Google's Personalized Advertising Recommendation System: A study and improvement of a big data application.

Student Name: Wangjun SHEN

School: The University of South Australia

Excellent work!
28/30

Content Table

Introduction

The concept of big data refers to the vast amounts of data that are too massive, move too quickly, or have too much variety to be processed using traditional methods. Big data offers new opportunities for insights and decision-making in various fields. As Mayer-Schönberger and Cukier (2013) pointed out, big data's value lies in its ability to uncover patterns and trends that were formerly beyond the reach of conventional approaches. This rapidly expanding field has led to the development of advanced analytics techniques that can extract valuable knowledge from the vast data landscape.

Google's Personalized Ad Recommendation System

Technical framework of Google's Personalized Ad Recommendation System

Google's personalized advertising system, a cornerstone of its advertising ecosystem, stands as a prime example of harnessing big data applications.

Google's personalized advertising recommendation system analyzes and processes massive amounts of user data to provide customized ads based on individual preferences and interests.

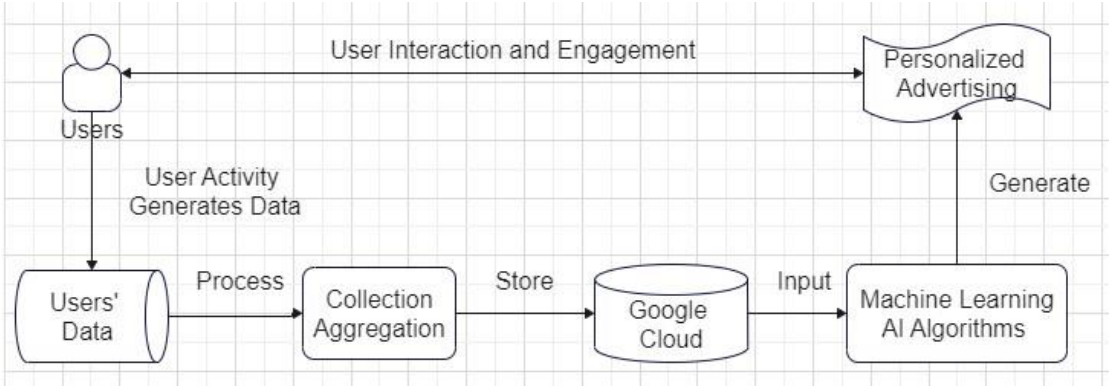


Figure 1 The architecture of Google's Personalized Ad Recommendation System

The above figure is a simplified illustration of Google's technical framework. Google has a large number of users who generate a vast amount of data. Google uses web crawlers to collect user behavior data from the internet, such as search queries and website browsing history, and tracks website visits and user interactions with tools like Google Analytics**.** Google then uses Google Cloud Storage for data storage, employing distributed database technologies such as Google Cloud Bigtable to manage large-scale user data (Google Cloud, n.d.). For stored data, Google uses Google Cloud AI services like AutoML to train custom machine learning models for predicting user behavior and interests. Google also develops deep learning models using the TensorFlow framework for more complex user behavior analysis. Google generates personalized ads based on user features and machine learning

models to ensure that ads match user interests and preferences. Using the Google AdWords platform, personalized ads are distributed to appropriate ad spaces (Google Ads, n.d.). Finally, Google uses Google Analytics to track ad click-through rates, conversion rates, and user interactions to evaluate ad effectiveness. In addition, Google is an avid user of A/B testing. Google extensively uses A/B testing in its personalized advertising services to improve user satisfaction and ad effectiveness (Kohavi et al., 2012).

2/2

Big Data Characteristics of Google's Personalized Ad Recommendation System

The platform efficiently manages, analyzes, and utilizes a massive amount of data, highlighting some of the core features of big data analysis.

First, the platform handles an immense volume of data. Since 2012, Google processes over 40,000 search queries per second, resulting in an estimated 1.2 trillion searches per year (Internet Live Stats, 2009). This volume includes not only search queries but also user interactions across various Google services such as YouTube, Gmail, and Google Maps. This generates a massive dataset encompassing user behaviors, preferences, and interests. Second, the system operates at exceptional velocities, analyzing user data and delivering tailored advertisements within milliseconds of a user's query or interaction (Google Ads, n.d.). This real-time processing ensures that users are presented with ads that are contextually relevant and aligned with their immediate interests. Next, the platform's variety of handled data types, including text, images, videos, and location-based information, allows Google to paint a holistic picture of user preferences and behaviors, enabling the delivery of advertisements that resonate across multiple dimensions. Finally, Google employs advanced data validation techniques and machine learning algorithms to guarantee that the displayed advertisements are not only relevant but also accurate and trustworthy, thereby building user confidence (Google Ads, n.d.). This showcases the Veracity feature of the application.

2/2

Emotion-Aware Ad Recommendation System

The Design of Technical Framework for the Emotion-Aware Ad Recommendation System

Google's personalized advertising system has some drawbacks. These limitations include the system's inability to establish an emotional connection and its inefficiency in capturing context (Ma et al., 2022). Although it can recommend ads based on users' past behavior, it may not be able to accurately capture their current situation or emotional state. This can lead to ads that feel inappropriate or ill-timed. Moreover, advertisers may find it difficult to convey emotions through personalized ads, resulting in a mismatch between the ad content and the

user's emotions.

An improved system, called the Emotion-Aware Ad Recommendation System, has been suggested to overcome these limitations and build on the original system.

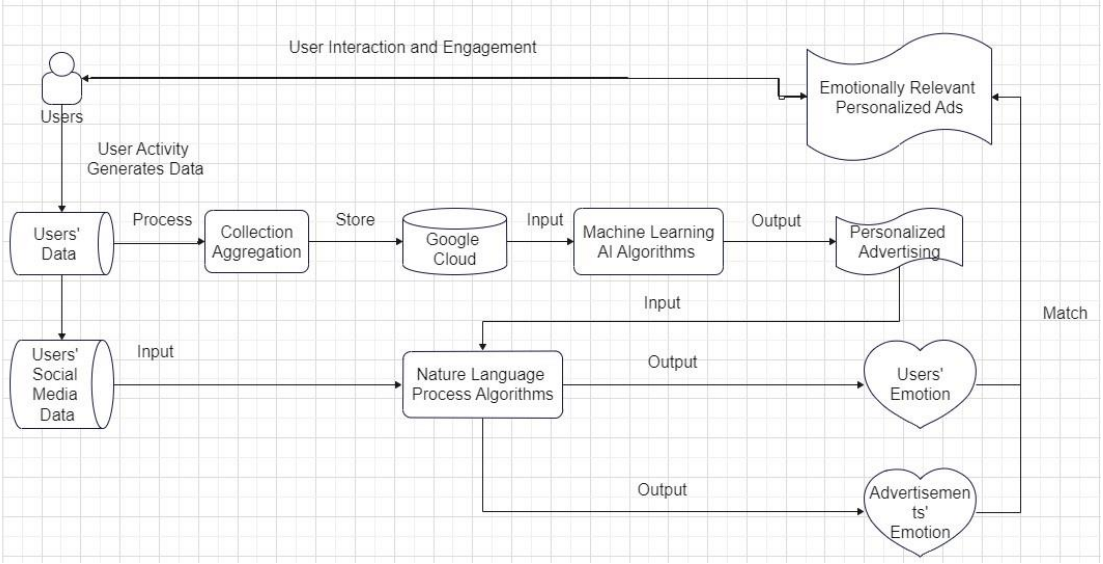


Figure 2 Emotion-Aware Ad Recommendation System

The figure above shows the framework architecture of the system. Additional functionality is added to the system while still maintaining its original features. To create a personalized emotional model, the system will utilize natural language processing techniques to analyze users' recent texts, comments, and social media content, and determine their emotional state. By combining the emotional state with the user's historical behavior data, the system creates a personalized emotional model. The system then analyzes the emotional elements and states of personalized ads recommended to users, matches the user's emotional state with ad emotional elements, and selects ads that align with the user's emotions. Finally, the system recommends the matched ads to users, enhancing emotional relevance and user engagement.

Good idea
4/4

Reasons for Using the Emotion-Aware Ad Recommendation System

The new system can enhance the relevance of personalized ads and user engagement. Emotions play a crucial role in influencing user behavior and decision-making.

An emotion-aware ad recommendation system can identify and utilize user emotions to deliver ads that resonate emotionally. Research shows that emotionally resonant ads are often more memorable and can increase engagement (Grewal, Gotlieb and Marmorstein, 1994). This new idea of understanding user emotions and providing emotion-related ads can increase click-through rates and improve ad effectiveness. Personalized and contextually relevant ads that combine emotional analysis with personalized recommendations allow the system to not only consider user preferences but also their emotional context of interacting with the platform. This makes ads not only cater to user interests but also their current emotional state. This combination of personalization and context enhances the user experience and fosters a stronger emotional connection with the brand or product (Batra and

Stayman, 1990).

Furthermore, the system can also improve user satisfaction and retention rates because ads that cater to the user's emotional state are less likely to be seen as intrusive or irrelevant. Users are more likely to appreciate ads that reflect their emotional experiences, thus increasing user satisfaction and generating positive views about the platform. This positive experience helps improve user retention and loyalty. Research on emotional ads and emotional analysis provides empirical support for the effectiveness of emotion-driven advertising. Studies have shown that emotionally appealing ads produce stronger reactions and better memory retention (Lang, 2000; Phelps et al., 2004).

Challenges Encountered by the Emotion-Aware Ad Recommendation System

One of the main challenges faced by Emotion-Aware Ad Recommendation Systems is ensuring accurate and diverse sentiment analysis. While sentiment analysis is crucial for such systems, there are several factors such as cultural differences and semantic ambiguity that can impact its effectiveness. In order to address this challenge, extensive research is required on sentiment lexicons, sentiment classification algorithms, and deep learning techniques (Cambria et al., 2013).

Another challenge is to ensure fast response time. The system must be able to identify and analyze emotional states within just a few seconds of user interaction for real-time sentiment analysis. This requires efficient text processing and sentiment classification technology, as well as fast data processing.

Integrating emotion and text context is also a challenge. Determining the emotion of an ad depends not only on the ad text but also on the context. Thus, one of the challenges is to integrate the user's emotional state with ad text and context. This may involve deep text analysis and context modeling (Pang and Lee, 2008).

With regards to anonymity, legal issues such as data protection and privacy are challenges that this application needs to face. However, these challenges are not unique to this application, and therefore will not be discussed in detail here.

Big Data Characteristics of the Emotion-Aware Ad Recommendation System

This new application demonstrates Volume, Velocity, Variety, and Veracity, which are the four major applications of big data.

Concerning Volume, the emotion-aware ad recommendation application must process an enormous amount of user sentiment data, including user interactions with the platform, textual comments, and more. The platform has already collected millions of interactions between users and ads, each of which contains user sentiment data, resulting in a vast dataset.

Regarding Velocity, user sentiment data is generated very quickly, requiring real-time analysis and response for adjusting ad recommendations. User sentiment data comes from various sources, including speech, text, images, and more. All these different data types need to be considered in sentiment analysis. Lastly, the application must ensure the accuracy and reliability of user sentiment data to avoid false or misleading sentiment labeling, which reflects the Veracity of big data.

Conclusion

In conclusion, big data applications have transformed the way businesses operate, and Google's personalized advertising system is a prime example of how to harness the power of big data. The system efficiently manages, analyzes, and utilizes a massive amount of data, highlighting the core features of big data analysis. The Emotion-Aware Ad Recommendation System builds on the original system by incorporating emotional analysis to enhance the relevance of personalized ads and increase user engagement. However, the system also faces challenges, such as accurate and diverse sentiment analysis, fast response time, and integrating emotion and text context. Despite these challenges, the benefits of such systems are clear in terms of increasing user satisfaction, engagement, and retention rates.

References

ads.google.com. (n.d.). *Google Ads - Get Customers and Sell More with Online Advertising*. [online] Available at: https://ads.google.com/intl/en_in/home/ [Accessed 19 Aug. 2023].

Batra, R. and Stayman, D.M. (1990). The Role of Mood in Advertising Effectiveness. Journal of Consumer Research, 17(2), p.203. doi:https://doi.org/10.1086/208550.

Google (n.d.). Cloud Storage. [online] Google Cloud. Available at: https://cloud.google.com/storage.

Cambria, E., Schuller, B., Xia, Y. and Havasi, C. (2013). New Avenues in Opinion Mining and Sentiment Analysis. IEEE Intelligent Systems, 28(2), pp.15–21. doi:https://doi.org/10.1109/mis.2013.30.

Grewal, D., Gotlieb, J. and Marmorstein, H. (1994). The Moderating Effects of Message Framing and Source Credibility on the Price-Perceived Risk Relationship. Journal of Consumer Research, 21(1), p.145. doi:https://doi.org/10.1086/209388.

Kohavi, R., Deng, A., Frasca, B., Longbotham, R., Walker, T. and Xu, Y. (2012). Trustworthy online controlled experiments. Knowledge Discovery and Data Mining. doi:https://doi.org/10.1145/2339530.2339653.

Lang, A. (2000). The Limited Capacity Model of Mediated Message Processing. Journal of

Communication, [online] 50(1), pp.46–70. doi:https://doi.org/10.1111/j.1460-2466.2000.tb02833.

Ma, G., Ma, J., Li, H., , Y., Wang, Z. and Zhang, B. (2022). Customer behavior in purchasing energy-saving products: Big data analytics from online reviews of e-commerce. Energy Policy, 165, p.112960. doi:https://doi.org/10.1016/j.enpol.2022.112960.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, [online] 2(1-2), pp.1–135. Available at: https://www.cs.cornell.edu/home/llee/omsa/omsa.pdf.

Phelps, J.E., Lewis, R., Mobilio, L., Perry, D. and Raman, N. (2004). Viral Marketing or Electronic Word-of-Mouth Advertising: Examining Consumer Responses and Motivations to Pass Along Email. Journal of Advertising Research, 44(4), pp.333–348. doi:https://doi.org/10.1017/s0021849904040371.

Šercar, T.-M. (2013). Viktor Mayer-Schönberger and Kenneth Cukier, Big Data: A Revolution That Will Transform How We Live, Work, and Think. Organizacija znanja, 18(1-4), pp.47–49. doi: https://doi.org/10.3359/oz1314047.

Bayesian Network Exploration Based on Breast Cancer Wisconsin

Data Exploratory and Build a Naive Bayes Model

First, determine the missing value status of the dataset:

Attribute Name	Missing Values
Sample_code_number	0
Clump_thickness	0
Uniformity_of_cell_size	0
Uniformity_of_cell_shape	0
Marginal_adhesion	0
Single_epithelial_cell_size	0
Bare_nuclei	16
Bland_chromatin	0
Normal_nucleoli	0
Mitoses	0
Class	0

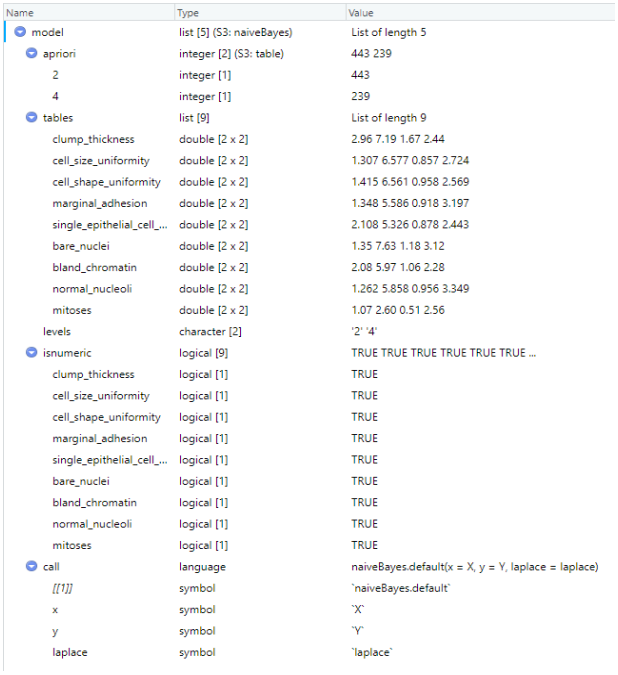
Table 1: Missing Values for Attributes

Since the number of rows containing missing values accounts for only about 2.29% of the entire dataset, and the dataset contains almost 700 examples, the method adopted here is to simply remove these rows containing missing values.

Attribute Name	Attribute Type
Clump_thickness	Numeric
Uniformity_of_cell_size	Numeric
Uniformity_of_cell_shape	Numeric
Marginal_adhesion	Numeric
Single_epithelial_cell_size	Numeric
Bare_nuclei	Numeric
Bland_chromatin	Numeric
Normal_nucleoli	Numeric
Mitoses	Numeric
Class	Factor

Table 2: Data Types for Attributes

Then, the Sample_code_number is removed from the dataset and all other attributes are convert from the original type to numeric type except the Class attribute, it is converted into factor type because is the target for the model training the next step.



Name	Type	Value
model	list [5] (S3: naiveBayes)	List of length 5
apriori	integer [2] (S3: table)	443 239
2	integer [1]	443
4	integer [1]	239
tables	list [9]	List of length 9
clump_thickness	double [2 x 2]	2.96 7.19 1.67 2.44
cell_size_uniformity	double [2 x 2]	1.307 6.577 0.857 2.724
cell_shape_uniformity	double [2 x 2]	1.415 6.561 0.958 2.569
marginal_adhesion	double [2 x 2]	1.348 5.586 0.918 3.197
single_epithelial_cell_...	double [2 x 2]	2.108 5.326 0.878 2.443
bare_nuclei	double [2 x 2]	1.35 7.63 1.18 3.12
bland_chromatin	double [2 x 2]	2.08 5.97 1.06 2.28
normal_nucleoli	double [2 x 2]	1.262 5.858 0.956 3.349
mitoses	double [2 x 2]	1.07 2.60 0.51 2.56
levels	character [2]	'2' '4'
isnumeric	logical [9]	TRUE TRUE TRUE TRUE TRUE TRUE ...
clump_thickness	logical [1]	TRUE
cell_size_uniformity	logical [1]	TRUE
cell_shape_uniformity	logical [1]	TRUE
marginal_adhesion	logical [1]	TRUE
single_epithelial_cell_...	logical [1]	TRUE
bare_nuclei	logical [1]	TRUE
bland_chromatin	logical [1]	TRUE
normal_nucleoli	logical [1]	TRUE
mitoses	logical [1]	TRUE
call	language	naiveBayes.default(x = X, y = Y, laplace = laplace)
[[1]]	symbol	`naiveBayes.default`
x	symbol	`X`
y	symbol	`Y`
laplace	symbol	`laplace`

Figure 3 Details for the Naïve Bayes Model

A Naive Bayesian model is trained to predict benign and malignant with the information shown above.

For the mathematical details of the model, you can refer to the following figure:

```
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        2         4
0.6495601 0.3504399

Conditional probabilities:
   clump_thickness
Y       [,1]     [,2]
  2 2.959368 1.671743
  4 7.188285 2.437907

   cell_size_uniformity
Y       [,1]      [,2]
  2 1.306998 0.8565007
  4 6.577406 2.7242438

   cell_shape_uniformity
Y       [,1]      [,2]
  2 1.415350 0.9579102
  4 6.560669 2.5691040

   marginal_adhesion
Y       [,1]      [,2]
  2 1.347630 0.9179766
  4 5.585774 3.1966314

   single_epithelial_cell_size
Y       [,1]      [,2]
  2 2.108352 0.8780881
  4 5.326360 2.4430866

   bare_nuclei
Y       [,1]     [,2]
  2 1.347630 1.179064
  4 7.627615 3.116679

   bland_chromatin
Y       [,1]     [,2]
  2 2.081264 1.062604
  4 5.974895 2.282422

   normal_nucleoli
Y       [,1]      [,2]
  2 1.261851 0.9556042
  4 5.857741 3.3488761

   mitoses
Y       [,1]      [,2]
  2 1.065463 0.5103046
  4 2.602510 2.5644946
```

3/3

Figure 4: Probabilities of the Naive Bayes Model

The first part is A-priori probabilities. These are the probabilities of each class appearing in the dataset without taking any features into account. According to your data, class 2 has an a-priori probability of approximately 0.6496, while class 4 has an a-priori probability of around 0.3504. These probabilities help the model understand the distribution of classes in the dataset. The second section covers conditional probabilities. Conditional probability refers to the likelihood of each feature value given a specific category. Each section corresponds to a feature and displays the conditional probability of each value of that feature for each category. For example, taking clump_thickness as an example, when clump_thickness is 2 and the category is 2, the conditional probability is approximately 2.9594. When clump_thickness is 2 and the category is 4, the conditional probability is approximately 1.6717. These probabilities indicate that, based on the training data, instances with a clump_thickness value of 2 are more likely to belong to category 2 than category 4. The same logic can be applied to other attributes.

Explain How the Model Makes Classification

Attribute	Value
clump_thickness	1
cell_size_uniformity	5
cell_shape_uniformity	4
marginal_adhesion	4
single_epithelial_cell_size	5
bare_nuclei	7
bland_chromatin	10
normal_nucleoli	3
mitoses	2

Table 4: A Record from the Dataset

The table above is a record from a dataset, which will be used as an example to illustrate how the Naive Bayes model performs classification.

The model first calculates the a-priori probabilities for each class:

$$P(Y = 2) = 0.6495601\ (Class\ 2)$$
$$P(Y = 4) = 0.3504399\ (Class\ 4)$$

For each feature in the record, the model calculates the conditional probability for each category. For example, taking clump thickness:

$$P(clump\_thickness = 1 \mid Y = 2) = 2.959368$$
$$P(clump\_thickness = 1 \mid Y = 4) = 7.188285$$

Probabilities <= 1

Similarly, for clump_thickness:

$$P(clump\_thickness = 1 \mid Y = 2) = 2.959368$$
$$P(clump\_thickness = 1 \mid Y = 4) = 7.188285$$

Similar calculations are performed for each feature. Next, the model calculates the likelihood for each class by multiplying the conditional probabilities for all features in the record. For example, for Class 2:

This is numerical data, you would have to use the Gaussian distribution formula to work out the probabilities

$$Likelihood\ for\ Class\ 2\ =\ P(Y=2)\ *\ P(clump\_thickness=1\mid Y=2)\ *\ P(cell\_size\_uniformity=5\mid Y=2)\ *\ldots$$

The calculated likelihood for each class is then multiplied with the a-priori probability for that class. For Class 2:

$$Final\ Likelihood\ for\ Class\ 2\ =\ Likelihood\ for\ Class\ 2\ *\ P(Y=2)$$

Repeat the calculation steps from above for Class 4:

$$Likelihood\ for\ Class\ 4\ =\ P(Y=4)\ *\ P(clump\_thickness=1\mid Y=4)\ *\ P(cell\_size\_uniformity=5\mid Y=4)\ *\ldots$$

$$Final\ Likelihood\ for\ Class\ 4\ =\ Likelihood\ for\ Class\ 4\ *\ P(Y=4)$$

Finally, compare the final likelihoods for both Class 2 and Class 4. The class with the higher final likelihood becomes the predicted class for the given record. By calculation, the probability of belonging to class = 2 is 0.6495601, and the probability of belonging to class = 4 is 0.3504399, so that record's class will be 2.

Confusion Matrix of the Model by Using 10-fold Cross Validation

Next, use 10-fold cross validation to calculate the accuracy and confusion matrix of the model:

	Reference	
Prediction	2	4
2	62.0	0.9
4	2.9	34.2
Accuracy (average) : 0.9619		

Table 5: Cross-Validated (10 fold) Confusion Matrix

The confusion matrix displays how well the model performed across various cross-validation folds. The rows indicate the predicted classes, while the columns indicate the actual (reference) classes.

In the top-left cell, the value of 62.0% shows the average percentage of instances that were accurately predicted as class 2 (True Negative). In the top-right cell, the value of 0.9% shows the average percentage of instances that were incorrectly predicted as class 4 instead of class 2 (False Positive). In the bottom-left cell, the value of 2.9% shows the average percentage of instances that were incorrectly predicted as class 2 instead of class 4 (False Negative). In the bottom-right cell, the value of 34.2% shows the average percentage of instances that were accurately predicted as class 4 (True Positive).

In order to calculate the accuracy of the model, we take the average of the diagonal values of the confusion matrix and divide it by the total number of instances. For this particular model, the average accuracy across the 10 folds of cross-validation is approximately 96.19%, indicating that the model correctly predicted the class of roughly 96.19% of the instances during the cross-validation process.

Precision is a metric that measures the accuracy of positive predictions made by a model. It specifically focuses on positive predictions for specific categories.

Precision is calculated as follows:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

From the cross-validated confusion matrix, it can been saw that for the "malignant" class (class 4): True Positives = 34.2% (average across folds) and False Positives = 0.9% (average across folds).

Substituting these values into the formula:

$$Precision = \frac{34.2}{34.2 + 0.9} \approx 0.974$$

Recall, also known as sensitivity or true positive rate, measures the ability of a model to correctly identify all instances of a particular class from all instances that actually belong to that class. In this case, it means the ability of the model to correctly identify all "malignant" cases (4 classes).

Recall can be calculated by using:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

From the cross-validated confusion matrix, it can saw that for the "malignant" class (class 4): True Positives = 34.2% (average across folds) and False Negatives = 2.9% (average across folds).

Substituting these values into the formula:

$$Recall = \frac{34.2}{34.2 + 2.9} \approx 0.921$$

So, the recall for the "malignant" class is approximately 0.921, or 92.1%.

The given selection describes the precision and recall metrics for a Naive Bayes model used to classify breast cancer cases as either benign or malignant. The precision of 97.4% indicates that when the model predicts a case as "malignant" (class 4), it is correct about 97.4% of the time. Similarly, the recall of 92.1% suggests that the model is able to correctly identify about 92.1% of the actual "malignant" cases in the dataset. Overall, these metrics indicate that the model is effective in accurately predicting malignant cases and capturing all instances of the "malignant" class.

In summary, precision focuses on the accuracy of positive predictions, while recall measures the ability to correctly identify all positive instances. High precision indicates that the model has fewer false positives, and high recall indicates that the model captures a significant portion of the actual positive cases.

Discretize the Dataset Using Three Bins (Equal-Frequency)

"3 bins (equal frequency)" is a binning method that discretizes continuous data into a specified number of discrete intervals in order to better understand data distribution, reduce noise, and handle some nonlinear relationships. In this method, data is divided into three intervals, each containing approximately equal numbers of data points. This means that the number of data points in each interval is relatively uniform, making the data more comparable and interpretable.

The number of bins can be set to 3 initially. As "class" is the target variable and a categorical variable, it is excluded from the operation. Each feature, excluding "class", is subjected to equal-frequency binning. The resulting data summary is as follows:

```
# define the number of bins
num_bins <- 3

# Create a new data frame to store the discretized data
data_discretized <- data_cleaned

# Get the column names of continuous feature variables
continuous_columns <- colnames(data_cleaned)[colnames(data_cleaned) !
= "class"]

# Equal frequency binning for each continuous feature
for (col in continuous_columns) {
  data_discretized[[col]] <- cut(data_cleaned[[col]],
                                 breaks = num_bins,
                                 labels = FALSE,
                                 include.lowest = TRUE)
}
```

Code 1: Equal Frequency Binning with 3-Bins

A summary of the discretized data looks like this:

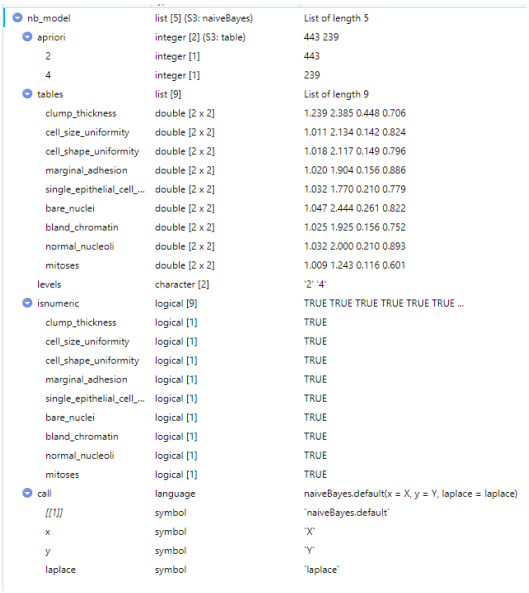
```
> summary(data_discretized)
 clump_thickness cell_size_uniformity cell_shape_uniformity marginal_adhesion single_epithelial_cell_size  bare_nuclei
 Min.   :1.000   Min.   :1.000        Min.   :1.000         Min.   :1.00      Min.   :1.00               Min.   :1.000
 1st Qu.:1.000   1st Qu.:1.000        1st Qu.:1.000         1st Qu.:1.00      1st Qu.:1.00               1st Qu.:1.000
 Median :1.000   Median :1.000        Median :1.000         Median :1.00      Median :1.00               Median :1.000
 Mean   :1.641   Mean   :1.405        Mean   :1.403         Mean   :1.33      Mean   :1.29               Mean   :1.537
 3rd Qu.:2.000   3rd Qu.:2.000        3rd Qu.:2.000         3rd Qu.:1.00      3rd Qu.:1.00               3rd Qu.:2.000
 Max.   :3.000   Max.   :3.000        Max.   :3.000         Max.   :3.00      Max.   :3.00               Max.   :3.000
 bland_chromatin normal_nucleoli    mitoses      class
 Min.   :1.00    Min.   :1.000   Min.   :1.000   2:443
 1st Qu.:1.00    1st Qu.:1.000   1st Qu.:1.000   4:239
 Median :1.00    Median :1.000   Median :1.000
 Mean   :1.34    Mean   :1.371   Mean   :1.091
 3rd Qu.:2.00    3rd Qu.:1.000   3rd Qu.:1.000
 Max.   :3.00    Max.   :3.000   Max.   :3.000
```

Figure 5: The Summary of the Discretized Data

The resulting dataset has a limited range of values, with each feature divided into three intervals of relatively uniform size. This makes the data easier to understand and interpret, as continuous numerical values are converted into discrete categories. Moreover, the distribution of each variable has changed, and the mean value has also been affected.

Discretization helps to better understand the characteristics of the data in the analysis and modeling process, and it reduces noise and handles some nonlinear relationships.

Build a New Naive Bayes Model by Using the Discretized Dataset



nb_model	list [5] (S3: naiveBayes)	List of length 5
apriori	integer [2] (S3: table)	443 239
2	integer [1]	443
4	integer [1]	239
tables	list [9]	List of length 9
clump_thickness	double [2 x 2]	1.239 2.385 0.448 0.706
cell_size_uniformity	double [2 x 2]	1.011 2.134 0.142 0.824
cell_shape_uniformity	double [2 x 2]	1.018 2.117 0.149 0.796
marginal_adhesion	double [2 x 2]	1.020 1.904 0.156 0.886
single_epithelial_cell_...	double [2 x 2]	1.032 1.770 0.210 0.779
bare_nuclei	double [2 x 2]	1.047 2.444 0.261 0.822
bland_chromatin	double [2 x 2]	1.025 1.925 0.156 0.752
normal_nucleoli	double [2 x 2]	1.032 2.000 0.210 0.893
mitoses	double [2 x 2]	1.009 1.243 0.116 0.601
levels	character [2]	'2' '4'
isnumeric	logical [9]	TRUE TRUE TRUE TRUE TRUE TRUE ...
clump_thickness	logical [1]	TRUE
cell_size_uniformity	logical [1]	TRUE
cell_shape_uniformity	logical [1]	TRUE
marginal_adhesion	logical [1]	TRUE
single_epithelial_cell_...	logical [1]	TRUE
bare_nuclei	logical [1]	TRUE
bland_chromatin	logical [1]	TRUE
normal_nucleoli	logical [1]	TRUE
mitoses	logical [1]	TRUE
call	language	naiveBayes.default(x = X, y = Y, laplace = laplace)
[[1]]	symbol	`naiveBayes.default`
x	symbol	`X`
y	symbol	`Y`
laplace	symbol	`laplace`

Figure 6: Details for the New Naïve Bayes Model

A New Naive Bayesian model is trained to predict benign and malignant with the information shown above.

```
call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        2         4
0.6495601 0.3504399

Conditional probabilities:
   clump_thickness
Y       [,1]     [,2]
  2 1.239278 0.4478116
  4 2.384937 0.7058750

   cell_size_uniformity
Y       [,1]     [,2]
  2 1.011287 0.1422474
  4 2.133891 0.8243038

   cell_shape_uniformity
Y       [,1]     [,2]
  2 1.018059 0.1493237
  4 2.117155 0.7957863

   marginal_adhesion
Y       [,1]     [,2]
  2 1.020316 0.1564391
  4 1.903766 0.8858994

   single_epithelial_cell_size
Y       [,1]     [,2]
  2 1.031603 0.2103518
  4 1.769874 0.7787706

   bare_nuclei
Y       [,1]     [,2]
  2 1.047404 0.2605447
  4 2.443515 0.8224463

   bland_chromatin
Y       [,1]     [,2]
  2 1.024831 0.1557848
  4 1.924686 0.7521520

   normal_nucleoli
Y       [,1]     [,2]
  2 1.031603 0.2103518
  4 2.000000 0.8934872

   mitoses
Y       [,1]     [,2]
  2 1.009029 0.1161591
  4 1.242678 0.6008561
```


Figure 7: Probabilities of the New Naive Bayes Model

For the interpretation of these probability data, please refer to the earlier part of this article for the mathematical interpretation of probability data for the first naive Bayes model. This will not be repeated here.

Confusion Matrix of the New Model by Using 5-fold Cross Validation

The confusion matrix of this model is shown below:

```
> print(confusion_matrix)
Confusion Matrix and Statistics

          Reference
Prediction   2   4
         2 416   8
         4  27 231

               Accuracy : 0.9487
                 95% CI : (0.9293, 0.964)
    No Information Rate : 0.6496
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8893

 Mcnemar's Test P-Value : 0.002346

            Sensitivity : 0.9391
            Specificity : 0.9665
         Pos Pred Value : 0.9811
         Neg Pred Value : 0.8953
             Prevalence : 0.6496
         Detection Rate : 0.6100
   Detection Prevalence : 0.6217
      Balanced Accuracy : 0.9528

       'Positive' Class : 2
```

In this confusion matrix example, there are two categories: category "2" (benign) and category "4" (malignant).

- Reference: Represents the actual category. The row labels are the actual categories, and the column labels are the predicted categories.
- Prediction: Represents the predicted category. The column labels are the predicted categories, and the row labels are the actual categories.

The values on the diagonal of the top left corner and bottom right corner of the matrix represent correct predictions by the model, while the values in other positions represent incorrect predictions. Specifically:

- True Positive (TP): The number of cases where the model correctly predicted "2" (benign) when the actual category was "2" is 416.
- True Negative (TN): The number of cases where the model correctly predicted "4" (malignant) when the actual category was "4" is 231.
- False Positive (FP): The number of cases where the model incorrectly predicted "2" (benign) when the actual category was "4" is 27.
- False Negative (FN): The number of cases where the model incorrectly predicted "4" (malignant) when the actual category was "2" is 8.

In addition, the confusion matrix provides a series of statistical indicators for evaluating model performance, including accuracy, recall, precision, specificity, etc.:

- Accuracy: The proportion of correctly predicted samples to the total number of samples, which is 0.9487. In this example, the model correctly predicted 94.87% of the samples overall.
- Sensitivity (recall): The proportion of correctly predicted category "2" (benign) samples to all actual category "2" samples, which is 0.9391. It measures the model's ability to identify benign samples.
- Specificity: The proportion of correctly predicted category "4" (malignant) samples to all actual category "4" samples, which is 0.9665. It measures the model's ability to identify malignant samples.
- Positive Predictive Value (precision): The proportion of correctly predicted category "2" (benign) samples to all predicted category "2" samples, which is 0.9811. It measures the accuracy of the model in predicting benign samples.
- Negative Predictive Value: The proportion of correctly predicted category "4" (malignant) samples to all predicted category "4" samples, which is 0.8953.

Explain How the New Model Makes Classification

feature	value
clump_thickness	3
cell_size_uniformity	3
cell_shape_uniformity	3
marginal_adhesion	3
single_epithelial_cell_size	2
bare_nuclei	3
bland_chromatin	3
normal_nucleoli	2
mitoses	1
class	4

Table 6: A Record from the Discretized Dataset

The goal is to classify the record into "2" (benign) or "4" (malignant) based on these attribute values.

First, we look at the prior probabilities of the model: prior probability of benign: 0.6495601 and prior probability of malignant: 0.3504399. Next, we calculate the conditional probabilities for each feature value corresponding to each class based on the conditional probability table. For example, for the case where the feature "clump_thickness" is 3: for class "2" (benign): the conditional probability is 1.239278 and for class "4" (malignant): the conditional probability is 2.384937. Similarly, we calculate the conditional probabilities of other feature values corresponding to different classes.

First, calculate the probability of belonging to category "2":

$$\begin{aligned} P(Y=2|data) &= P(Y=2) * P(clump_thickness = 3|Y=2) \\ &\quad * P(cell_size_uniformity = 3|Y=2) * \dots \\ &= 0.6495601 * 1.239278 * 1.011287 * 1.018059 * 1.020316 * 1.031603 \\ &\quad * 1.024831 * 1.031603 * 1.009029 \\ &\approx 0.1256317 \end{aligned}$$

This time it's ok as we have categorical data 3/3

Then Calculate the probability of belong to category "4":

$$\begin{aligned} P(Y=4|data) &= P(Y=4) * P(clump\_thickness = 3|Y=4) * P(cell\_size\_uniformity \\ &= 3|Y=4) * \dots \\ &= 0.3504399 * 2.384937 * 2.133891 * 2.117155 * 1.903766 \\ &\quad * 2.443515 * 1.924686 * 2.000000 * 1.242678 \\ &\approx 0.1074146 \end{aligned}$$

As P(Y=2|data) > P(Y=4|data), the model will predict that the data record belongs to category "2" (benign).

The given example data demonstrates well how the Naive Bayes model calculates the probability of data belonging to categories "2" and "4", and then selects the category with the highest probability as the final prediction result. However, it is unfortunate that in this example, the model incorrectly predicted that the data record belongs to category "2" (benign), while in reality the original data classified the data as category "4" (malignant).

Conclusion

In conclusion, the provided analysis showcased the utilization of a Naïve Bayes model to classify breast cancer cases as either benign or malignant. By discretizing the dataset into three bins using equal-frequency binning, the data's distribution was transformed for better interpretability and handling of non-linear relationships. The Naïve Bayes model's performance was evaluated using both 5-fold and 10-fold cross-validation techniques.

The confusion matrix presented the model's classification accuracy and various performance metrics, including sensitivity, specificity, positive predictive value, negative predictive value, and more. Precision and recall were explained in detail as metrics that assess the accuracy of positive predictions and the model's ability to identify all instances of a specific class, respectively.

Furthermore, the practical application of the model's classification process was elucidated using a specific data record. The process involved calculating the probability of the record belonging to each class and predicting the class with the highest probability.

In essence, this analysis highlighted the Naïve Bayes model's effectiveness in classifying breast cancer cases while emphasizing the significance of precision and recall metrics in evaluating its performasddnce. However, it is essential to consider false positives and false negatives in medical contexts, as misclassifications can have critical consequences.