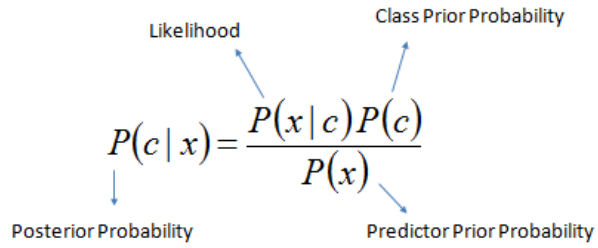
Task

Explore the behavior of the Bayesian classifier in continuously valued, numerical domains. Observe differences between assuming a normal distribution and using kernel functions.

Background

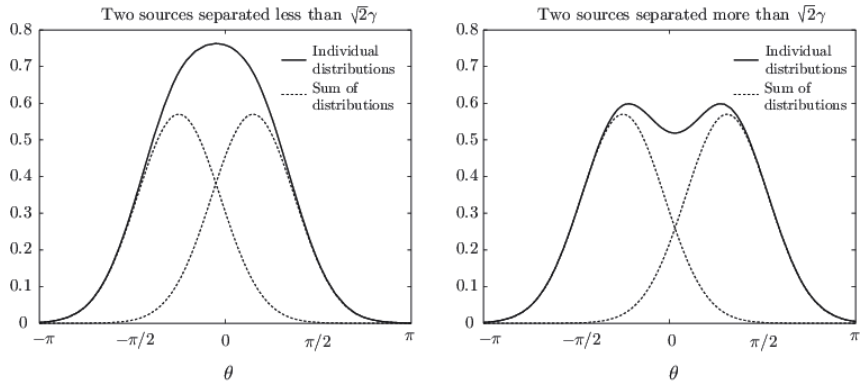
The Bayesian classifier is a probabilistic model used to make the most probable classifications of a new example. The classifier stems from the Bayes Theorem which provides a system to calculate the conditional probability of a given outcome, given the probability of another outcome that has already occurred. This project will be evaluating the performance of two alternative Bayesian classifiers: one based on the assumption of normal distributions, the other using kernel functions (Sums of Gaussians)

The Naive Bayes assumption is the assumption that each attribute either makes an independent or equal contribution to the outcome of a given example. It is represented by the formula below:



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

The Sum of Gaussians assumption utilizes multiple normal distributions with different averages to come up with a composition of all the normal distributions.



Classification Procedure

To ensure that the classification was done fairly, the following procedure was followed rigorously for any of our selected data sets:

- Continuous attributes/features were selected to determine dependent variable (X)
- One continuous attribute was selected as our target value/class (Y)
- For the Y training and testing set, continuous values were discretized to ensure that the model ran efficiently without bias. We discretized values into 4 buckets since we observed anything above this adversely affected model outcome
- X and Y were split into training and testing sets with a test size of 0.33 and a random state value of 17. We used the random_state parameter to ensure the outcomes were the same for subsequent runs of the model.
- Run the Naive Bayes and Kernel Function model on our data. The models return an accuracy score and error rate which we represent in a data frame.
- Using the information on the data frame, we plot a simple bar graph of the accuracy score
- Interpreted the potential differences between the alternative versions of the classifier

Goal

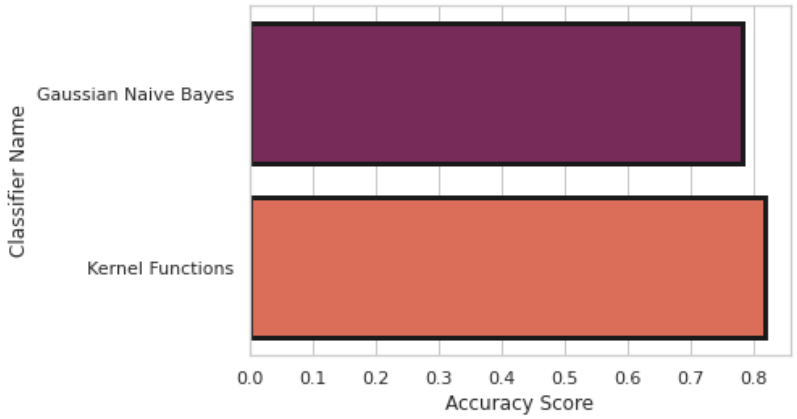
For this task, we will examine three datasets namely: Dow Jones Index data set, Absenteeism at work data set and the popular Iris data set. Each data set will be run on alternative versions of the Bayesian classifier: one based on the assumption of normal distributions, the other using kernel functions (sums of Gaussians). We will comment on why both classifiers give different or similar results and why one performs better than the other for a given data set. Since the focus of this project is classifier comparison, we will not focus on the spread of training and testing values. This will however be evaluated after further interrogation in class.

Dow Jones Index (Stock) Data Set

The stocks comprising the Dow Jones Index are some of the largest and most successful publicly traded stocks in the United States. The index features companies like AT&T, Microsoft, Walmart, Coca-Cola, etc thus being able to predict the success of the entire index based on the previous week's data serves as a lucrative endeavor. Here is a breakdown of the dataset:

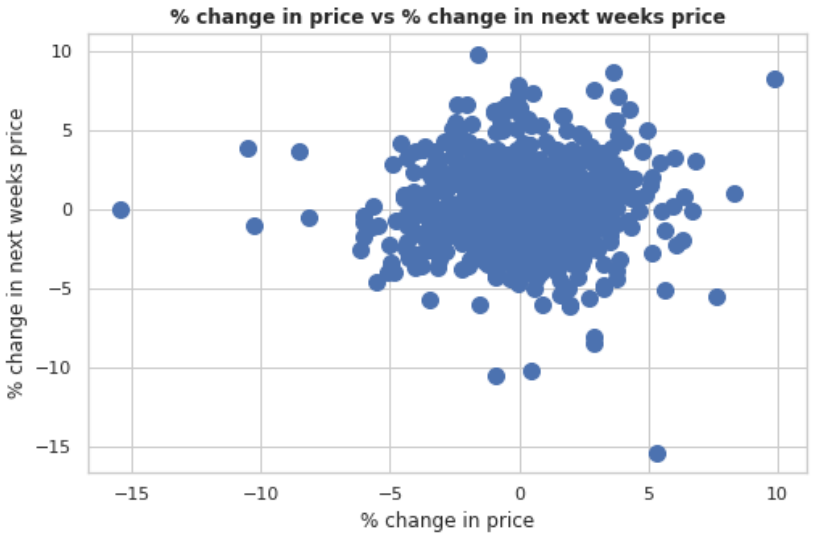
- 750 Instances
- 21 Attributes
  Quarter, stock, date, open, high, low, close, volume, percent_change_price, percent_change_volume_over_last_wk, previous_weeks_volume, next_weeks_open,next_weeks_close, percent_change_next_weeks_price, days_to_next_dividend, percent_return_next_dividend
- We selected **percent_change_next_weeks_price** as our class identifier.

These were the results we obtained after running against the two Bayesian classifiers:



Classifier Name	Accuracy Score	Error Rate
Gaussian Naive Bayes	0.782258	0.217742
Kernel Functions	0.818548	0.181452

The Gaussian Naive Bayes had an accuracy score of approximately 78% (22% error rate) whilst the kernel functions obtained an accuracy score of approximately 82% (18% error rate). Even though the values here are relatively close, we believe the kernel functions achieved a lower error rate since it was able to account for values within our data set that were widely spread amidst a given cluster of values. This is evident when you observe the relationship between the percent change in next week's price and the percentage change in price. The kernel functions will account for outliers greater than a 5% change in price and less than a -5% change in price.

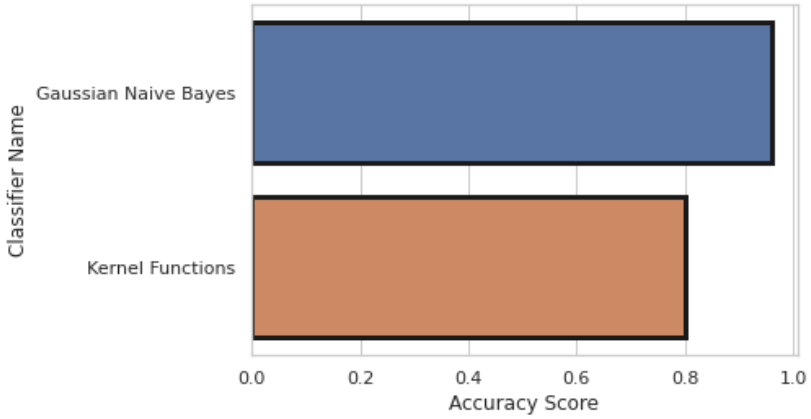


Absenteeism at Work Data Set

Amidst the growing need to be absent for work especially in a pandemic, we selected a dataset that evaluated absenteeism at work at the Universidade Nove de Julho in Sau Paulo, Brazil. This dataset contains the demographics as well as the reasons why employees failed to show up on a given workday. For this exercise, we wanted to decipher whether an employee's average daily workload could increase the chances of them getting absent. Here is a breakdown of the dataset:

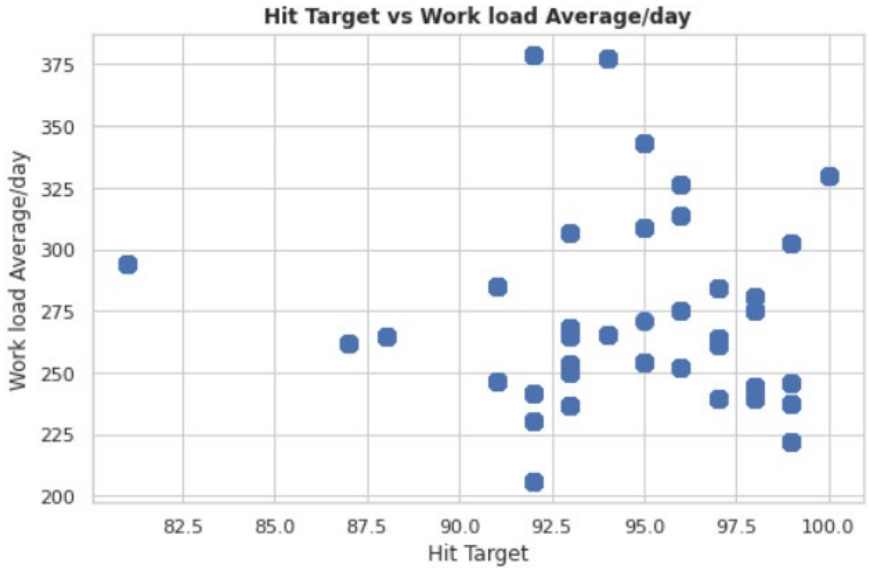
- 740 instances
- 21 attributes
  ID, Reason for absence, Month of absence, Day of the week, Seasons, Transportation expense, Distance from Residence to Work, Service time, Age, Work load Average/day, Hit target, Disciplinary failure, Education, Son, Social drinker, Social smoker, Pet, Weight, Height, Body mass index, Absenteeism time in hours
- We selected **Work load Average/day** as our class identifier.

These were the results we obtained after running the dataset against the two Bayesian classifiers:



Classifier Name	Accuracy Score	Error Rates
Gaussian Naive Bayes	0.955102	0.044898
Kernel Functions	0.910204	0.089796

The Gaussian Naive Bayes had an accuracy score of approximately 96% (4% error rate) whilst the kernel functions obtained an accuracy score of approximately 91% (9% error rate). Even though the values here are relatively close, we believe that the Naive Bayes achieved a relatively higher score since the data from below follows the trend of a normal distribution. The kernel function method accounts for outliers in this scenario and this slightly affects the accuracy score. This dataset proved to be the most difficult to plot and identify a normal relationship.



Iris Data Set

The Iris Data Set is a multivariate data set introduced by the British statistician, eugenicist, and biologist Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis*. The data set contains information that details the measurements of different Irises petals and sepals. The data set is broken down into:

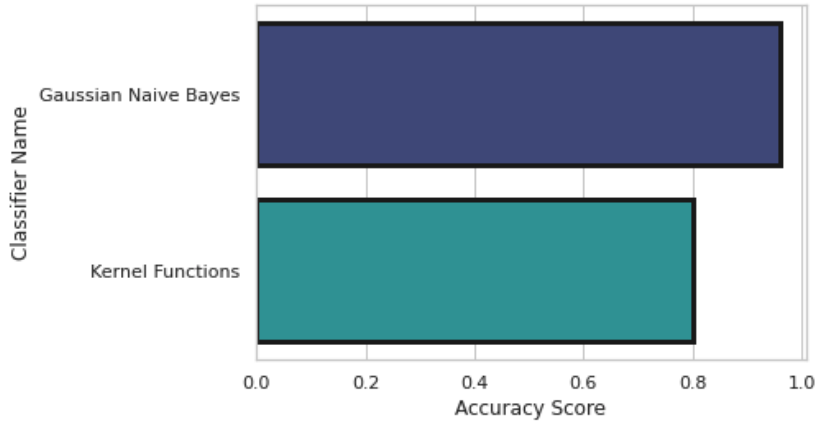
-150 Instances

-4 Attributes

Sepal Length in cm, Sepal Width in cm, Petal length in cm, and Petal width in cm

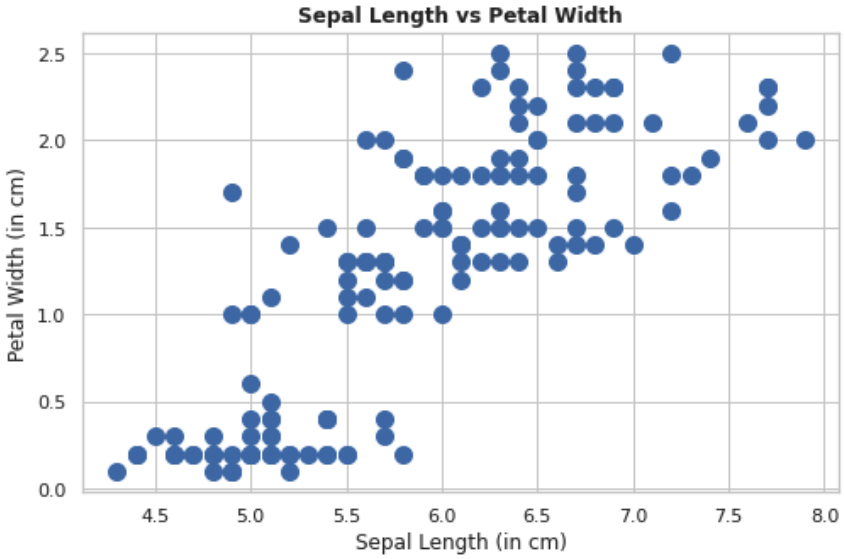
-3 Class Labels: Iris Setosa, Iris Versicolour, and Iris Virginica

Results after running the Gaussian and the Kernel Function algorithms on the data set



Classifier Name	Accuracy Score	Error Rates
Gaussian Naive Bayes	0.96	0.04
Kernel Functions	0.80	0.20

For this data set, the Gaussian Naive Bayes algorithm ran with 96% accuracy (4% error rate) and the Kernel Functions algorithm ran with 80% accuracy (20% error rate). As seen in the graph below, Sepal Length vs Petal Width, the data points plotted trend towards a normal. In this experiment, the Gaussian Naive Bayes algorithm was able to run with a higher degree of success due to the semi-normal distribution of the attribute values.



Conclusion

The Naive Bayes Algorithm was very effective in classifying data sets. Specifically, in the Iris data set, it outperformed the Kernel Functions algorithm due to the normal distribution of the data. In the Absenteeism at Work Data set, the data followed a more random distribution and for that reason, the performance of the Naive Bayes and the Kernel Functions were closer to one another.

We also observed that kernel functions performed better when the dataset values form clusters. This is particularly observed in the Dow Jones Index data set, which presented one main cluster with some noise around it when plotted.

Data Sources

Google Collaboratory Implementation
https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work
https://archive.ics.uci.edu/ml/datasets/Dow+Jones+Index
https://archive.ics.uci.edu/ml/datasets/Iris