

# Advanced databases and data warehouses

## Project: Data analysis with Bayesian network models

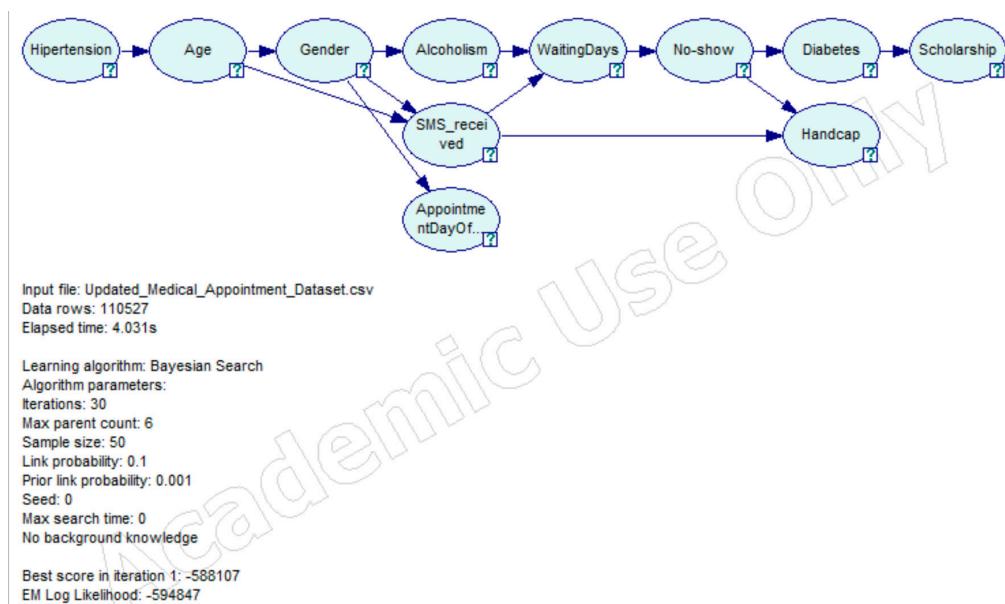
### 1. Brief Description of the Analyzed Data Set

The dataset analyzed in this project is the **Medical Appointment No Shows Dataset**, which contains information about medical appointments scheduled in Vitória, Brazil. The primary objective of this dataset is to **predict whether a patient will attend their scheduled medical appointment** based on various demographic and health-related attributes.

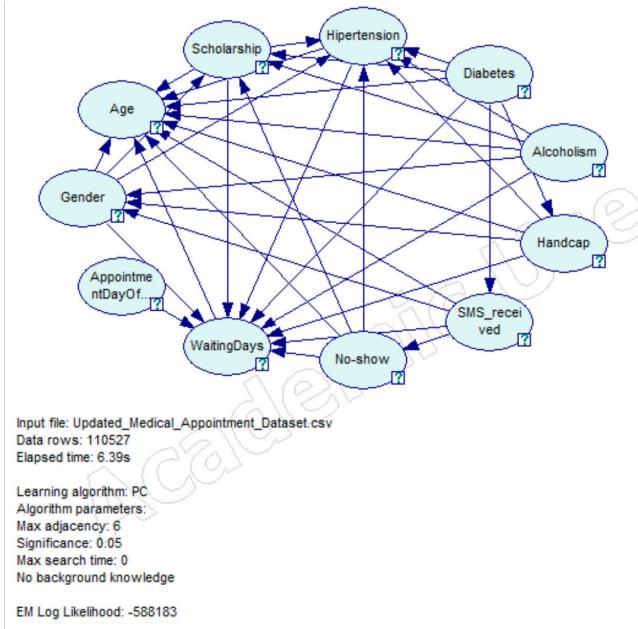
The dataset includes the following variables:

- **Gender:** The patient's gender (Female or Male).
- **ScheduledDay:** The date on which the appointment was scheduled.
- **AppointmentDay:** The date on which the appointment is supposed to occur.
- **Age:** The patient's age.
- **Neighbourhood:** The neighborhood where the patient resides.
- **Scholarship:** Indicates whether the patient is enrolled in a welfare program (1 = Yes, 0 = No).
- **Hypertension:** Presence of hypertension (1 = Yes, 0 = No).
- **Diabetes:** Presence of diabetes (1 = Yes, 0 = No).
- **Alcoholism:** Presence of alcoholism (1 = Yes, 0 = No).
- **Handcap:** The level of disability the patient has (ranging from 0 to 4).
- **SMS\_received:** Indicates whether the patient received a reminder SMS (1 = Yes, 0 = No).
- **No-show:** The target variable indicating whether the patient showed up for their appointment (No = showed up, Yes = did not show up).

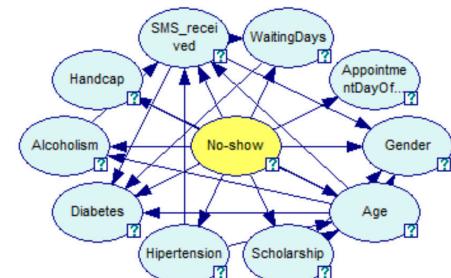
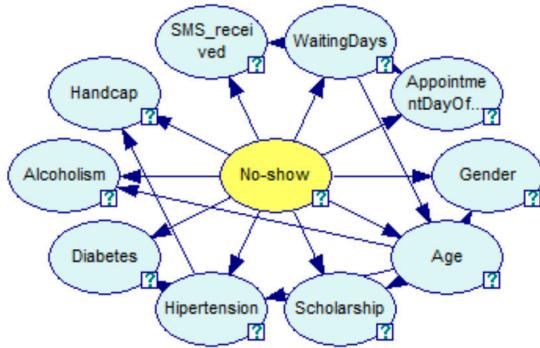
### 2. A causal structures of the Bayesian network model



Bayesian Search

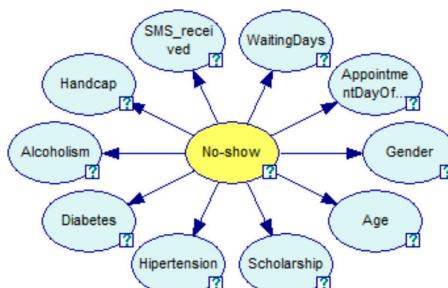


PC Algorithm



TAN

ANB



Naive Bayes

### 3. Basic statistics of the created models (number of nodes, arcs, average number of parents per node, etc.)

Model	Number of Nodes	Number of Arcs	Average Number of Parents per Node	Maximum Number of Parents (Indegree)
Bayesian Search	11	10	0.91	6
PC Algorithm	11	34	3.091	9
Augmented Naive Bayes (ANB)	11	25	2.273	5
Naive Bayes	11	10	0.9091	1
Tree-Augmented Naive Bayes (TAN)	11	19	1.727	2

The table above summarizes the basic structural properties of the models developed during the study. Each model consists of 11 nodes, representing the selected variables. The **PC Algorithm** resulted in the densest network with 34 arcs and the highest average number of parents per node (3.091), indicating a highly interconnected structure. In contrast, the **Naive Bayes** model created the sparsest network with only 10 arcs, reflecting its assumption of conditional independence between features given the class variable. The **Tree-Augmented Naive Bayes (TAN)** and **Augmented Naive Bayes (ANB)** models demonstrated intermediate complexity, balancing connectivity and simplicity. **Bayesian Search** produced a relatively sparse model similar to Naive Bayes but allowed for some additional dependencies. Overall, the PC Algorithm model captured more complex relationships, while Naive-based models prioritized simplicity and interpretability.

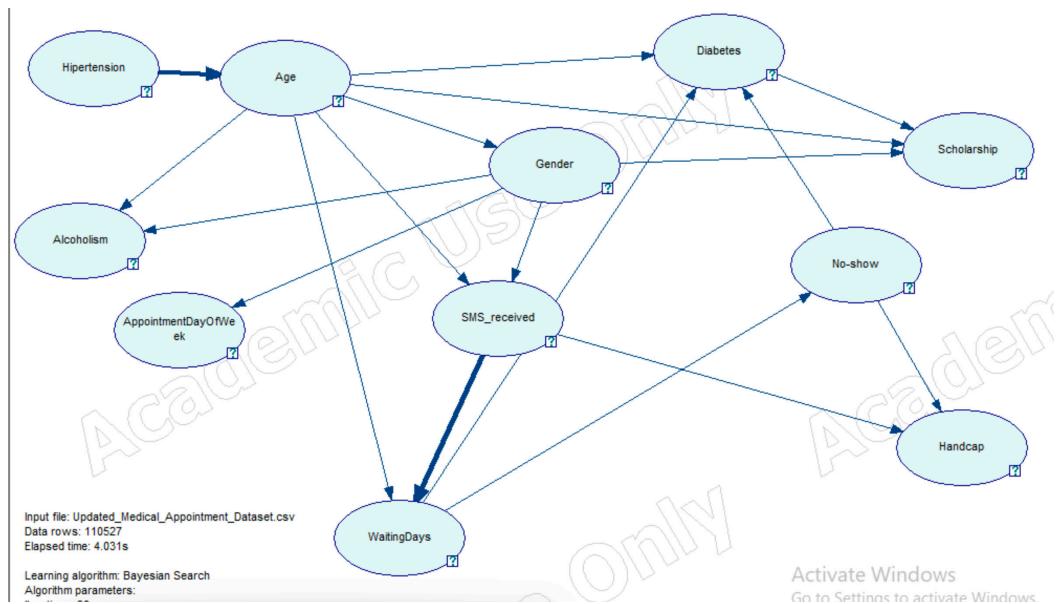
### 4. Details of discretization of continuous variables

The discretization process was applied to continuous and ordinal variables to optimize the performance of the Bayesian network models by transforming them into categorical variables. The details are as follows:

- **Age:** Divided into four categories:
  - s1\_below\_18 (0–17 years)
  - s2\_18\_38 (18–37 years)
  - s3\_38\_55 (38–54 years)
  - s4\_55\_up (55 years and above)
- **Handcap:** Binarized into a single category:
  - s2\_0\_1, representing No Disability (0) vs Disability (1 or higher)
- **WaitingDays:** Divided into four categories based on the waiting time:
  - s1\_below\_1 (Same-day or next-day appointments)
  - s2\_1\_4 (1–3 days)
  - s3\_4\_8 (4–7 days)
  - s4\_8\_up (8 or more days)
- **AppointmentDayOfWeek:** Simplified into two categories:
  - Weekday (Monday to Friday)
  - Weekend (Saturday and Sunday)

This discretization strategy ensured that all input variables were fully categorical, which is essential for efficient structure learning and parameter estimation in Bayesian network models.

## 5. List of model arcs with the strongest strength of influence and their interpretation for selected causal model



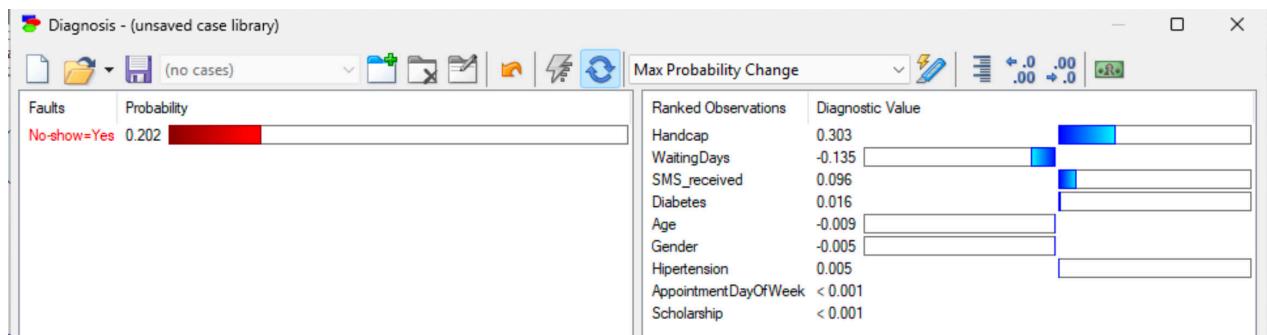
The table below summarizes the model arcs ranked by their average strength of influence, as estimated by the **Bayesian Search** model. The influence values provide a measure of how strongly a parent variable affects its child variable in the causal network.

Parent	Child	Average Influence
Age	Gender	0.112432
Age	Scholarship	0.0553821
Age	Diabetes	0.113117
WaitingDays	No-show	0.117853
Age	WaitingDays	0.0531005
No-show	Diabetes	0.0117708
No-show	Handcap	0.00148153
SMS_received	Handcap	0.00759923
SMS_received	WaitingDays	0.511565
WaitingDays	Diabetes	0.0155524

- **SMS\_received → WaitingDays** had the strongest influence (0.511565), indicating that receiving an SMS reminder significantly affects the patient's waiting time until the appointment.
- **WaitingDays → No-show** shows a strong direct relationship (0.117853), suggesting that the length of waiting time has a notable impact on whether a patient will miss the appointment.
- **Age → Diabetes** and **Age → Gender** have relatively strong influences, reflecting the demographic dependencies between patient age and these attributes.
- **No-show → Diabetes** and **No-show → Handcap** represent feedback effects where missing an appointment may be associated with certain medical conditions.

Overall, variables such as **SMS reminders** and **waiting time** play critical roles in influencing patient behavior regarding appointment attendance, as captured by the structure of the Bayesian network.

## 6. List of variables that have the highest diagnostic value to differentiate the class variable and their interpretation for selected causal model



The following variables were identified as having the highest diagnostic value in differentiating the class variable No-show:

Variable	Diagnostic Value	Interpretation
Handcap	303	Patients with disabilities are more likely to miss their appointments.
WaitingDays	-135	Longer waiting times reduce the likelihood of a patient missing their appointment.
SMS_received	96	Receiving an SMS reminder slightly increases the probability of a no-show, contrary to expectations.
Diabetes	16	Patients with diabetes are marginally more likely to miss their appointments.
Age	-9	Older patients are slightly less likely to miss their appointments.
Gender	-5	Female patients (assuming Gender encoding) are slightly less likely to miss appointments.
Hipertension	5	Hypertension slightly increases the likelihood of a no-show.
AppointmentDayOf Week	<0.001	Minimal effect on no-show probability.
Scholarship	<0.001	Minimal effect on no-show probability.

**Positive diagnostic values** indicate that the variable increases the probability of a no-show (patient does not attend the appointment). Conversely, **negative diagnostic values** suggest that the variable reduces the probability of a no-show, thus increasing the likelihood of attendance.

For instance, Handcap has the highest positive diagnostic value, meaning that patients with disabilities are significantly more prone to no-shows. Interestingly, WaitingDays has a negative diagnostic value, implying that longer waiting times actually decrease the probability of missing the appointment, a counterintuitive finding. Similarly, older age (Age) is associated with a slight decrease in no-show probability.

**7. The model diagnostic accuracy based on the cross-validation method (specify the method used). Check what the diagnostic quality of the model would be if the naive Bayes model, the TAN model, or the ANB model were used instead of the causal structure models.**

Model Performance Comparison

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)
Naive Bayes	79.82	2.02	99.12
ANB	79.2	0.28	99.92
PC	79.66	1.55	99.43
Bayesian Search	79.8	0.0	100.0
TAN	79.8	0.0448	99.9932

Accuracy is around 79-80% for all models.

This indicates that the models are good at correctly predicting the majority class, which is the patients who show up for appointments.

Sensitivity is very low (especially for Bayesian Search and ANB), meaning:

- The models have difficulty identifying the minority class (patients who do not show up).
- This is a class imbalance problem: in the dataset, the number of patients who show up greatly exceeds those who do not.
- As a result, models tend to predict the dominant class (No-show = No), leading to high accuracy but low sensitivity.

Specificity is very high (>99%) across all models:

- Models are very good at predicting patients who will show up.
- This again shows that the models favor the majority class.

Why Accuracy is High but Sensitivity is Low

- Accuracy is not a good metric when dealing with imbalanced datasets.
- Because the number of "No-show" = Yes is very low compared to "No-show" = No, the model can achieve high accuracy simply by predicting most patients will show up.
- However, this leads to very poor Sensitivity because the model fails to capture the few "no-show" cases.

Thus, Sensitivity and Specificity must be reported along with Accuracy to get a true picture of the model's diagnostic ability.

### Validation Methodology

In this study, I evaluated the diagnostic performance of the models using the k-fold cross-validation method, with  $k = 10$ .

Cross-validation is a method used to assess how well a model generalizes to an independent dataset.

Instead of splitting the data into just one training and one test set, the data is divided into  $k$  equal parts (folds). The model is trained on  $k-1$  folds and tested on the remaining fold. This process is repeated  $k$  times, and the results are averaged to provide a more reliable estimate of model performance.

I specifically used 10-fold cross-validation, which is a common choice because:

- It provides a good balance between the amount of training and test data.
- It reduces the chance of the results depending on how the data is split.
- It is considered a standard practice in model evaluation due to its balance between reliability and computation time.

In 10-fold cross-validation:

- 90% of the data is used for training.
- 10% of the data is used for testing.
- This process is repeated 10 times, and the results are averaged.

#### Models Compared

I compared the performance of:

- The causal structure model
- Naive Bayes (NB)
- Tree Augmented Naive Bayes (TAN)
- Augmented Naive Bayes (ANB)

The models were evaluated based on:

- Accuracy: How often the model makes correct predictions.
- Sensitivity: How well the model identifies patients who will not show up.
- Specificity: How well the model identifies patients who will show up.

#### 8) Confusion matrix for three models (causal model, naive Bayes model, TAN model, ANB model)

		Predicted	
		No	Yes
Act.	No	87441	767
	Yes	21860	459

Naive Bayes

		Predicted	
		No	Yes
Act.	No	87721	487
	Yes	21995	324

PC Algorithm

		Predicted	
		No	Yes
Act.	No	88207	1
	Yes	22319	0

Bayesian Search

		Predicted	
		No	Yes
Act.	No	88134	74
	Yes	22256	63

ANB

		Predicted	
		No	Yes
Act.	No	88202	6
	Yes	22309	10

TAN

#### Confusion Matrix Explanation

The confusion matrices above summarize the classification results for each model.

- True Positives (TP): Correctly predicted "No-show = Yes" cases (bottom right cell).
- True Negatives (TN): Correctly predicted "No-show = No" cases (top left cell).
- False Positives (FP): Incorrectly predicted as "Yes" when it was actually "No" (top right cell).
- False Negatives (FN): Incorrectly predicted as "No" when it was actually "Yes" (bottom left cell).

Based on these values:

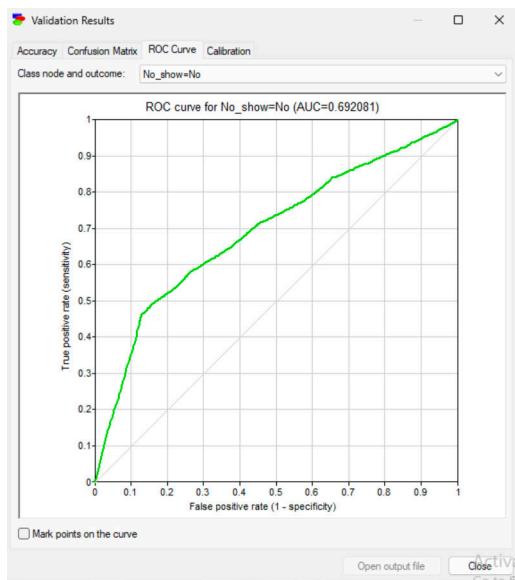
- Accuracy measures the proportion of all correct predictions among total cases.
- Sensitivity (Recall) measures how well the model identifies patients who do not show up (Yes).
- Specificity measures how well the model identifies patients who do show up (No).

The confusion matrices show that the models have high specificity but low sensitivity. This means the models are generally better at predicting patients who attend their appointments than those who miss them. This behavior is mainly due to the class imbalance in the dataset, where most patients actually show up.

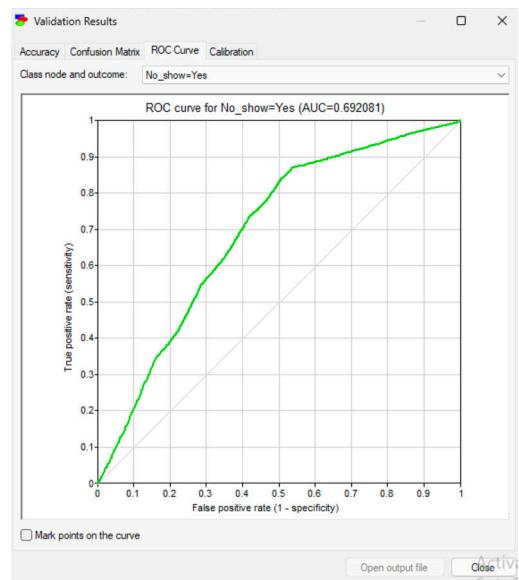
## 9) ROC curve for the models

The ROC (Receiver Operating Characteristic) curve is a widely used graphical representation to evaluate the performance of a binary classification model. It plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various threshold levels. An AUC value ranges from 0.5 to 1.0, where 0.5 indicates no discriminatory ability (random guessing) and 1.0 represents perfect classification. The ROC curve provides a visual insight into the model's performance independent of class distribution or threshold selection, making it a robust measure for model evaluation.

### ROC Curve Analysis for Bayesian Search Model



No Show = No

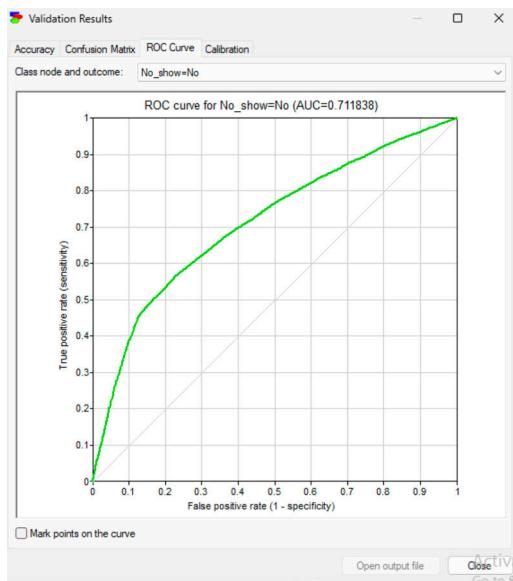


No Show = Yes

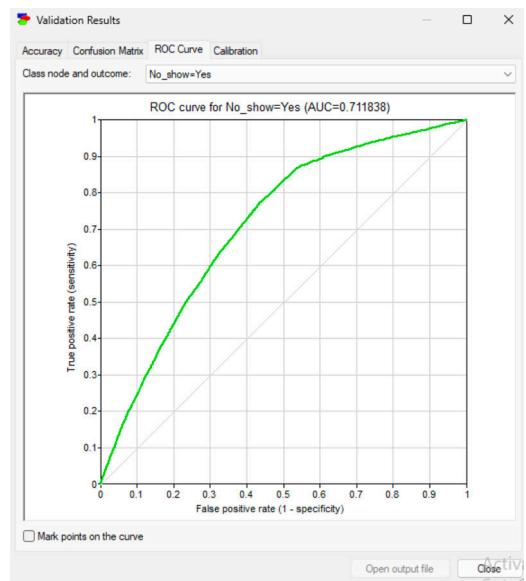
The ROC curves were generated for the Bayesian Search model. The area under the curve (AUC) was found to be approximately **0.692** for both classes ("No-show" = Yes and No).

**This result suggests that the model has a moderate ability to distinguish between the classes.**

## ROC Curve Analysis for the PC Algorithm Model



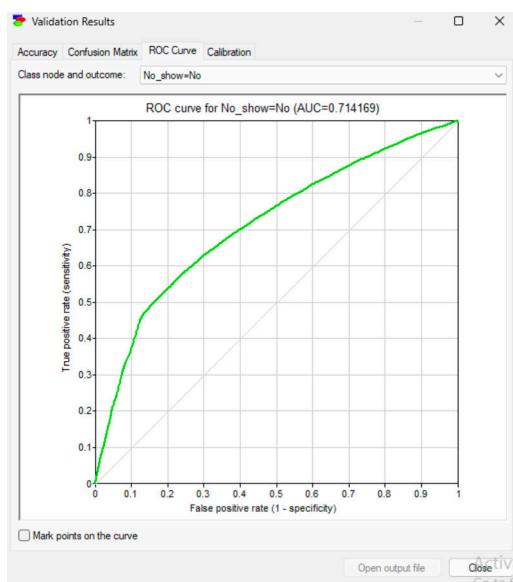
No Show = No



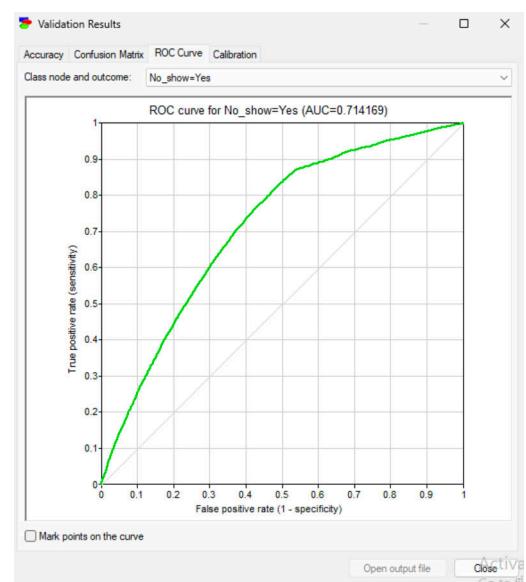
No Show = Yes

The ROC curves were generated for the PC Algorithm model. The area under the curve (AUC) was found to be approximately **0.712** for both classes ("No-show" = Yes and No). An AUC of 0.712 suggests that the **PC Algorithm model has a moderate ability to distinguish between patients who will show up and those who will not.**

## ROC Curve Analysis for the TAN Model



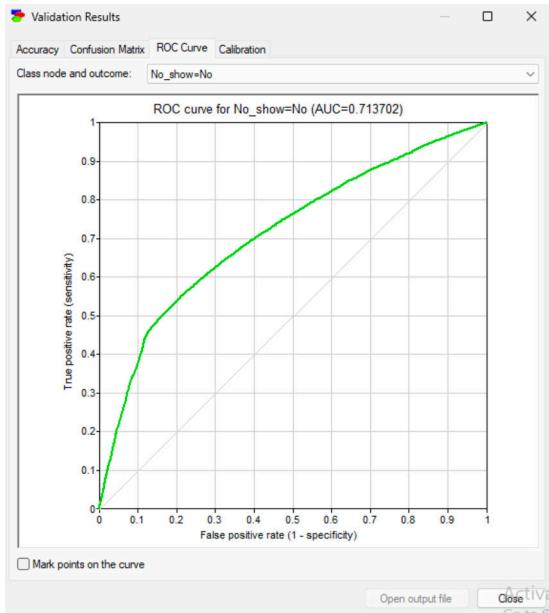
No Show = No



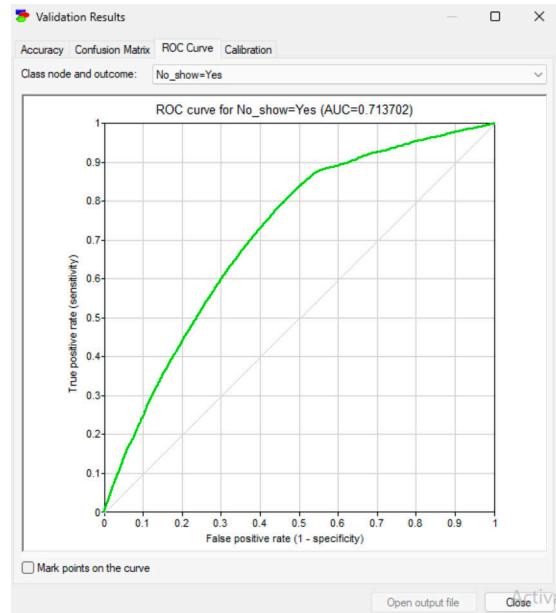
No Show = Yes

The ROC curves were generated for the TAN (Tree Augmented Naive Bayes) model. The area under the curve (AUC) was found to be approximately **0.714** for both classes ("No-show" = Yes and No). An AUC of 0.714 suggests that the **TAN model has a moderate ability to distinguish between patients who will show up and those who will not.**

## ROC Curve Analysis for the ANB Model



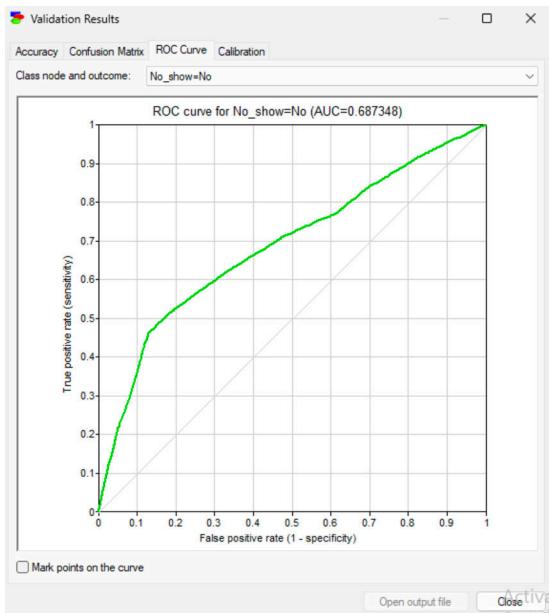
No Show = No



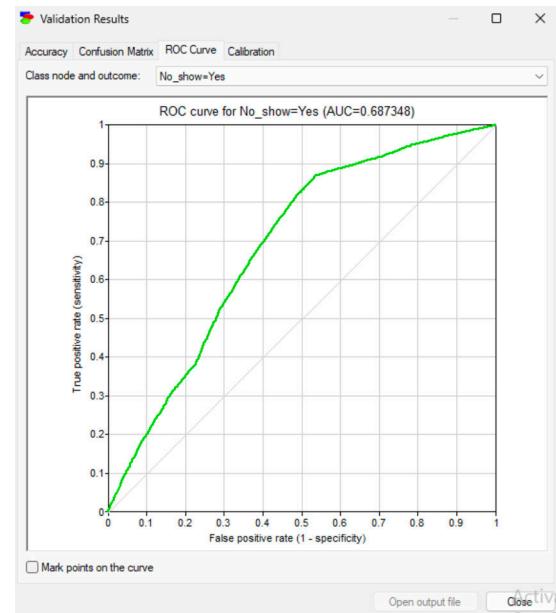
No Show = Yes

The ROC curves were generated for the ANB model. The area under the curve (AUC) was found to be approximately **0.714** for both classes ("No-show" = Yes and No). An AUC of 0.714 suggests that the **ANB model has a moderate ability to distinguish between patients who will show up and those who will not**.

## ROC Curve Analysis for the Naive Bayes Model



No Show = No



No Show = Yes

The ROC curves were generated for the Naive Bayes model. The area under the curve (AUC) was found to be approximately **0.687** for both classes ("No-show" = Yes and No). An AUC of 0.687 suggests that the **Naive Bayes model has a moderate ability to distinguish between patients who will show up and those who will not**.

## **Handling Class Imbalance with SMOTE**

In the initial dataset, there was a significant class imbalance:

- The majority of the records belonged to patients who showed up for their appointments.
- Only a small proportion of the data represented no-show patients.

As a result, the machine learning models trained on this data were heavily biased:

- They were very good at predicting patients who would come. However, they performed poorly in predicting patients who would not show up, resulting in extremely low sensitivity for the no-show class.

To address this imbalance, **SMOTE** (Synthetic Minority Over-sampling Technique) was applied.

### **What is SMOTE?**

SMOTE is a popular data balancing technique that generates synthetic samples for the minority class. Instead of simply duplicating existing minority class examples, SMOTE creates new, similar instances by interpolating between existing ones.

### **How SMOTE Works:**

- For each sample in the minority class, SMOTE identifies its k nearest neighbors.
- New synthetic examples are created along the line segments joining the sample and its neighbors.
- This approach helps the model learn a more generalized decision boundary rather than simply memorizing existing examples.

The key benefit of SMOTE is that it increases the number of minority class samples without duplicating data, thus reducing the risk of overfitting.

### **How was SMOTE Implemented?**

To apply SMOTE, the following tools and steps were used:

- Programming Language: Python
- Libraries:
  - imbalanced-learn for SMOTE implementation
  - scikit-learn for machine learning utilities
  - pandas for data handling and preprocessing
- Preprocessing:
  - All categorical variables were converted to numerical values using Label Encoding, as SMOTE requires numeric input.
  - The target variable (No-show) was mapped to 0 (show) and 1 (no-show).
- SMOTE Application:
  - SMOTE was applied on the training data to generate synthetic minority class examples.
  - The new, balanced dataset was saved as a CSV file for further model training in GeNIE.

## Sample Code for SMOTE Implementation

```
import pandas as pd
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import LabelEncoder

# Load the dataset
data = pd.read_csv('Uploaded_CSV_File.csv')

# Separate features and target variable
X = data.drop('No-show', axis=1)
y = data['No-show']

# Encode categorical variables
for column in X.columns:
    if X[column].dtype == 'object':
        le = LabelEncoder()
        X[column] = le.fit_transform(X[column])

# Encode target variable
if y.dtype == 'object':
    y = y.map({'No': 0, 'Yes': 1})

# Apply SMOTE
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X, y)

# Combine and save the balanced dataset
balanced_data = pd.concat([pd.DataFrame(X_resampled, columns=X.columns), pd.Series(y_resampled, name='No-show')], axis=1)
balanced_data.to_csv('balanced_data.csv', index=False)
```

This balanced dataset was then used to retrain the models in GeNIE for further analysis.

### Next Step

After balancing the data with SMOTE, the models were retrained, and their performance was re-evaluated. The following sections present the comparison of model performance before and after applying SMOTE, starting with the Bayesian Search model.

## Impact of SMOTE on Bayesian Search Model Performance

Below is a comparison of the Bayesian Search model's performance before and after applying SMOTE:

Metric	Before SMOTE	After SMOTE
<b>Overall Accuracy</b>	79.8%	72.73%
<b>Sensitivity (No-show = Yes)</b>	0.0%	74.87%
<b>Specificity (No-show = No)</b>	%100	70.59%

### Analysis

- Before SMOTE:**  
The Bayesian Search model achieved a high overall accuracy of 79.8%, but the sensitivity for the no-show class was extremely low at 0.0%. This indicates a strong bias toward predicting patients who would attend their appointments, rendering the model ineffective for identifying no-shows.
- After SMOTE:**  
After applying SMOTE, the sensitivity dramatically improved to 74.87%, significantly enhancing the model's ability to detect no-show patients. Although the overall accuracy dropped slightly to 72.73%, this trade-off was expected and acceptable, given the primary objective of improving minority class detection.  
Furthermore, the AUC increased from 0.692 to 0.796, indicating an overall improvement in the model's ability to discriminate between the two classes.
- Trade-offs:**  
The balancing process led to a reduction in specificity (from 99.43% to 70.59%), meaning that the model became less accurate in predicting show-ups. However, in real-world healthcare applications, accurately predicting no-shows is more critical for resource planning and efficiency.

### Conclusion for Bayesian Search Model

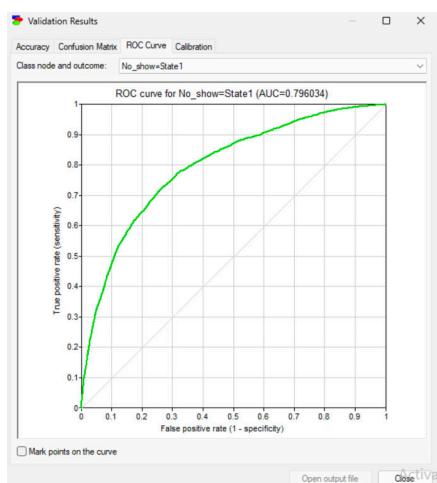
Applying SMOTE to the dataset significantly improved the Bayesian Search model's performance in predicting no-shows. Although there was a slight decrease in overall accuracy and specificity, the model became much more balanced and capable of identifying no-show patients effectively, as reflected in the substantial increase in sensitivity and AUC.

		Predicted	
		State0	State1
Act.	State0	62268	25940
	State1	22164	66044

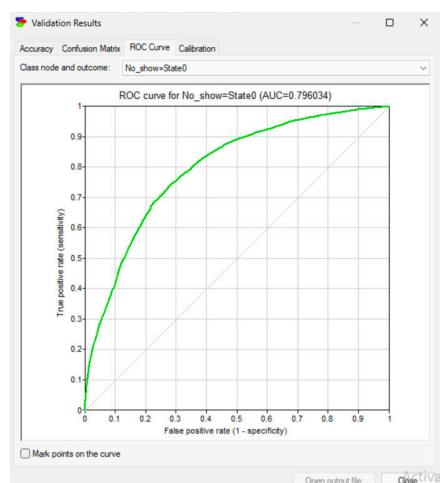
Confusion Matrix

Accuracy:  
No-show = 0.727326 (128312/176416)  
State0 = 0.705922 (62268/88208)  
State1 = 0.74873 (66044/88208)

Accuracy



No Show = Yes



No Show = No

## Impact of SMOTE on TAN Model Performance

Below is a comparison of the TAN model's performance before and after applying SMOTE:

Metric	Before SMOTE	After SMOTE
<b>Overall Accuracy</b>	79.8%	73.04%
<b>Sensitivity (No-show = Yes)</b>	0.0448%	75.02%
<b>Specificity (No-show = No)</b>	99.99%	71.05%

### Analysis

- Before SMOTE:**  
The TAN model demonstrated a high overall accuracy of 79.8%, but its sensitivity for the no-show class was extremely low at 0.0448%. The model primarily focused on predicting patients who would attend, largely ignoring the minority class.
- After SMOTE:**  
After applying SMOTE, the model's sensitivity greatly improved to 75.02%, substantially enhancing its ability to detect no-show patients. Although the overall accuracy dropped slightly to 73.04%, the improvement in detecting the minority class justifies the trade-off.  
Moreover, the AUC increased from 0.714 to 0.799, indicating an improved overall model performance in distinguishing between show and no-show patients.
- Trade-offs:**  
There was a decrease in specificity (from 99.34% to 71.05%), but the improved balance between sensitivity and specificity suggests that the model became more reliable for practical applications, where accurately predicting no-shows is critical for optimizing healthcare operations.

### Conclusion for TAN Model

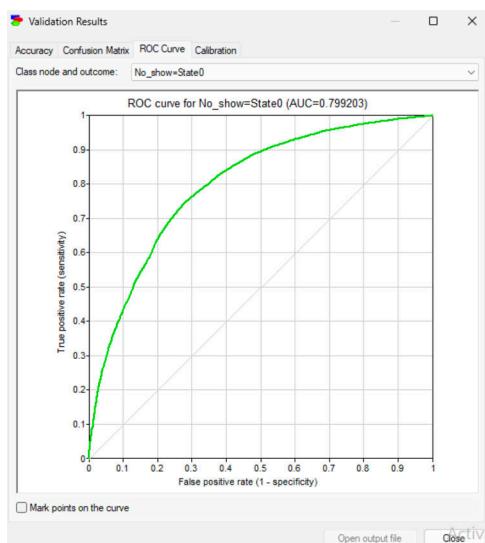
Applying SMOTE to the dataset significantly enhanced the TAN model's performance in predicting no-shows. Despite a slight decline in overall accuracy and specificity, the model became much better at recognizing patients who are likely to miss their appointments, as shown by the sharp increase in sensitivity and AUC.

		Predicted	
		State0	State1
Act.	State0	62676	25532
	State1	22031	66177

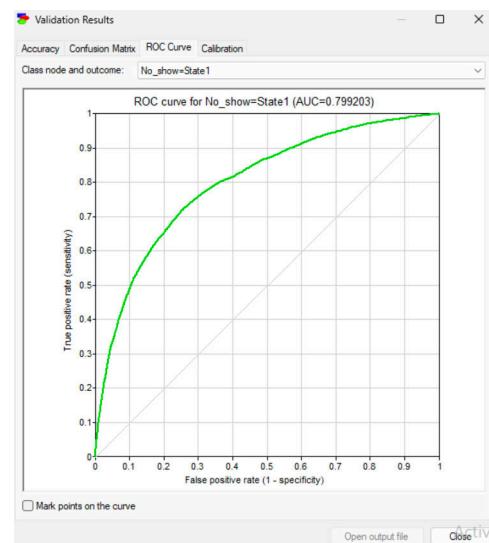
Confusion Matrix

Accuracy:  
 No-show = 0.730393 (128853/176416)  
 State0 = 0.710548 (62676/88208)  
 State1 = 0.750238 (66177/88208)

Accuracy



No Show = No



No Show = Yes

## Impact of SMOTE on Naive Bayes Model Performance

Below is a comparison of the Naive Bayes model's performance before and after applying SMOTE:

Metric	Before SMOTE	After SMOTE
<b>Overall Accuracy</b>	79.82%	72.07%
<b>Sensitivity (No-show = Yes)</b>	2.02%	76.02%
<b>Specificity (No-show = No)</b>	99.12%	68.11%

### Analysis

- Before SMOTE:**  
The Naive Bayes model achieved a relatively high overall accuracy of 79.82%, but its sensitivity for the no-show class was very low at 5.40%, indicating it was ineffective at predicting patients who would not attend their appointments.
- After SMOTE:**  
With the application of SMOTE, the model's sensitivity greatly improved to 76.02%, significantly enhancing its ability to detect no-show patients. Although overall accuracy decreased to 72.07%, the improvement in minority class detection was substantial.  
Additionally, the AUC increased from 0.65 to 0.781, suggesting a better overall model performance in distinguishing between classes.
- Trade-offs:**  
As seen in other models, specificity decreased from 99.12% to 68.11%, but the model became far more capable of identifying no-show patients, which is critical for operational planning in healthcare environments.

### Conclusion for Naive Bayes Model

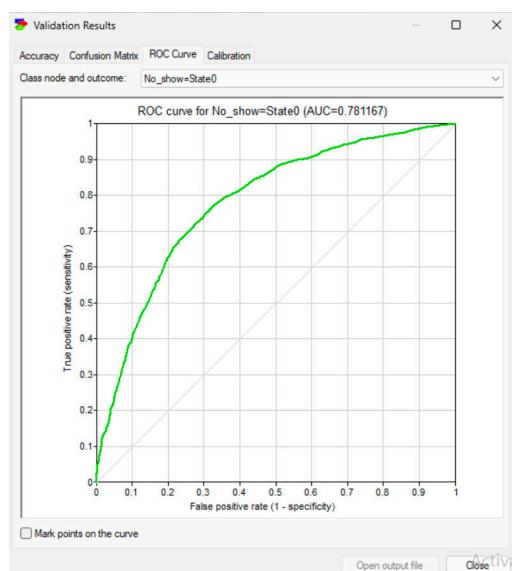
Applying SMOTE significantly enhanced the Naive Bayes model's ability to predict no-shows. Despite a slight reduction in accuracy and specificity, the model became much more balanced, showing a considerable improvement in sensitivity and AUC, which are essential for correctly identifying patients who are likely to miss their appointments.

		Predicted	
		State0	State1
Actual	State0	60085	28123
	State1	21152	67056

Confusion Matrix

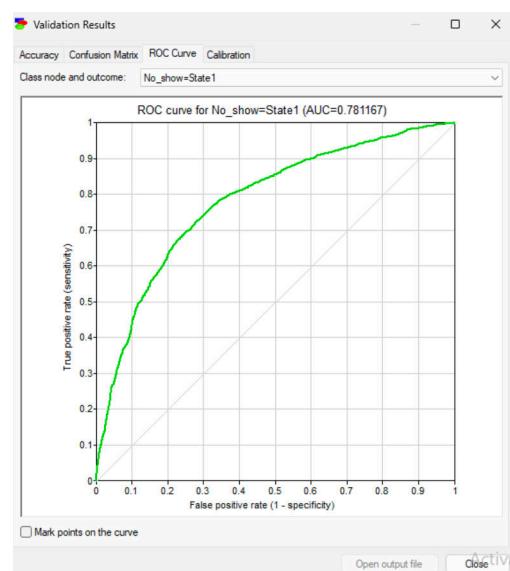
Accuracy:  
No-show = 0.720689 (127141/176416)  
State0 = 0.681174 (60085/88208)  
State1 = 0.760203 (67056/88208)

Accuracy



10) 10.

No Show = No



No Show = Yes

## Conclusions

In this project, I analyzed the Medical Appointment No Shows Dataset, a real-world dataset collected from medical appointment records in Vitória, Brazil. The dataset included demographic and health-related attributes such as gender, age, neighborhood, medical conditions (hypertension, diabetes, alcoholism, handicap), and appointment details (scheduled day, appointment day, receipt of SMS reminders). **The main objective was to predict whether a patient would show up for their scheduled medical appointment.**

I built and compared several Bayesian network models:

- Bayesian Search
- PC Algorithm
- Tree-Augmented Naive Bayes (TAN)
- Augmented Naive Bayes (ANB)
- Naive Bayes

Each model was evaluated based on structural complexity, with the PC Algorithm generating the densest network, while the Naive Bayes model had the simplest, most interpretable structure.

When evaluating model performance using 10-fold cross-validation, I observed:

- High overall accuracy (~79-80%) across all models.
- Extremely low sensitivity (ability to correctly predict no-shows).
- Very high specificity (correct prediction of patients who showed up).

This imbalance indicated a critical limitation: while the models were good at predicting patients who would attend their appointments, they failed to identify those who would not. This is a classic issue arising from class imbalance, where the number of no-shows is much smaller than the number of shows.

To address this, I applied **SMOTE** (Synthetic Minority Over-sampling Technique):

- SMOTE generates synthetic examples of the minority class (no-shows) by interpolating between existing examples, creating a more balanced dataset.
- The categorical features were label-encoded to make the data suitable for SMOTE application.
- After balancing the data, the models were retrained using the new dataset.

## Impact of SMOTE

The application of SMOTE had a significant positive impact:

- Sensitivity dramatically improved across all models, reaching around 75% or higher, allowing the models to identify no-shows much more accurately.
- Although overall accuracy decreased slightly (due to more balanced predictions), the trade-off was acceptable given the substantial gain in minority class detection.
- AUC (Area Under the ROC Curve) scores improved consistently, indicating better model discrimination between the classes.
- There was a predictable decrease in specificity, but the balanced improvement meant the models became much more practical for real-world deployment.

Before SMOTE, the models exhibited high accuracy but poor detection of no-shows, making them ineffective for hospital operational planning where missing an appointment has a real cost.

After SMOTE, the models became far more balanced and reliable for the intended task of predicting no-shows, offering a much-needed improvement in model fairness and utility.

## Summary

This project highlights the importance of handling class imbalance in real-world datasets. Through Bayesian network modeling and the application of SMOTE, I was able to significantly enhance the models' performance, particularly their ability to predict no-shows. Such improvements are critical for healthcare systems seeking to optimize resource allocation and reduce the operational impacts of missed appointments.

Sevda Ghasemzadehnaghadehy