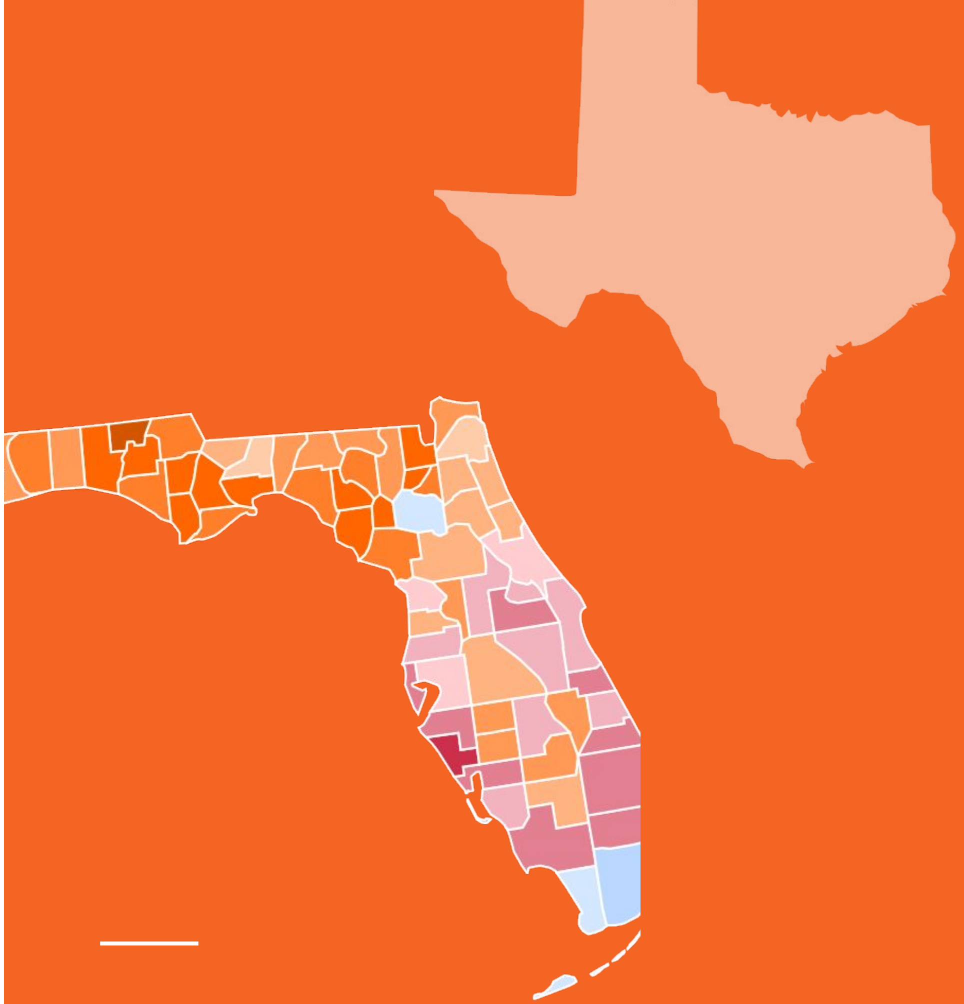
Determining at risk Pima Indian diabetic patients using data mining techniques

Rana Putta
MIS637
Professor Mahmoud Daneshmand

One in ten U.S. adults has diabetes now, according to the American Diabetes Association. In 2018, 34.8 million Americans had diabetes, of which 7.3 million were undiagnosed.

THAT'S MORE THAN THE POPULATION OF FLORIDA AND TEXAS COMBINED



Source: travel.trade.gov

Introduction

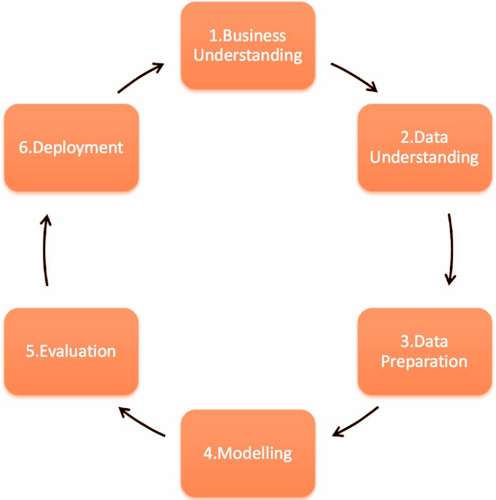
Diabetes is known to be one of the leading causes of death and oftentimes there is a steep cost associated with treatment post diagnosis.

In this project the objective of this project is to predict whether or not a patient has diabetes.

CRISP-DM

In this project I will be using the CRISP-DM approach because it is a proven and robust methodology.

The CRISP-DM (Cross Industry Standard Process for Data-Mining) methodology provides a structured approach to planning a data mining project.



CRISP-DM

1. Business Understanding

What are the desired outputs of the project? Access the current situation and determine the data mining goals.

2. Data Understanding

Describe data, explore data, verify data quality and data quality report.

3. Data Preparation

Select data, clean data, construct required data and integrate the data.

CRISP-DM

4. Modeling

Select modeling technique, generate test design, build and assess model.

5. Evaluation

Evaluate results, review process and determine next steps.

6. Deployment

Plan deployment, plan monitoring and maintenance, produce final report and review project.

Business Understanding

WHAT are we trying to solve?
Predict if a patient is suffering from a diabetic disease?

HOW are we trying to solve?
By using data mining techniques that use classification algorithms to classify is a patient has been suffering from diabetes or not and to derive rules for this.

WHY are we trying to solve?
The derived rules from the would be of interest to both the patients and the doctors. This would ease the doctors job in identifying patients suffering from diabetes It is of interest to the patients because early diagnosis of diabetes will help them be more conscious of their health choices , thereby reducing their post diagnosis expenses and increase the patient’s survival rate.

Data Understanding

Data Source

The data set was obtained from Kaggle. However, this dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases.

The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset.

Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Data Understanding

Data Set

The dataset contains 768 entries of Pima Indian patients and has a total of 9 attributes.

The dataset contains 768 patients, out of which 268 are 1 i.e diabetic and 500 are 0 i.e nondiabetic.

Out of the 9 attributes, there are 8 medical predictor (independent) variables and 1 target (dependent) variable, Outcome.

The target variable classifies if the patient is diabetic or not.

Data Understanding

Sample Dataset

```
display(data.head(20))
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1
10	4	110	92	0	0	37.6	0.191	30	0
11	10	168	74	0	0	38.0	0.537	34	1
12	10	139	80	0	0	27.1	1.441	57	0
13	1	189	60	23	846	30.1	0.398	59	1
14	5	166	72	19	175	25.8	0.587	51	1
15	7	100	0	0	0	30.0	0.484	32	1
16	0	118	84	47	230	45.8	0.551	31	1
17	7	107	74	0	0	29.6	0.254	31	1
18	1	103	30	38	83	43.3	0.183	33	0
19	1	115	70	30	96	34.6	0.529	32	1

Data Understanding

Dataset Info

```
display(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB

None
```

Data Understanding

Attribute details (Independent)

1
Pregnancies:
Number of times pregnant
[continuous]

2
Glucose:
Plasma glucose concentration a 2 hours in an oral glucose tolerance test
[continuous]

3
Blood Pressure:
Diastolic blood pressure
(mm Hg)
[continuous]

4
Skin Thickness:
Triceps skin fold thickness
(mm)
[continuous]

Data Understanding

Attribute details (Independent)

5
Insulin
2-Hour serum insulin
(mu U/ml)
[continuous]

6
BMI
Body mass index
Weight in kg
(height in m)^2)
[continuous]

7
Diabetes Pedigree
Function
[continuous]

8
Age
Age of patient
(Years)
[continuous]

Data Understanding

Attribute details (Dependent)

9
Outcome
Classify patients as diabetic or nondiabetic
(1 for diabetic or 0 for otherwise)
[discrete]

Data Preparation

The raw dataset needs to be cleaned and prepared for analysis.

We will examine the raw dataset for the following:

1. Missing Values
2. Outliers

Data Preparation

1. Missing Values

The dataset does not seem to have any missing values. All attributes have a non-null datafield count. Hence we move on to check for outliers.

```
display(data.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB

None
```

Data Preparation

2. Outliers

There seems to be either outliers or errors in several attributes of the data set. For example: In the Insulin attribute there are several rows which have a value of 0. It is not humanly possible to have an insulin level 0. We check for all the attributes (Glucose, Blood Pressure, Skin Thickness, Insulin and BMI) which have a value of 0 and replace it with NaN.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	NaN	33.6	0.627	50	1
1	1	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	8	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

Data Preparation

2. Outliers Continued

Now there are a total of 6 attributes which have been replaced with NaN for 0. These 6 attributes are numeric fields, so we replace the missing values with the mean of the rest of the non missing values associated with it's target value. If a row in 'Glucose' has NaN we replace it with 107 if the target is 0 (Non-Diabetic) and with 140 if 1 (Diabetic). We repeat this for all 6 attributes.

```
def mean_target(var):
    temp = data[data[var].notnull()]
    temp = temp[[var, 'Outcome']].groupby(['Outcome'])[[var]].median().reset_index()
    return temp
```

```
print(mean_target('Glucose'))
data.loc[(data['Outcome'] == 0 ) & (data['Glucose'].isnull()), 'Glucose'] = 107.0
data.loc[(data['Outcome'] == 1 ) & (data['Glucose'].isnull()), 'Glucose'] = 140
```

```
   Outcome  Glucose
0        0    107.0
1        1    140.0
```

Data Preparation

2. Outliers Continued (Before vs After)

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	NaN	33.6	0.627	50	1
1	1	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	8	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	169.5	33.6	0.627	50	1
1	1	85.0	66.0	29.0	102.5	26.6	0.351	31	0
2	8	183.0	64.0	32.0	169.5	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

Modeling

- We will use a Machine Learning Algorithm to model the dataset.
- ML algorithms are suitable in our case because rules can be derived from the data by splitting our dataset into training and testing.
- The derived rules can be applied to find relationships between dependent and independent variables.

Modeling

- There are several different ML techniques to choose from.
- The main goal in this project is to classify lima patients into diabetic or nondiabetic.
- Recursive Partitioning seems like a good fit here because the data needs to be split and broken down into smaller and smaller subsets resulting in a decision tree.
- The target variable in our project is categorical (1- diabetic and 0 - nondiabetic) so Classification Trees is appropriate.
- We will use CART Algorithm available on MiniTab to solve this problem.

Modeling

The cleaned dataset is export to a csv file and then opened using MiniTab.

We look at the Binary Response Information to double check if we have imported all rows.

CART Classification can be found in Stat -> Predictive Analysis -> CART Classification.

Binary Response Information

Variable	Class	Count	%
Outcome	1 (Event)	267	34.8
	0	500	65.2
	All	767	100.0

Minitab - Untitled

File Edit Data Calc Stat Graph View Help Assistant Additional Tools

	C1	C2	C3	C4	C5	C6	C7	C8
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabtesPedigr	Age
1	1	85	66.0	29	102.5	26.6	0.351	31
2	8	183	64.0	32	169.5	23.3	0.672	32
3	1	89	66.0	23	94.0	28.1	0.167	21
4	0	137	40.0	35	168.0	43.1	2.288	33
5	5	116	74.0	27	102.5	25.6	0.201	30
6	3	78	50.0	32	88.0	31.0	0.248	26
7	10	115	70.0	27	102.5	35.3	0.134	29
8	2	197	70.0	45	543.0	30.5	0.158	53
9	8	125	96.0	32	169.5	34.3	0.232	54
10	4	110	92.0	27	102.5	37.6	0.191	30
11	10	168	74.0	32	169.5	38.0	0.537	34
12	10	139	80.0	27	102.5	27.1	1.441	57
13	1	189	60.0	23	846.0	30.1	0.398	59
14	5	166	72.0	19	175.0	25.8	0.587	51
15	7	100	74.5	32	169.5	30.0	0.484	32
16	0	118	84.0	47	230.0	45.8	0.551	31
17	7	107	74.0	32	169.5	29.6	0.254	31
18	1	103	30.0	38	83.0	43.3	0.183	33
19	1	115	70.0	30	96.0	34.6	0.529	32
20	3	126	88.0	41	235.0	39.3	0.704	27
21	8	99	84.0	27	102.5	35.4	0.388	50
22	7	196	90.0	32	169.5	39.8	0.451	41
23	9	119	80.0	35	169.5	29.0	0.263	29
24	11	143	94.0	33	146.0	36.6	0.254	51
25	10	125	70.0	26	115.0	31.1	0.205	41
26	7	147	76.0	32	169.5	39.4	0.257	43
27	1	97	66.0	15	140.0	23.2	0.487	22
28	13	145	82.0	19	110.0	22.2	0.245	57
29	5	117	92.0	27	102.5	34.1	0.337	38
30	5	109	75.0	26	102.5	36.0	0.546	60
31	3	158	76.0	36	245.0	31.6	0.851	28
32	3	88	58.0	11	54.0	24.8	0.267	22
33	6	92	92.0	27	102.5	19.9	0.188	28
34	10	122	78.0	31	102.5	27.6	0.512	45
35	4	103	60.0	33	192.0	24.0	0.966	33
36	11	138	76.0	27	102.5	33.2	0.420	35
37	9	102	76.0	37	169.5	32.9	0.665	46
38	2	90	68.0	42	169.5	38.2	0.503	27
39	4	111	72.0	47	207.0	37.1	1.390	56
40	3	180	64.0	25	70.0	34.0	0.271	26

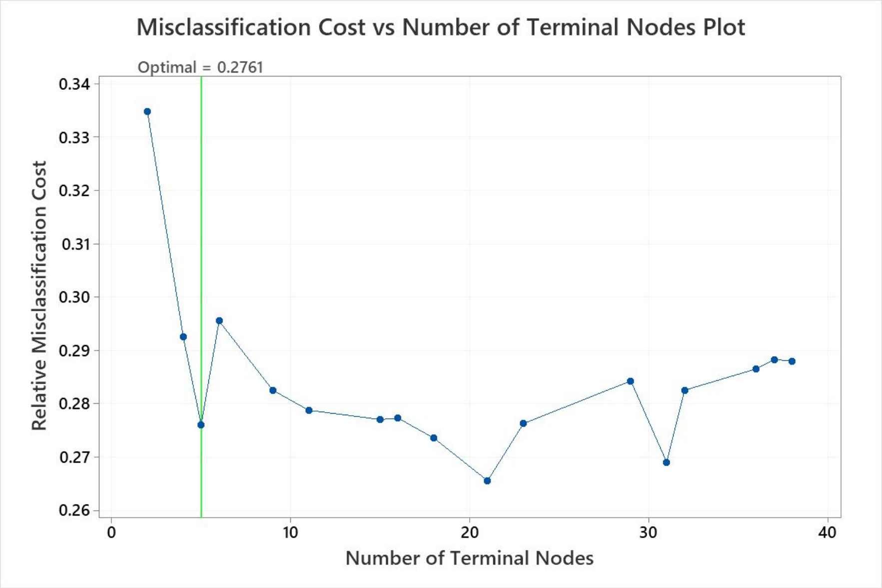
data.csv

data.csv

Modeling

The algorithm automatically calculates the optimal value for the Number of Terminal Nodes vs Relative Misclassification Cost.

The Optimal value used was 0.2761 for relative misclassification cost with 5 terminal nodes.



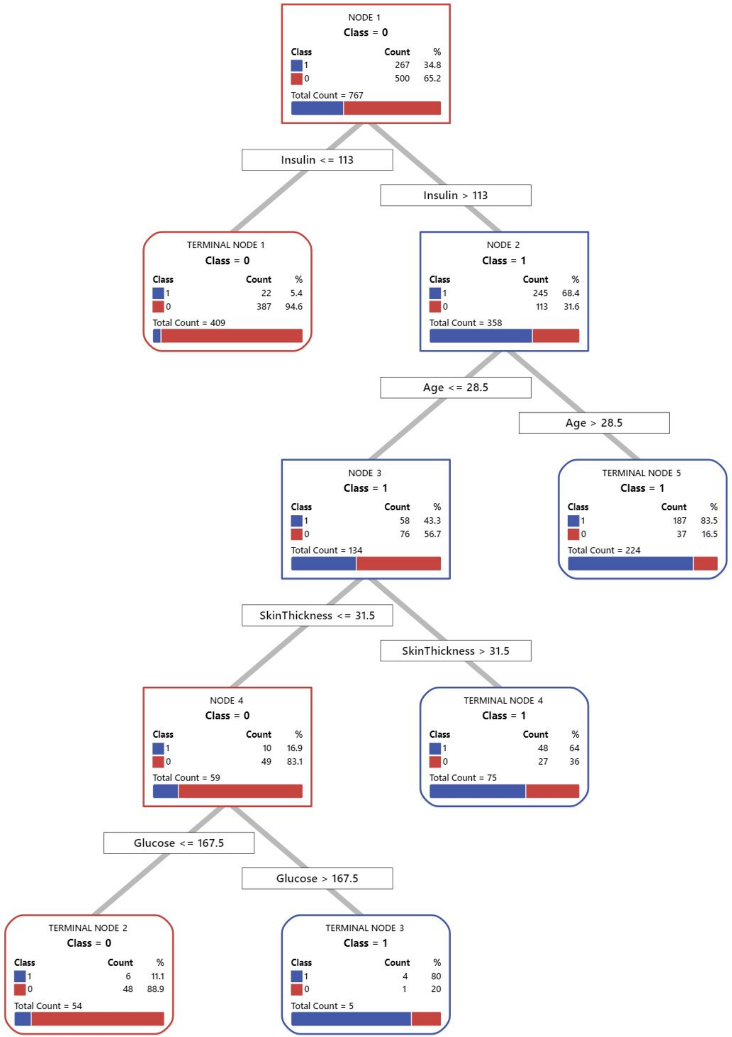
Modeling

CART displays the optimal tree that it has found by comparing the relative cost and number of nodes.

The optimal tree has 5 terminal nodes and shows the decision rules.

The first split occurs for the insulin attribute which. Terminal Node 1 classifies 409 patients which is more than half the data set and has 387 of all the 500 nondiabetic patients.

Second split for Age, third for Skin Thickness



Evaluation

Minitab also produces the model summary. The model categorized all the 8 independent variables as important predictors.

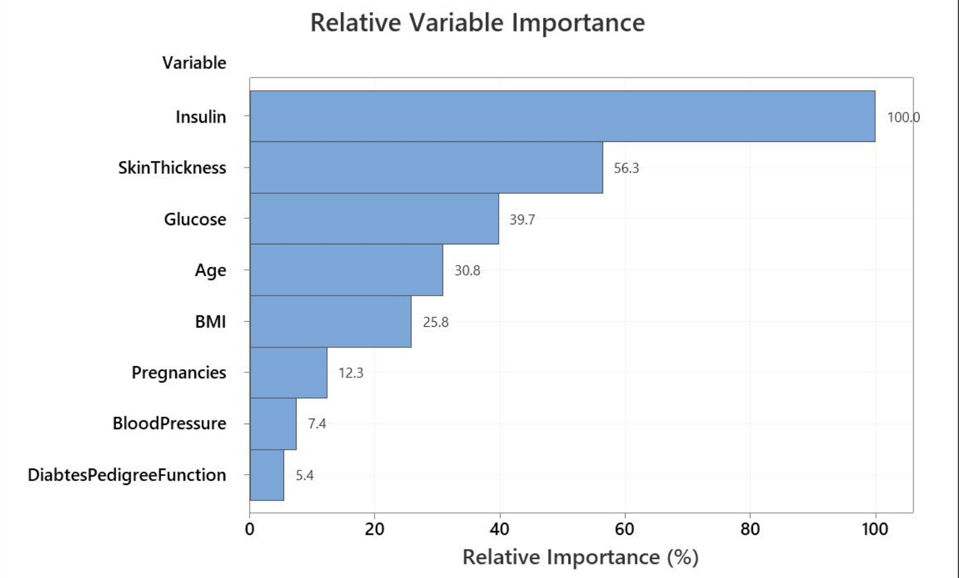
Insulin has the highest relative importance of 100% compared to the rest of the attributes. This makes sense because insulin is the first step to predict if a patient has diabetes or not.

Diabetes Pedigree Function and Blood Pressure have the least relative importance.

Model Summary

Total predictors	8
Important predictors	8
Number of terminal nodes	5
Minimum terminal node size	5

Statistics	Training	Test
Deviance R-Squared	0.4827	0.2624
Average -LogLikelihood	0.3343	0.4767
Area under ROC curve	0.8999	0.8745
95% CI	(0.4422, 1)	(0.8458, 0.9031)
Lift	2.3982	2.1847
Misclassification cost	0.2349	0.2761



Variable importance measures model improvement when splits are made on a predictor.
Relative importance is defined as % improvement with respect to the top predictor.

Evaluation

The Misclassification Table displays the % Error for Testing Set. There seems to be a higher percent error that were wrongly classified as diabetic.

The confusion Matrix which displays the statistics for errors. The false negative (type II error) is higher than False positive rate (type I error).

Misclassification

Input Misclassification Cost	Predicted Class	
Actual Class	1	0
1		1.00
0	1.00	

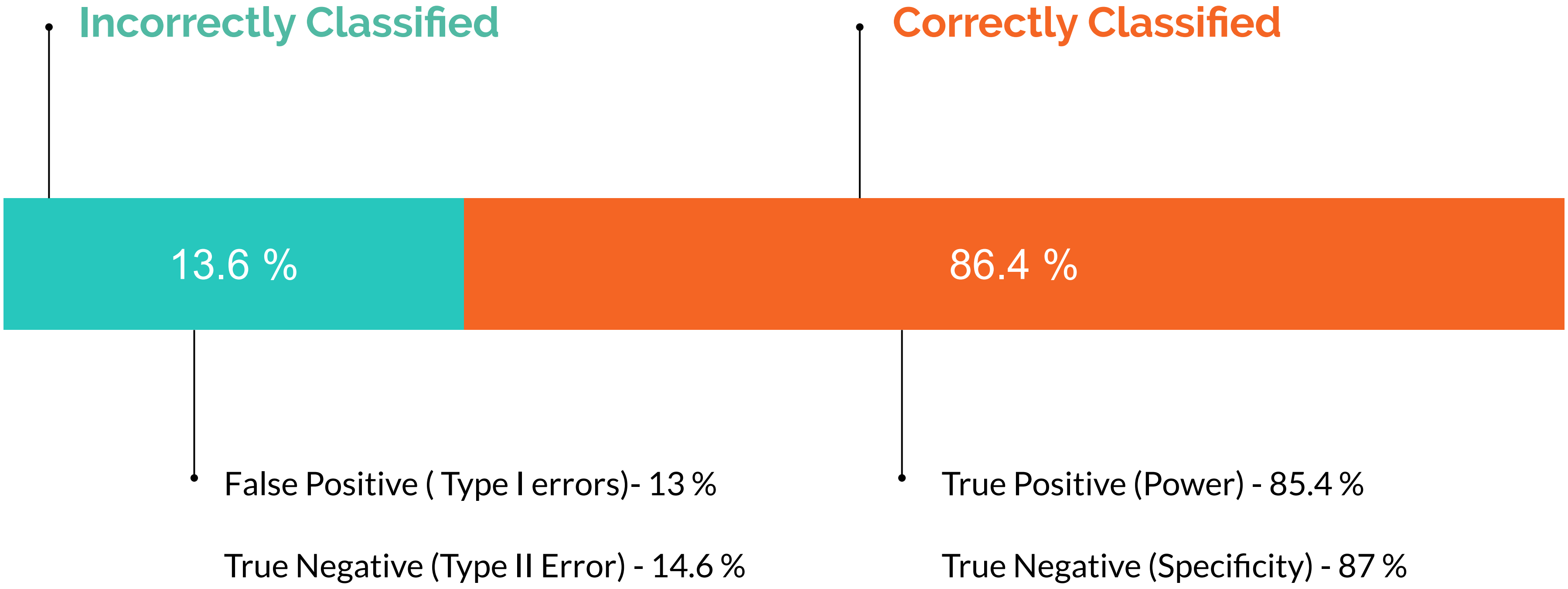
		Training			Test		
Actual Class	Count	Misclassed	% Error	Cost	Misclassed	% Error	Cost
1 (Event)	267	28	10.5	0.1049	39	14.6	0.1461
0	500	65	13.0	0.1300	65	13.0	0.1300
All	767	93	12.1	0.1174	104	13.6	0.1380

Confusion Matrix

		Predicted Class (Training)			Predicted Class (Test)		
Actual Class	Count	1	0	%Correct	1	0	%Correct
1 (Event)	267	239	28	89.5	228	39	85.4
0	500	65	435	87.0	65	435	87.0
All	767	304	463	87.9	293	474	86.4

Statistics	Training (%)	Test (%)
True positive rate (sensitivity or power)	89.5	85.4
False positive rate (type I error)	13.0	13.0
False negative rate (type II error)	10.5	14.6
True negative rate (specificity)	87.0	87.0

Prediction Accuracy on Test Set

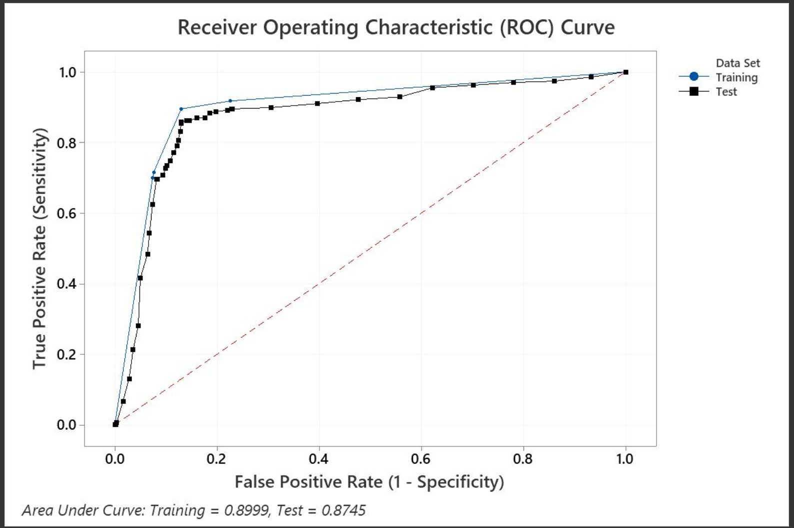


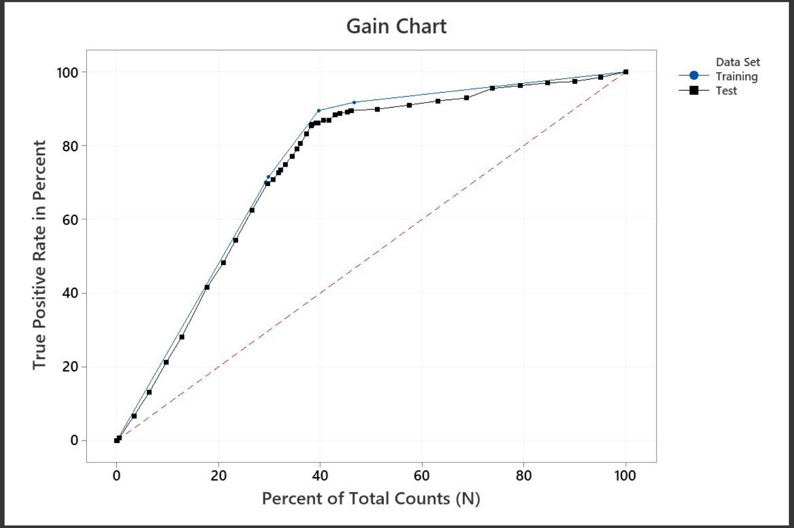
Evaluation

The figures show the ROC curve and the Gain Chart.

The false positive rate increase sharply after a true positive rate crosses 0.8

The Gain Chart shows the percent of total counts vs the true positive rate. In a way this graph is the reciprocal function of the above graph. We notice that the true positive rate climbly to 0.8 before 40% of total counts.

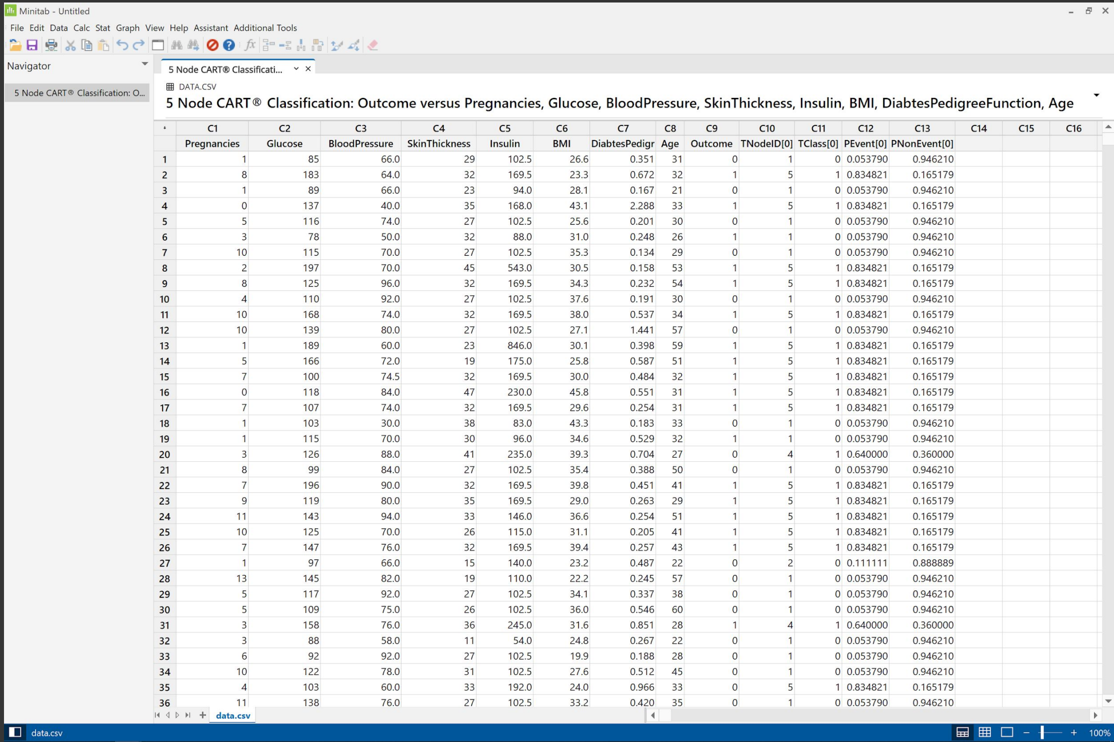




Evaluation

MiniTab also provides the option for the user to check how how it classifies each patients into classes.

The PEvent[0] attribute is the probability of that patient classified as diabetic and the other PNonEvent[0] field displays the probability for that patient to be classified as non diabetic.



5 Node CART® Classification: Outcome versus Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabtesPedigreeFunction, Age

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabtesPedigr	Age	Outcome	TNodeID[0]	TClass[0]	PEvent[0]	PNonEvent[0]
1	1	85	66.0	29	102.5	26.6	0.351	31	0	1	0	0.053790	0.946210
2	8	183	64.0	32	169.5	23.3	0.672	32	1	5	1	0.834821	0.165179
3	1	89	66.0	23	94.0	28.1	0.167	21	0	1	0	0.053790	0.946210
4	0	137	40.0	35	168.0	43.1	2.288	33	1	5	1	0.834821	0.165179
5	5	116	74.0	27	102.5	25.6	0.201	30	0	1	0	0.053790	0.946210
6	3	78	50.0	32	88.0	31.0	0.248	26	1	1	0	0.053790	0.946210
7	10	115	70.0	27	102.5	35.3	0.134	29	0	1	0	0.053790	0.946210
8	2	197	70.0	45	543.0	30.5	0.158	53	1	5	1	0.834821	0.165179
9	8	125	96.0	32	169.5	34.3	0.232	54	1	5	1	0.834821	0.165179
10	4	110	92.0	27	102.5	37.6	0.191	30	0	1	0	0.053790	0.946210
11	10	168	74.0	32	169.5	38.0	0.537	34	1	5	1	0.834821	0.165179
12	10	139	80.0	27	102.5	27.1	1.441	57	0	1	0	0.053790	0.946210
13	1	189	60.0	23	846.0	30.1	0.398	59	1	5	1	0.834821	0.165179
14	5	166	72.0	19	175.0	25.8	0.587	51	1	5	1	0.834821	0.165179
15	7	100	74.5	32	169.5	30.0	0.484	32	1	5	1	0.834821	0.165179
16	0	118	84.0	47	230.0	45.8	0.551	31	1	5	1	0.834821	0.165179
17	7	107	74.0	32	169.5	29.6	0.254	31	1	5	1	0.834821	0.165179
18	1	103	30.0	38	83.0	43.3	0.183	33	0	1	0	0.053790	0.946210
19	1	115	70.0	30	96.0	34.6	0.529	32	1	1	0	0.053790	0.946210
20	3	126	88.0	41	235.0	39.3	0.704	27	0	4	1	0.640000	0.360000
21	8	99	84.0	27	102.5	35.4	0.388	50	0	1	0	0.053790	0.946210
22	7	196	90.0	32	169.5	39.8	0.451	41	1	5	1	0.834821	0.165179
23	9	119	80.0	35	169.5	29.0	0.263	29	1	5	1	0.834821	0.165179
24	11	143	94.0	33	146.0	36.6	0.254	51	1	5	1	0.834821	0.165179
25	10	125	70.0	26	115.0	31.1	0.205	41	1	5	1	0.834821	0.165179
26	7	147	76.0	32	169.5	39.4	0.257	43	1	5	1	0.834821	0.165179
27	1	97	66.0	15	140.0	23.2	0.487	22	0	2	0	0.111111	0.888889
28	13	145	82.0	19	110.0	22.2	0.245	57	0	1	0	0.053790	0.946210
29	5	117	92.0	27	102.5	34.1	0.337	38	0	1	0	0.053790	0.946210
30	5	109	75.0	26	102.5	36.0	0.546	60	0	1	0	0.053790	0.946210
31	3	158	76.0	36	245.0	31.6	0.851	28	1	4	1	0.640000	0.360000
32	3	88	58.0	11	54.0	24.8	0.267	22	0	1	0	0.053790	0.946210
33	6	92	92.0	27	102.5	19.9	0.188	28	0	1	0	0.053790	0.946210
34	10	122	78.0	31	102.5	27.6	0.512	45	0	1	0	0.053790	0.946210
35	4	103	60.0	33	192.0	24.0	0.966	33	0	5	1	0.834821	0.165179
36	11	138	76.0	27	102.5	33.2	0.420	35	0	1	0	0.053790	0.946210

Deployment

The is the last phase of the CRISP-DM process This part of the cycle is responsible for a smooth transfer of the application to the consumer.
This phase will include how our model performs on a small set of data to ensure success of model. Reports on the performance of our model is also provided. Here we once again check if we have met our goals and objectives first set out during the Business Understanding phase.
The consumer is also informed about the instructions and maintenance of the product.
Finally, this model should help doctors in diagnosing and identifying patients who are diabetic. This model has the potential to increase the survival chances as well as reduce medical expenses by predicting if someone is diabetic or not.

Conclusion

The model has a prediction accuracy of 86.4 %. It can correctly classify is a patient is diabetic or not 86.4 % of the times.

The model can also be used for predicting if a patient is diabetic or not without the insulin measure because there are other attributes that also play an important role in classifying patients into diabetic or non diabetic.

We can apply other machine learning algorithms to further improve the model and also try testing the model on a wider pool of patients.