.097910
566          0.08455           0.10230         0.092510              0.053020
567          0.11780           0.27700         0.351400              0.152000
568          0.05263           0.04362         0.000000              0.000000

     ...  radius_worst  texture_worst  perimeter_worst  area_worst  \
0    ...        25.380          17.33           184.60      2019.0
1    ...        24.990          23.41           158.80      1956.0
2    ...        23.570          25.53           152.50      1709.0
3    ...        14.910          26.50            98.87       567.7
4    ...        22.540          16.67           152.20      1575.0
5    ...        15.470          23.75           103.40       741.6
6    ...        22.880          27.66           153.20      1606.0
7    ...        17.060          28.14           110.60       897.0
8    ...        15.490          30.73           106.20       739.3
9    ...        15.090          40.68            97.65       711.4
10   ...        19.190          33.88           123.80      1150.0
11   ...        20.420          27.28           136.50      1299.0
12   ...        20.960          29.94           151.70      1332.0
13   ...        16.840          27.66           112.00       876.5
14   ...        15.030          32.01           108.80       697.7
15   ...        17.460          37.13           124.10       943.2
16   ...        19.070          30.88           123.40      1138.0
17   ...        20.960          31.48           136.80      1315.0
18   ...        27.320          30.88           186.80      2398.0
19   ...        15.110          19.26            99.70       711.2
20   ...        14.500          20.49            96.09       630.5
21   ...        10.230          15.66            65.13       314.9
22   ...        18.070          19.08           125.10       980.9
23   ...        29.170          35.59           188.00      2615.0
24   ...        26.460          31.56           177.00      2215.0
25   ...        22.250          21.40           152.40      1461.0
26   ...        17.620          33.21           122.40       896.9
27   ...        21.310          27.26           139.90      1403.0
28   ...        20.270          36.71           149.30      1269.0
29   ...        20.010          19.52           134.90      1227.0
..   ...           ...            ...              ...         ...
539  ...         8.678          31.89            54.49       223.6
540  ...        12.260          19.68            78.78       457.8
541  ...        16.220          31.73           113.50       808.9
542  ...        16.510          32.29           107.40       826.4
543  ...        14.370          37.17            92.48       629.6
544  ...        15.050          24.75            99.17       688.6
545  ...        15.350          29.09            97.58       729.8
546  ...        11.250          21.77            71.12       384.9
547  ...        10.830          22.04            71.08       357.4
548  ...        10.930          25.59            69.10       364.2
```

```
549  ...        13.030          31.45           83.90       505.6
550  ...        11.660          24.77           74.08       412.3
551  ...        12.020          28.26           77.80       436.6
552  ...        13.870          36.00           88.10       594.7
553  ...         9.845          25.05           62.86       295.8
554  ...        13.890          35.74           88.84       595.7
555  ...        10.840          34.91           69.57       357.6
556  ...        10.650          22.88           67.88       347.3
557  ...        10.490          34.24           66.50       330.6
558  ...        15.480          27.27          105.90       733.5
559  ...        12.480          37.16           82.28       474.2
560  ...        15.300          33.17          100.20       706.7
561  ...        11.920          38.30           75.19       439.6
562  ...        17.520          42.79          128.70       915.0
563  ...        24.290          29.41          179.10      1819.0
564  ...        25.450          26.40          166.10      2027.0
565  ...        23.690          38.25          155.00      1731.0
566  ...        18.980          34.12          126.70      1124.0
567  ...        25.740          39.42          184.60      1821.0
568  ...         9.456          30.37           59.16       268.6

     smoothness_worst  compactness_worst  concavity_worst  \
0             0.16220            0.66560          0.71190
1             0.12380            0.18660          0.24160
2             0.14440            0.42450          0.45040
3             0.20980            0.86630          0.68690
4             0.13740            0.20500          0.40000
5             0.17910            0.52490          0.53550
6             0.14420            0.25760          0.37840
7             0.16540            0.36820          0.26780
8             0.17030            0.54010          0.53900
9             0.18530            1.05800          1.10500
10            0.11810            0.15510          0.14590
11            0.13960            0.56090          0.39650
12            0.10370            0.39030          0.36390
13            0.11310            0.19240          0.23220
14            0.16510            0.77250          0.69430
15            0.16780            0.65770          0.70260
16            0.14640            0.18710          0.29140
17            0.17890            0.42330          0.47840
18            0.15120            0.31500          0.53720
19            0.14400            0.17730          0.23900
20            0.13120            0.27760          0.18900
21            0.13240            0.11480          0.08867
22            0.13900            0.59540          0.63050
23            0.14010            0.26000          0.31550
24            0.18050            0.35780          0.46950
25            0.15450            0.39490          0.38530
```

26	0.15250	0.66430	0.55390
27	0.13380	0.21170	0.34460
28	0.16410	0.61100	0.63350
29	0.12550	0.28120	0.24890
..	...	...	...
539	0.15960	0.30640	0.33930
540	0.13450	0.21180	0.17970
541	0.13400	0.42020	0.40400
542	0.10600	0.13760	0.16110
543	0.10720	0.13810	0.10620
544	0.12640	0.20370	0.13770
545	0.12160	0.15170	0.10490
546	0.12850	0.08842	0.04384
547	0.14610	0.22460	0.17830
548	0.11990	0.09546	0.09350
549	0.12040	0.16330	0.06194
550	0.10010	0.07348	0.00000
551	0.10870	0.17820	0.15640
552	0.12340	0.10640	0.08653
553	0.11030	0.08298	0.07993
554	0.12270	0.16200	0.24390
555	0.13840	0.17100	0.20000
556	0.12650	0.12000	0.01005
557	0.10730	0.07158	0.00000
558	0.10260	0.31710	0.36620
559	0.12980	0.25170	0.36300
560	0.12410	0.22640	0.13260
561	0.09267	0.05494	0.00000
562	0.14170	0.79170	1.17000
563	0.14070	0.41860	0.65990
564	0.14100	0.21130	0.41070
565	0.11660	0.19220	0.32150
566	0.11390	0.30940	0.34030
567	0.16500	0.86810	0.93870
568	0.08996	0.06444	0.00000

	concave points_worst	symmetry_worst	fractal_dimension_worst
0	0.26540	0.4601	0.11890
1	0.18600	0.2750	0.08902
2	0.24300	0.3613	0.08758
3	0.25750	0.6638	0.17300
4	0.16250	0.2364	0.07678
5	0.17410	0.3985	0.12440
6	0.19320	0.3063	0.08368
7	0.15560	0.3196	0.11510
8	0.20600	0.4378	0.10720
9	0.22100	0.4366	0.20750
10	0.09975	0.2948	0.08452

11	0.18100	0.3792	0.10480
12	0.17670	0.3176	0.10230
13	0.11190	0.2809	0.06287
14	0.22080	0.3596	0.14310
15	0.17120	0.4218	0.13410
16	0.16090	0.3029	0.08216
17	0.20730	0.3706	0.11420
18	0.23880	0.2768	0.07615
19	0.12880	0.2977	0.07259
20	0.07283	0.3184	0.08183
21	0.06227	0.2450	0.07773
22	0.23930	0.4667	0.09946
23	0.20090	0.2822	0.07526
24	0.20950	0.3613	0.09564
25	0.25500	0.4066	0.10590
26	0.27010	0.4264	0.12750
27	0.14900	0.2341	0.07421
28	0.20240	0.4027	0.09876
29	0.14560	0.2756	0.07919
..	...	...	...
539	0.05000	0.2790	0.10660
540	0.06918	0.2329	0.08134
541	0.12050	0.3187	0.10230
542	0.10950	0.2722	0.06956
543	0.07958	0.2473	0.06443
544	0.06845	0.2249	0.08492
545	0.07174	0.2642	0.06953
546	0.02381	0.2681	0.07399
547	0.08333	0.2691	0.09479
548	0.03846	0.2552	0.07920
549	0.03264	0.3059	0.07626
550	0.00000	0.2458	0.06592
551	0.06413	0.3169	0.08032
552	0.06498	0.2407	0.06484
553	0.02564	0.2435	0.07393
554	0.06493	0.2372	0.07242
555	0.09127	0.2226	0.08283
556	0.02232	0.2262	0.06742
557	0.00000	0.2475	0.06969
558	0.11050	0.2258	0.08004
559	0.09653	0.2112	0.08732
560	0.10480	0.2250	0.08321
561	0.00000	0.1566	0.05905
562	0.23560	0.4089	0.14090
563	0.25420	0.2929	0.09873
564	0.22160	0.2060	0.07115
565	0.16280	0.2572	0.06637
566	0.14180	0.2218	0.07820

```
567               0.26500          0.4087                  0.12400
568               0.00000          0.2871                  0.07039

[569 rows x 32 columns]
```

In [45]:
```
# display column names
df1.columns
```

Out[45]:
```
Index([u'id', u'diagnosis', u'radius_mean', u'texture_mean', u'perimeter_mean',
       u'area_mean', u'smoothness_mean', u'compactness_mean',
       u'concavity_mean', u'concave points_mean', u'symmetry_mean',
       u'fractal_dimension_mean', u'radius_se', u'texture_se', u'perimeter_se',
       u'area_se', u'smoothness_se', u'compactness_se', u'concavity_se',
       u'concave points_se', u'symmetry_se', u'fractal_dimension_se',
       u'radius_worst', u'texture_worst', u'perimeter_worst', u'area_worst',
       u'smoothness_worst', u'compactness_worst', u'concavity_worst',
       u'concave points_worst', u'symmetry_worst', u'fractal_dimension_worst'],
      dtype='object')
```

In [46]:
```
# drop id column because it is of no use for analysis
df2 = df1.drop('id', axis=1)
df2.head()
df2.columns
```

Out[46]:
```
Index([u'diagnosis', u'radius_mean', u'texture_mean', u'perimeter_mean',
       u'area_mean', u'smoothness_mean', u'compactness_mean',
       u'concavity_mean', u'concave points_mean', u'symmetry_mean',
       u'fractal_dimension_mean', u'radius_se', u'texture_se', u'perimeter_se',
       u'area_se', u'smoothness_se', u'compactness_se', u'concavity_se',
       u'concave points_se', u'symmetry_se', u'fractal_dimension_se',
       u'radius_worst', u'texture_worst', u'perimeter_worst', u'area_worst',
       u'smoothness_worst', u'compactness_worst', u'concavity_worst',
       u'concave points_worst', u'symmetry_worst', u'fractal_dimension_worst'],
      dtype='object')
```

In [68]:
```
# transform class labesls from original string representation ('M' for malignant and
from sklearn.preprocessing import LabelEncoder

X = df2.iloc[:, 1:].values
y = df2.iloc[:, 0].values

class_le = LabelEncoder()
y = class_le.fit_transform(df2['diagnosis'].values)
y
```

Out[68]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
```

```
       0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 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, 1, 1, 1, 1, 1, 1, 0])
```

In [73]:
```
# perform one-hot encoding
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

ohe = OneHotEncoder(categorical_features=[0])
ohe.fit_transform(X).toarray()
```

```
/Applications/anaconda2/lib/python2.7/site-packages/sklearn/preprocessing/_encoders.py:392: Dep
  "use the ColumnTransformer instead.", DeprecationWarning)
```

Out[73]:
```
array([[0.     , 0.     , 0.     , ..., 0.2654 , 0.4601 , 0.1189 ],
       [0.     , 0.     , 0.     , ..., 0.186  , 0.275  , 0.08902],
       [0.     , 0.     , 0.     , ..., 0.243  , 0.3613 , 0.08758],
       ...,
       [0.     , 0.     , 0.     , ..., 0.1418 , 0.2218 , 0.0782 ],
       [0.     , 0.     , 0.     , ..., 0.265  , 0.4087 , 0.124  ],
       [0.     , 1.     , 0.     , ..., 0.     , 0.2871 , 0.07039]])
```

In [75]:
```
# split data into training (80% of data) and test set (20% of data)
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, stratify=y,
```

In [ ]: