Rank Thu Jan 02 20, 09:01:32

Input

Features: preg, plasma_gluc, dias_bp, triceps_skinfold, 2hr_serum_insulin, bmi, db_ped_fcn, age
Meta attributes: Mahalanobis
Target: diab_yn

Ranks

	#	Info. gain	Gain ratio	Gini	ANOVA	χ^2	ReliefF	I
plasma_gluc		0.15281209803972373	0.07642970256359283	0.08880035449262308	191.9518393591607	113.66766428629387	0.03599999999999998	
age		0.09550815556304859	0.04779520522959059	0.05557865016233249	56.75876941808394	64.86325199216469	0.016285714285714268	
bmi		0.06716192701115653	0.033581446133582694	0.03728260706758796	68.93178481386173	44.81479417201122	0.01760571428571425	
preg		0.04859749956091797	0.024454681637858598	0.030618642385672745	44.420145748126686	38.948321682903256	0.007999999999999983	
2hr_serum_insulin		0.05480457933108385	0.029768547918399123	0.02982778408300507	6.296934682475103	5.9258918066380835	0.001307216494845349	
triceps_skinfold		0.03817244837850331	0.019098057211048177	0.02168883763946089	1.565826205303654	3.152083610955976	0.004066666666666656	
dias_bp		0.027107790884340233	0.01358861253248941	0.016403455896090047	28.37361807559602	21.49753718500616	0.00228888888888887	
db_ped_fcn		0.023413467017581513	0.011706760063178142	0.01442175104735699	25.86375168679761	15.842671246125526	-0.00020727945712524897	

Output

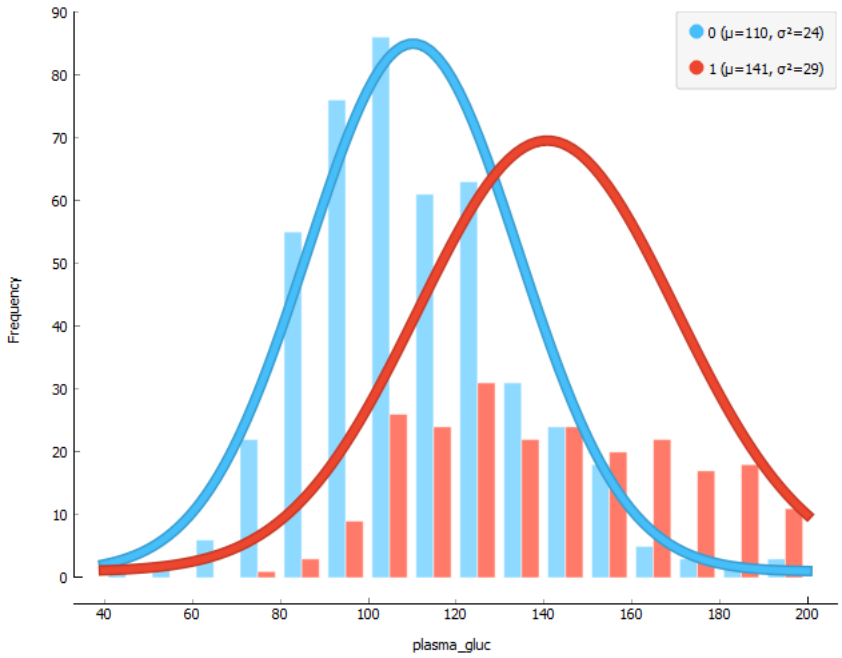
Features: plasma_gluc, age, bmi
Meta attributes: Mahalanobis
Target: diab_yn

The purpose of this report is to explore a dataset that measures a variety of features across Pima Indian women, ultimately scoring whether or not th
have diabetes.

In ranking features, the top 3 most important variables explaining the variance in the distribution are:
1. Plasma glucose(referring to blood glucose concentration at 2 hours in an oral glucose tolerance test);
2. Age;
3. BMI.

This dataset originally included n = 736 participants; after removing 77 outliers, I had 636 inliners.

Distributions Thu Jan 02 20, 09:45:36

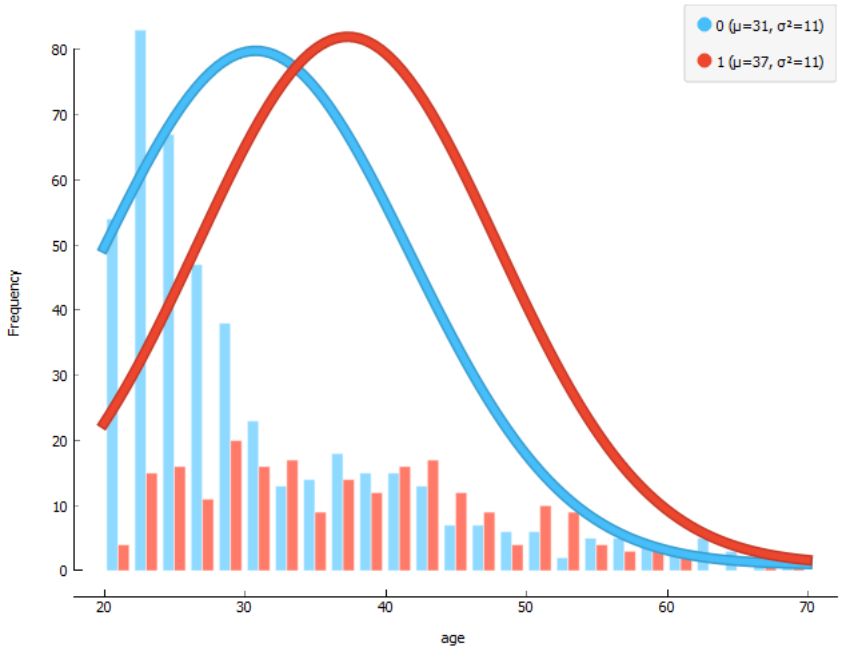


Distribution of 'plasma_gluc' with columns split by 'diab_yn'

Plasma glucose by Diabetes

This vis displays the number 1 ranked feature, blood plasma glucose, of diabetic vs non-diabetic women. As we can see, there are more non-diabetic s
in the dataset; furthermore, plasma glucose is higher for diabetic women (mean = 141, std = 29) than for non-diabetic women (mean = 110, std = 24).

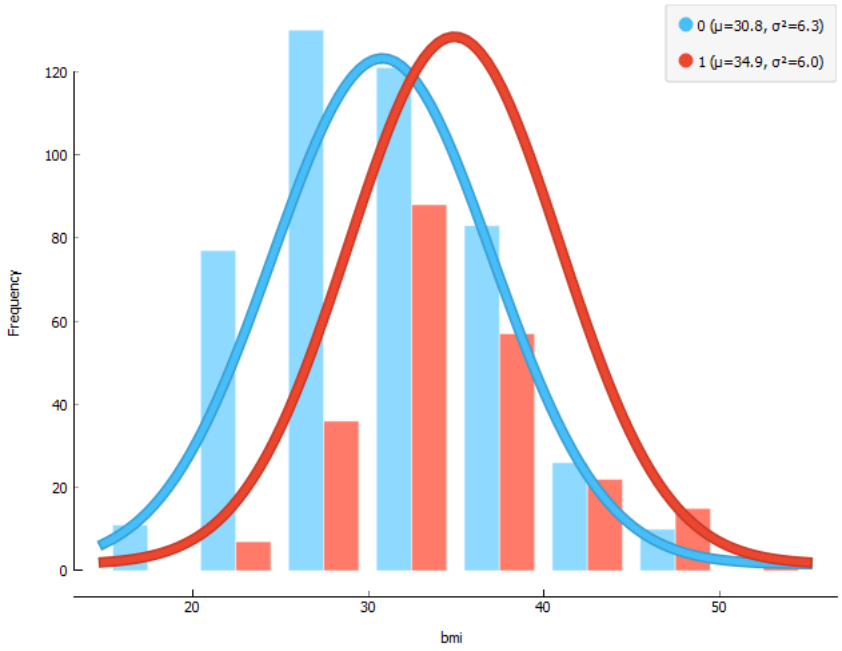
Distributions Thu Jan 02 20, 09:46:26



Distribution of 'age' with columns split by 'diab_yn'

Age is the second most important component of our dataset, as per feature selection. On average, non-diabetics tend to be younger (mean = 31, std = 11) than diabetics (mean age = 37, std = 11).

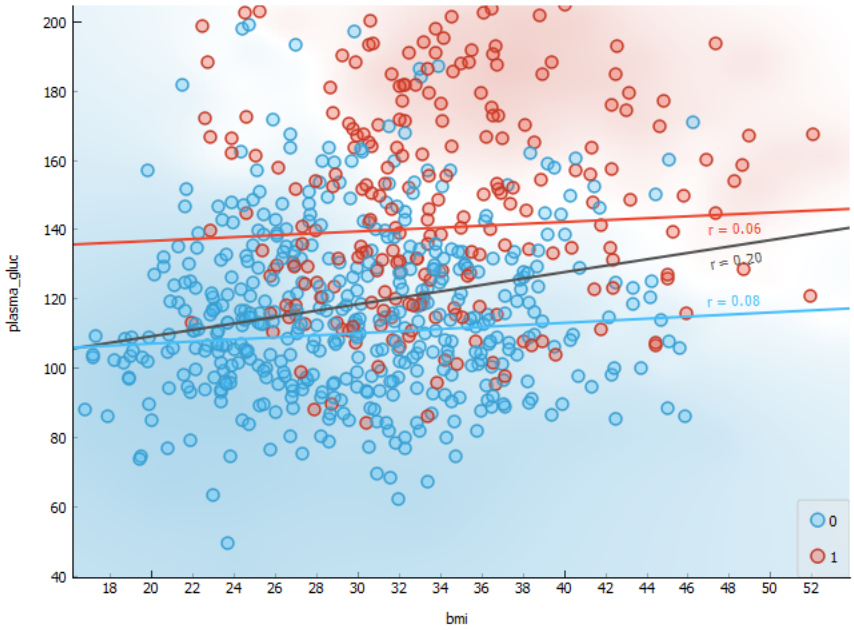
Distributions Thu Jan 02 20, 09:47:11



Distribution of 'bmi' with columns split by 'diab_yn'

BMI by Diabetes
The no diabetes group tends to have a lower BMI (mean = 30.8, std =6.3) than the group with diabetes (mean = 34.9, std = 6.0). This is the third most important feature in predicting diabetes.

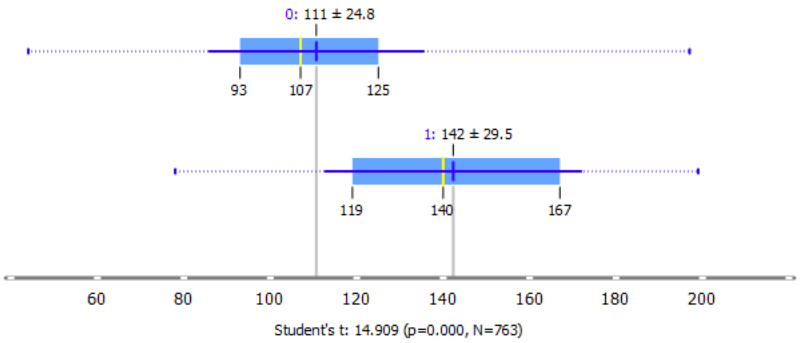
Scatter Plot Thu Jan 02 20, 09:48:29



Color: diab_yn, **Jittering**: 2 %

k-Means clustering. 2 clusters found: diabetics and non-diabetics.

Box Plot Thu Jan 02 20, 10:13:11



Box plot for attribute 'plasma_gluc' grouped by 'diab_yn'

Box plot of feature selection: blood glucose.

Test and Score Thu Jan 02 20, 11:32:55

Settings

Sampling type: Stratified 10-fold Cross validation
Target class: Average over classes

Scores

Model	AUC	CA	F1	Precision	Recall	Specificity
Neural Network	0.8322642622653211	0.7719528178243774	0.7692540554778845	0.7681814470821304	0.7719528178243774	0.7049612984522295
Logistic Regression	0.8375970106352402	0.7745740498034076	0.7673250584637523	0.7694333383103561	0.7745740498034076	0.6801543204179203
Naive Bayes	0.8172985280101662	0.7391874180865007	0.742908065134962	0.7508820294318259	0.7391874180865007	0.7223714085575743
Random Forest Classification	0.8023101012087563	0.744429882044561	0.7426220639446832	0.7414124322663475	0.744429882044561	0.6797467567354878
kNN	0.7485968442232342	0.7221494102228048	0.7175911440170252	0.7156910483422337	0.7221494102228048	0.6381174541211538
Tree	0.6030241599975794	0.7051114023591087	0.704450459546816	0.7038507714720926	0.7051114023591087	0.641229802766381
AdaBoost	0.6485491898760988	0.6775884665792923	0.6786687426209543	0.6798967220592813	0.6775884665792923	0.6195099131729052
SVM	0.7026973873315079	0.6657929226736566	0.6719305004593507	0.6858461667359493	0.6657929226736566	0.649890652461364

Next, I passed the entire dataset through the models above, using cross-validation in order to split my dataset into Training and Testing. The results are above.
Again, the Neural Network came on top: (F1 = 0.769), barely edging out Logistic Regression (0.767. Third was Naive Bayes (F1 = 0.74), last was of all them models was SVM (F1 = 0.67)

Confusion Matrix Thu Jan 02 20, 11:37:11

Confusion matrix for Neural Network (showing proportion of predicted)

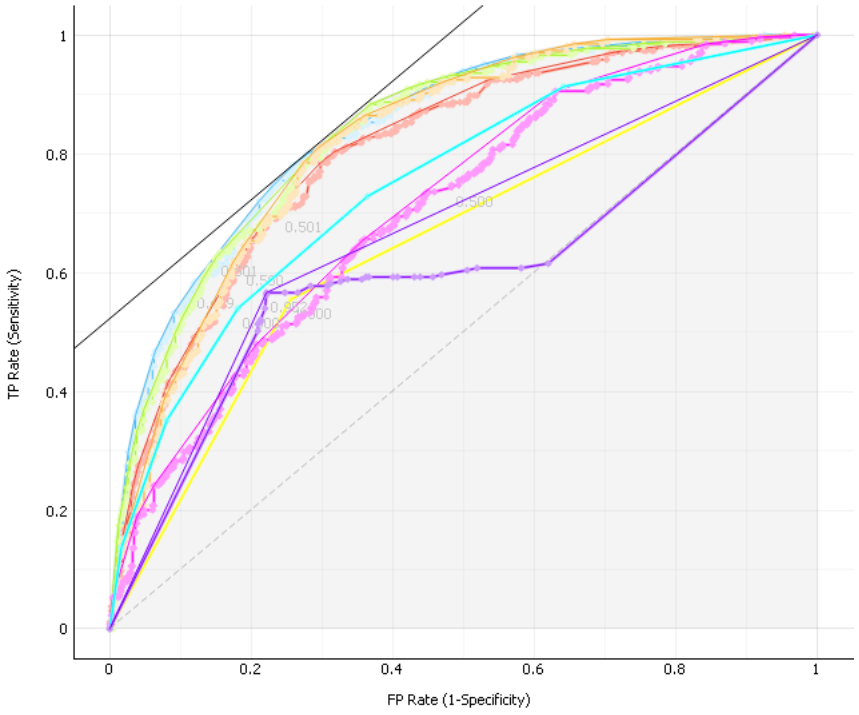
		Predicted		
		0	**1**	∑
Actual	**0**	81.0 %	31.0 %	**497**
	1	19.0 %	69.0 %	**266**
	∑	**521**	**242**	**763**

This is the Confusion Matrix for the top-performing algorithm, Neural Network, comparing the proportion of actual (diabetes = 1, no diabetes = 0) vs predicted (diabetes = 1, no diabetes = 0). I chose to show the proportion of predicted.
The confusion matrix shows us the relative proportions of true positives (I predicted true and it was actually true), True Negatives (I predicted False and it was actually False), False Positives (I predicted True and it was actually False), and False Negatives (I predicted false and it was
actually true). These are important to keep in mind, especially with regards to medical information. The cost of false positives and false negatives can be gravely important.
For the Neural Network:
True Positives (had diabetes, predicted diabetes): 69%
True Negatives (no diabetes, predicted no diabetes): 81%
False Positives (predicted diabetes, actually no diabetes): 31%
False Negatives (predicted no diabetes, had diabetes): 19%

When dealing with medical information, we want to keep the false positives low.

ROC Analysis Thu Jan 02 20, 11:45:28

Target class: 1
Costs: FP = 500, FN = 500
Target probability: 50.0 %



Logistic Regression Random Forest Classification Neural Network Naive Bayes AdaBoost SVM kNN Tree

ROC Analysis plots a true positive rate against a false positive rate of a test.

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system (0 or 1, diabetes or no diabetes) as its discrimination threshold is varied. The curve plots a false positive rate on an x-axis (1-specificity; probability that target=1 when true value=0) against a true positive rate on a y-axis (sensitivity; probability that target=1 when true value=1).

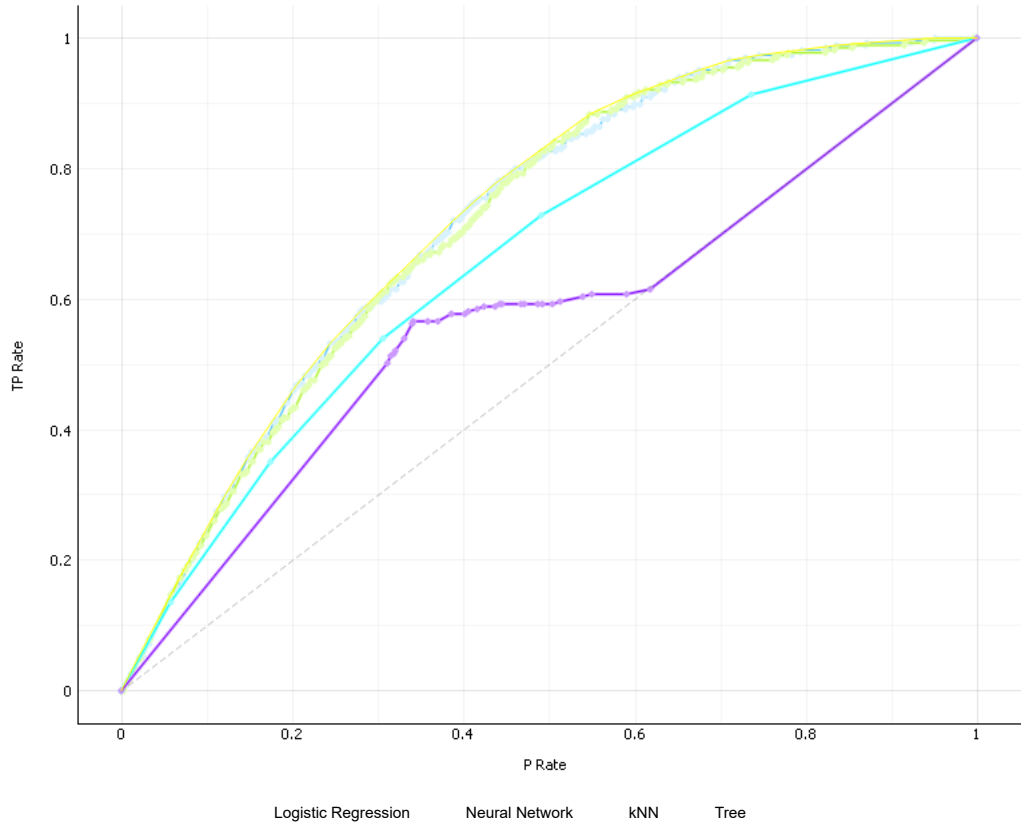
The diagonal dotted line represents the behavior of a random classifier.

In my case, the Neural Network and Logistic Regression are the most accurate models.

This compares the means across different classification models. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the classifier.

Lift Curve Thu Jan 02 20, 11:59:34

Target class: 1



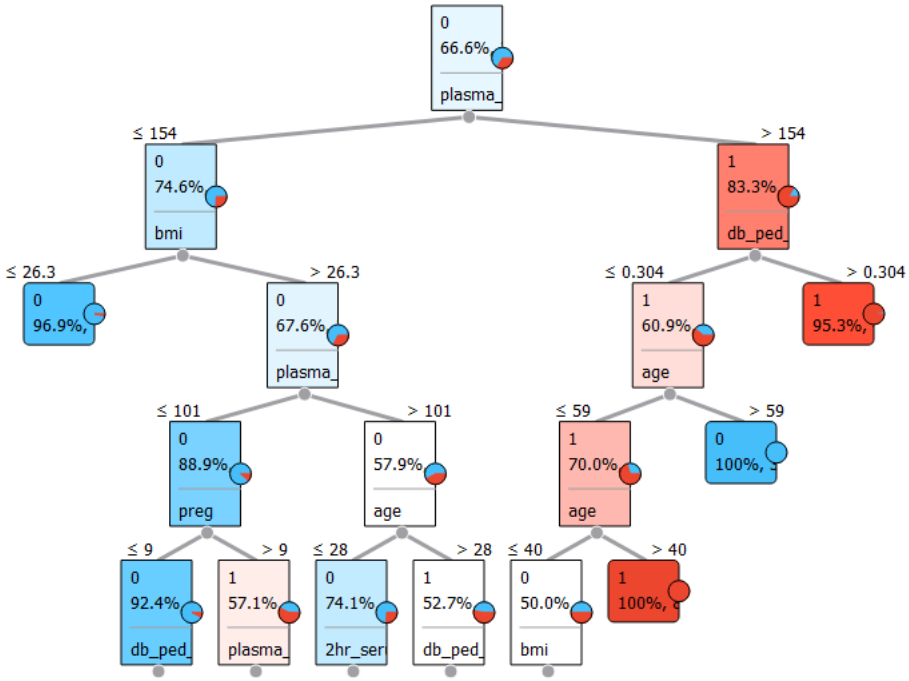
This is the Lift Curve on the Diabetes (1) variable, Positive Rate along the X-asis and True Positive Rate along the y-axis. The Lift Curve is another way to test the classification algorithms.

The Lift curve shows the relation between the number of instances which were predicted positive and those that are indeed positive and thus measures the performance of a chosen classifier against a random classifier. The graph is constructed with the cumulative number of cases (in descending order of probability) on the x-axis and the cumulative number of true positives on the y-axis.

For clarity, I chose to only include Neural Network (green), Logistic Regression (blue, virtually indistinguishable from the NN green line), kNN, and the Tree classification. To interpret the curve, the the perfect classifier would have a steep slope towards 1 until all classes are guessed correctly and then run straight along 1 on y-axis to (1,1). We can see the Neural Network and Logistic Regression are the best of the models.

Tree Viewer Thu Jan 02 20, 12:21:08

Tree size: 141 nodes, 71 leaves
Edge widths: Fixed
Target class: None



In order to make predictions, I manually split up my dataset into Training (70%) and Testing (30%). After removing outliers, I had 686 inliers in my Training dataset and 230 in my Test dataset.

Here, I made a Tree using the Training dataset, simplified for viewing in this report. I pruned to 5 layers deep. First branch is always blood plasma glucose.

Tree Classification was one of the models employed to make predictions.

Predictions Thu Jan 02 20, 12:28:07

Info

Data: 207 instances.
Predictors: 4
Task: Classification
Showing probabilities for: 0, 1

Data & Predictions

	Classification Tree (binary)	LogRegTrain	Naive Bayes	Neural Network: Training	diab_yn	Mahalanobis
1	0.00 : 1.00 → 1	0.61 : 0.39 → 0	0.41 : 0.59 → 1	0.68 : 0.32 → 0	0	5.64805
2	0.00 : 1.00 → 1	0.43 : 0.57 → 1	0.09 : 0.91 → 1	0.19 : 0.81 → 1	1	9.14019
3	0.96 : 0.04 → 0	0.58 : 0.42 → 0	0.05 : 0.95 → 1	0.41 : 0.59 → 1	1	9.01745
4	0.00 : 1.00 → 1	0.68 : 0.32 → 0	0.56 : 0.44 → 0	0.74 : 0.26 → 0	1	2.41915
5	0.00 : 1.00 → 1	0.58 : 0.42 → 0	0.41 : 0.59 → 1	0.80 : 0.20 → 0	1	13.3579
6	1.00 : 0.00 → 0	0.88 : 0.12 → 0	0.99 : 0.01 → 0	0.93 : 0.07 → 0	0	11.2241
7	0.96 : 0.04 → 0	0.93 : 0.07 → 0	0.97 : 0.03 → 0	0.93 : 0.07 → 0	0	3.75972
8	0.05 : 0.95 → 1	0.13 : 0.87 → 1	0.02 : 0.98 → 1	0.04 : 0.96 → 1	1	11.5225
9	0.05 : 0.95 → 1	0.03 : 0.97 → 1	0.04 : 0.96 → 1	0.02 : 0.98 → 1	1	21.5953
10	0.00 : 1.00 → 1	0.73 : 0.27 → 0	0.74 : 0.26 → 0	0.77 : 0.23 → 0	0	3.9876
11	0.05 : 0.95 → 1	0.43 : 0.57 → 1	0.46 : 0.54 → 1	0.22 : 0.78 → 1	0	19.4572
12	0.05 : 0.95 → 1	0.17 : 0.83 → 1	0.10 : 0.90 → 1	0.22 : 0.78 → 1	0	24.2208
13	1.00 : 0.00 → 0	0.90 : 0.10 → 0	0.94 : 0.06 → 0	0.93 : 0.07 → 0	0	5.36848
14	0.96 : 0.04 → 0	0.90 : 0.10 → 0	0.98 : 0.02 → 0	0.93 : 0.07 → 0	0	2.82257
15	1.00 : 0.00 → 0	0.78 : 0.22 → 0	0.49 : 0.51 → 1	0.88 : 0.12 → 0	0	22.3934
16	0.96 : 0.04 → 0	0.94 : 0.06 → 0	1.00 : 0.00 → 0	0.95 : 0.05 → 0	0	2.21071
17	0.96 : 0.04 → 0	0.90 : 0.10 → 0	0.99 : 0.01 → 0	0.91 : 0.09 → 0	0	4.40796
18	0.97 : 0.03 → 0	0.83 : 0.17 → 0	0.36 : 0.64 → 1	0.83 : 0.17 → 0	0	8.52263
19	1.00 : 0.00 → 0	0.86 : 0.14 → 0	0.86 : 0.14 → 0	0.90 : 0.10 → 0	0	6.81493
20	0.00 : 1.00 → 1	0.78 : 0.22 → 0	0.39 : 0.61 → 1	0.91 : 0.09 → 0	0	10.5079
21	0.00 : 1.00 → 1	0.47 : 0.53 → 1	0.27 : 0.73 → 1	0.79 : 0.21 → 0	0	19.5837
22	1.00 : 0.00 → 0	0.84 : 0.16 → 0	0.63 : 0.37 → 0	0.79 : 0.21 → 0	0	13.4446
23	1.00 : 0.00 → 0	0.65 : 0.35 → 0	0.33 : 0.67 → 1	0.67 : 0.33 → 0	1	9.63321
24	0.05 : 0.95 → 1	0.12 : 0.88 → 1	0.34 : 0.66 → 1	0.10 : 0.90 → 1	1	17.4462
25	0.96 : 0.04 → 0	0.89 : 0.11 → 0	0.99 : 0.01 → 0	0.92 : 0.08 → 0	0	3.25741
26	0.96 : 0.04 → 0	0.86 : 0.14 → 0	0.99 : 0.01 → 0	0.81 : 0.19 → 0	0	2.85889
27	0.96 : 0.04 → 0	0.89 : 0.11 → 0	0.92 : 0.08 → 0	0.92 : 0.08 → 0	0	8.21055
28	0.97 : 0.03 → 0	0.90 : 0.10 → 0	1.00 : 0.00 → 0	0.97 : 0.03 → 0	0	6.07334
29	0.96 : 0.04 → 0	0.82 : 0.18 → 0	0.98 : 0.02 → 0	0.90 : 0.10 → 0	0	3.97426
30	1.00 : 0.00 → 0	0.90 : 0.10 → 0	0.90 : 0.10 → 0	0.90 : 0.10 → 0	0	8.50718
31	0.00 : 1.00 → 1	0.50 : 0.50 → 0	0.20 : 0.80 → 1	0.31 : 0.69 → 1	0	2.92336
32	1.00 : 0.00 → 0	0.83 : 0.17 → 0	0.42 : 0.58 → 1	0.70 : 0.30 → 0	1	5.54014
33	0.96 : 0.04 → 0	0.93 : 0.07 → 0	0.84 : 0.16 → 0	0.82 : 0.18 → 0	0	8.02047
34	0.97 : 0.03 → 0	0.87 : 0.13 → 0	0.95 : 0.05 → 0	0.99 : 0.01 → 0	0	13.1785
35	1.00 : 0.00 → 0	0.85 : 0.15 → 0	0.99 : 0.01 → 0	0.80 : 0.20 → 0	0	3.99503
36	1.00 : 0.00 → 0	0.90 : 0.10 → 0	1.00 : 0.00 → 0	0.95 : 0.05 → 0	0	4.59992
37	1.00 : 0.00 → 0	0.58 : 0.42 → 0	0.40 : 0.60 → 1	0.82 : 0.18 → 0	0	15.5165
38	1.00 : 0.00 → 0	0.79 : 0.21 → 0	0.95 : 0.05 → 0	0.80 : 0.20 → 0	0	13.1259
39	0.97 : 0.03 → 0	0.80 : 0.20 → 0	0.92 : 0.08 → 0	0.78 : 0.22 → 0	0	12.3028
40	1.00 : 0.00 → 0	0.48 : 0.52 → 1	0.66 : 0.34 → 0	0.68 : 0.32 → 0	1	13.2559
41	1.00 : 0.00 → 0	0.62 : 0.38 → 0	0.61 : 0.39 → 0	0.55 : 0.45 → 0	0	8.24962
42	1.00 : 0.00 → 0	0.39 : 0.61 → 1	0.47 : 0.53 → 1	0.51 : 0.49 → 0	1	14.9899
43	0.97 : 0.03 → 0	0.89 : 0.11 → 0	0.97 : 0.03 → 0	0.84 : 0.16 → 0	0	6.65424
44	1.00 : 0.00 → 0	0.63 : 0.37 → 0	0.73 : 0.27 → 0	0.95 : 0.05 → 0	0	11.1251
45	1.00 : 0.00 → 0	0.70 : 0.30 → 0	0.28 : 0.72 → 1	0.62 : 0.38 → 0	0	9.97704
46	0.97 : 0.03 → 0	0.96 : 0.04 → 0	1.00 : 0.00 → 0	0.98 : 0.02 → 0	0	4.95718
47	0.00 : 1.00 → 1	0.44 : 0.56 → 1	0.10 : 0.90 → 1	0.37 : 0.63 → 1	1	7.83717
48	0.97 : 0.03 → 0	0.90 : 0.10 → 0	0.84 : 0.16 → 0	0.91 : 0.09 → 0	0	4.155
49	0.05 : 0.95 → 1	0.11 : 0.89 → 1	0.06 : 0.94 → 1	0.03 : 0.97 → 1	1	23.0738
50	0.00 : 1.00 → 1	0.16 : 0.84 → 1	0.08 : 0.92 → 1	0.25 : 0.75 → 1	1	18.209
51	1.00 : 0.00 → 0	0.79 : 0.21 → 0	0.80 : 0.20 → 0	0.81 : 0.19 → 0	0	7.11535
52	1.00 : 0.00 → 0	0.58 : 0.42 → 0	0.35 : 0.65 → 1	0.30 : 0.70 → 1	1	6.93626
53	0.00 : 1.00 → 1	0.56 : 0.44 → 0	0.07 : 0.93 → 1	0.25 : 0.75 → 1	0	5.79226
54	0.05 : 0.95 → 1	0.24 : 0.76 → 1	0.53 : 0.47 → 0	0.37 : 0.63 → 1	1	12.6302
55	0.96 : 0.04 → 0	0.93 : 0.07 → 0	1.00 : 0.00 → 0	0.96 : 0.04 → 0	0	18.0777
56	1.00 : 0.00 → 0	0.35 : 0.65 → 1	0.25 : 0.75 → 1	0.14 : 0.86 → 1	1	12.9745
57	0.97 : 0.03 → 0	0.93 : 0.07 → 0	0.99 : 0.01 → 0	0.96 : 0.04 → 0	0	9.66423
58	1.00 : 0.00 → 0	0.90 : 0.10 → 0	0.98 : 0.02 → 0	0.96 : 0.04 → 0	0	6.66494
59	1.00 : 0.00 → 0	0.88 : 0.12 → 0	0.54 : 0.46 → 0	0.82 : 0.18 → 0	0	9.04024

60	1.00 : 0.00 → 0	0.24 : 0.76 → 1	0.05 : 0.95 → 1	0.21 : 0.79 → 1	1	7.45215
61	1.00 : 0.00 → 0	0.68 : 0.32 → 0	0.92 : 0.08 → 0	0.68 : 0.32 → 0	0	7.16757
62	0.05 : 0.95 → 1	0.08 : 0.92 → 1	0.25 : 0.75 → 1	0.05 : 0.95 → 1	1	21.1865
63	0.97 : 0.03 → 0	0.97 : 0.03 → 0	1.00 : 0.00 → 0	0.98 : 0.02 → 0	0	8.31533
64	1.00 : 0.00 → 0	0.47 : 0.53 → 1	0.15 : 0.85 → 1	0.47 : 0.53 → 1	0	10.5187
65	0.97 : 0.03 → 0	0.95 : 0.05 → 0	0.99 : 0.01 → 0	0.99 : 0.01 → 0	0	6.98838
66	1.00 : 0.00 → 0	0.88 : 0.12 → 0	0.93 : 0.07 → 0	0.94 : 0.06 → 0	0	4.2115
67	0.05 : 0.95 → 1	0.26 : 0.74 → 1	0.42 : 0.58 → 1	0.17 : 0.83 → 1	1	10.3089
68	0.05 : 0.95 → 1	0.13 : 0.87 → 1	0.01 : 0.99 → 1	0.03 : 0.97 → 1	1	12.7077
69	0.00 : 1.00 → 1	0.64 : 0.36 → 0	0.51 : 0.49 → 0	0.53 : 0.47 → 0	0	9.28812
70	1.00 : 0.00 → 0	0.26 : 0.74 → 1	0.19 : 0.81 → 1	0.21 : 0.79 → 1	1	7.65885
71	0.97 : 0.03 → 0	0.92 : 0.08 → 0	0.96 : 0.04 → 0	0.93 : 0.07 → 0	0	3.12293
72	0.00 : 1.00 → 1	0.81 : 0.19 → 0	0.37 : 0.63 → 1	0.92 : 0.08 → 0	0	10.2741
73	0.97 : 0.03 → 0	0.99 : 0.01 → 0	1.00 : 0.00 → 0	1.00 : 0.00 → 0	0	7.3212
74	1.00 : 0.00 → 0	0.41 : 0.59 → 1	0.20 : 0.80 → 1	0.51 : 0.49 → 0	1	22.0441
75	1.00 : 0.00 → 0	0.78 : 0.22 → 0	0.56 : 0.44 → 0	0.78 : 0.22 → 0	0	5.89024
76	0.96 : 0.04 → 0	0.83 : 0.17 → 0	0.98 : 0.02 → 0	0.93 : 0.07 → 0	0	7.48795
77	1.00 : 0.00 → 0	0.89 : 0.11 → 0	0.97 : 0.03 → 0	0.90 : 0.10 → 0	0	4.22658
78	0.96 : 0.04 → 0	0.84 : 0.16 → 0	0.89 : 0.11 → 0	0.88 : 0.12 → 0	0	7.78263
79	0.97 : 0.03 → 0	0.90 : 0.10 → 0	0.98 : 0.02 → 0	0.94 : 0.06 → 0	0	5.01293
80	0.00 : 1.00 → 1	0.74 : 0.26 → 0	0.83 : 0.17 → 0	0.78 : 0.22 → 0	0	5.58883
81	1.00 : 0.00 → 0	0.64 : 0.36 → 0	0.28 : 0.72 → 1	0.46 : 0.54 → 1	0	6.05079
82	0.97 : 0.03 → 0	0.95 : 0.05 → 0	1.00 : 0.00 → 0	0.97 : 0.03 → 0	0	5.7379
83	1.00 : 0.00 → 0	0.68 : 0.32 → 0	0.33 : 0.67 → 1	0.62 : 0.38 → 0	1	7.36744
84	1.00 : 0.00 → 0	0.90 : 0.10 → 0	0.82 : 0.18 → 0	0.92 : 0.08 → 0	0	4.90366
85	0.97 : 0.03 → 0	0.90 : 0.10 → 0	0.99 : 0.01 → 0	0.91 : 0.09 → 0	0	3.35329
86	1.00 : 0.00 → 0	0.89 : 0.11 → 0	0.92 : 0.08 → 0	0.97 : 0.03 → 0	0	8.38434
87	0.97 : 0.03 → 0	0.90 : 0.10 → 0	0.94 : 0.06 → 0	0.84 : 0.16 → 0	0	11.5324
88	1.00 : 0.00 → 0	0.61 : 0.39 → 0	0.31 : 0.69 → 1	0.53 : 0.47 → 0	1	16.2315
89	0.00 : 1.00 → 1	0.87 : 0.13 → 0	0.56 : 0.44 → 0	0.79 : 0.21 → 0	0	7.37906
90	0.96 : 0.04 → 0	0.87 : 0.13 → 0	1.00 : 0.00 → 0	0.95 : 0.05 → 0	0	3.21302
91	0.96 : 0.04 → 0	0.52 : 0.48 → 0	0.48 : 0.52 → 1	0.45 : 0.55 → 1	1	8.60816
92	0.97 : 0.03 → 0	0.97 : 0.03 → 0	1.00 : 0.00 → 0	0.99 : 0.01 → 0	0	8.11644
93	1.00 : 0.00 → 0	0.88 : 0.12 → 0	0.89 : 0.11 → 0	0.93 : 0.07 → 0	0	7.58033
94	0.00 : 1.00 → 1	0.66 : 0.34 → 0	0.55 : 0.45 → 0	0.58 : 0.42 → 0	0	5.36817
95	1.00 : 0.00 → 0	0.55 : 0.45 → 0	0.17 : 0.83 → 1	0.57 : 0.43 → 0	1	6.77298
96	1.00 : 0.00 → 0	0.85 : 0.15 → 0	0.64 : 0.36 → 0	0.89 : 0.11 → 0	0	4.19691
97	1.00 : 0.00 → 0	0.50 : 0.50 → 0	0.10 : 0.90 → 1	0.61 : 0.39 → 0	0	27.4936
98	0.05 : 0.95 → 1	0.61 : 0.39 → 0	0.82 : 0.18 → 0	0.61 : 0.39 → 0	1	9.42788
99	0.05 : 0.95 → 1	0.26 : 0.74 → 1	0.53 : 0.47 → 0	0.25 : 0.75 → 1	1	17.0807
100	1.00 : 0.00 → 0	0.55 : 0.45 → 0	0.14 : 0.86 → 1	0.48 : 0.52 → 1	1	9.91121
	+ 7 more					

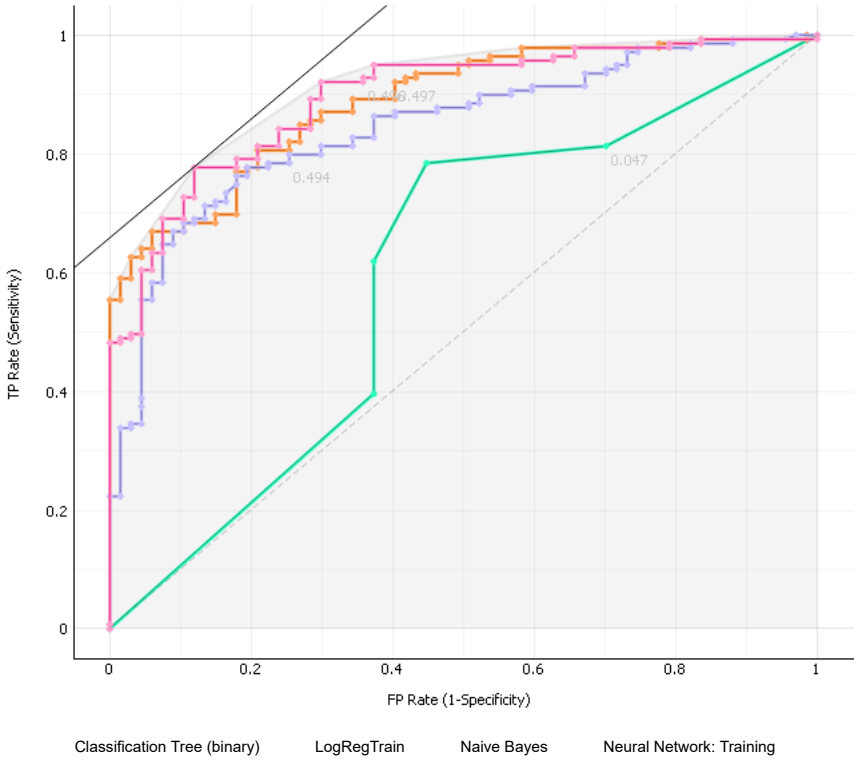
To make actual predictions for whether a person would have diabetes given their features (blood glucose, age, BMI, etc..), I manually split up the dataset into a Training (70%) and Testing (30%), used the former to train my dataset and then make predictions on the Test dataset. Each model gives the probability of a person having diabetes or no diabetes, given their inputs.
Above, you can compare each model's prediction across all of the Testing dataset; whether or not the person actually had diabetes is denoted by the 'diab_yn' col: 0 = no diabetes, 1 = diabetes.

To judge the models, I was looking for the highest F1 as the indicator for the best model; F1 is a weighted average of the accuracy and precision of a model. The results of 4 machine learning models were:
1. Neural Network: F1 = 0.83, which is rather good
2. Logistic Regression: F1 = 0.81
3. Naive Bayes: F1 = 0.78
4. Classification Tree, as seen above: F1 = 0.71.

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. They help us cluster and clarify, and are good for deciding "Thing" or "Not a Thing", as in our case here (diabetes/no diabetes). You can thank neural networks for filtering your email (Spam or Not Spam).

ROC Analysis Thu Jan 02 20, 12:33:04

Target class: 0
Costs: FP = 500, FN = 500
Target probability: 50 %



This is the ROC analysis from the prediction when I split up my dataset manually. I included 3 model classificaitons here: Classification Tree (green), Logistic Regression (orange), and Neural Network (pink). Again, the neural network and logistic regression are far and away the best of the 3, and are very close together in their accuracy.