Predicting Restaurant Health Inspection Grades

Ipek Sayar

10/21/2024

Table of Contents

Project Goal

The main goal of this project is to classify the GRADE field and predict which restaurants are safe to eat at based on different factors. This is important for both consumers looking for safe dining options and restaurant owners who want to improve their health inspection results. By figuring out which attributes lead to a better grade, this project can help people make healthier dining choices and encourage restaurants to maintain higher food safety standards.

This project focuses on food safety, which is a big deal for everyone. With more people concerned about what they eat, having a reliable way to assess restaurant safety can help consumers choose where to dine with confidence. The project has a direct impact on community health. Foodborne illnesses can be serious, so giving consumers clear information about restaurant safety can help prevent risks and build trust in food places.

Also, the findings from this project can help restaurant owners by pointing out specific areas they need to work on. By understanding which factors lead to better inspection grades, they can focus on improving their practices and providing safer food. This two-way benefit—helping consumers and supporting restaurant operators—makes this project both interesting and relevant in today's food scene.

Introducing the Dataset

Instances	103,426
Attributes	26

Attribute	**Description**
CAMIS	Unique Identifying Number for each restaurant.
DBA	Business Name.
BORO	Name of the Borough.
BUILDING	Building Number.
STREET	Street Name.
ZIPCODE	The zip code of the restaurant.

PHONE	The contact number of the restaurant.
CUISINE DESCRIPTION	The specializations of the restaurant.
INSPECTION DATE	The date that the inspection occurred.
ACTION	Steps taken after the inspection.
VIOLATION CODE	The code given to that specific violation.
VIOLATION DESCRIPTION	The description of the violation.
CRITICAL FLAG	Whether the restaurant is in critical condition or not.
SCORE	The score given to the restaurant during the inspection.
GRADE DATE	The date that the grade was given.
RECORD DATE	The date that the grade was recorded in the database.
INSPECTION TYPE	The type of inspection that was conducted.
Latitude	The latitude location of the restaurant.
Longitude	The longitude of the location of the restaurant.
Community Board	The local administrative district that handles neighborhood matters, potentially impacting restaurant regulations and health inspections.
Council District	The political division represented by a council member, which may influence local policies affecting restaurants.
Census Tract	A small, relatively permanent statistical subdivision of a county used for demographic data collection.
BIN	Building Identification number.
BBL	Parcel numbers that identify the location of buildings and its properties
NTA	Neighborhood Tabulation Area, a statistical geographic unit used to analyze neighborhoods within a city

Class	Description
GRADE	The grade that the specific score given to the restaurant is associated with.

Class Proportions

A	68%
B	15%
C	17%

The data distribution shows that the score and grade attributes are right-skewed, meaning most restaurants tend to perform well on health inspections, with fewer establishments receiving poorer grades. The remaining features consist of categorical data, which are uniformly distributed, suggesting a balanced representation of different categories such as cuisine type, location, or establishment type.

Preprocessing

Part 1

Handling Missing Class Values
Approximately 40% of the "Grade" values were missing in the dataset. To address this, the "Score" column, which contained precise health inspection scores ranging from 0 to 70 with no missing entries, was used to calculate the missing grades. According to the NYC Health guidelines, scores from 0-13 correspond to an A grade, 14-27 to a B, and 28 or higher to a C. Using Excel's IF-THEN statements, missing grades were filled based on the scores. Once this was done, the "Score" column was removed to prevent the model from learning to predict the grade solely from this column, which would diminish its real-world utility where the score may not always be available.

Removing Irrelevant or Redundant Columns
The dataset contained columns like CAMI (restaurant identifier) and Phone Number, which were unique to each instance and did not provide meaningful patterns for the model to learn from. These columns were removed, leaving only attributes relevant for predicting the health inspection grade. After removing the "Score" column and unique identifiers, the dataset now included 23 attributes for further refinement.

Addressing Missing and Inconsistent Data
Around 20% of the "Grade Date" values were missing, which were replaced with a placeholder date (1/1/2024) for easy identification during future analysis.

Compatibility with Weka
To ensure compatibility with Weka, punctuation was removed using Excel's "Find and Replace" function. Some values were also portrayed as Strings in the Weka interfaces. So, the filter StringToNominal was used in order to not cause problems with the attribute selection and classifier algorithms.

Sampling
Since the original dataset contains a large number of instances, reducing the dataset size was necessary to make computations faster and more efficient. Python code was used to conduct stratified sampling and reduce the number of instances to 1500. Stratified sampling ensures that class proportions are preserved and allows us to have an accurate representation of the original population.

<u>Part 2</u>
Came back after getting poor results.

Dealing with Class Imbalances:
Various classifiers attempted to predict health inspection grades in restaurants with extremely unbalanced class proportions. This resulted in poor outcomes (50%-60% accuracy, ~0.68 precision and recall), with classifiers consistently predicting the more abundant class (the A grade). Although the usual approach to fix this would involve evaluating classifiers based on metrics like precision, recall, or the ROC curve, these metrics indicate poor performance without addressing the underlying problem. Analyzing performance metrics alone did not yield significant improvements.

Weka's SMOTE filter was used to slightly better this situation. It generates synthetic instances by interpolating between existing minority instances which helps to maintain the underlying patterns in the data. It must be kept in mind that generating too many synthetic samples can make the model too focused on synthetic data rather than real data. Due to this, the Class B's instances were only increased about 500 using 122.22% ($\frac{500-225}{225} * 100\%$) and Class C's instances were increased to 600 using 135.29% ($\frac{600-255}{255} * 100\%$).

Using SMOTE slightly changes the class proportions, but this adjustment is not a significant concern because the goal is to predict health inspection grades based on various attributes. While more restaurants often receive an A on their inspections, this should not matter as much when predicting the specific grade a restaurant will receive before the inspection based on attribute values. SMOTE helps improve the balance of the dataset, allowing the model to learn more effectively from the minority class and reducing bias toward the majority class which improves the overall prediction performance.

Dataset After Preprocessing:

Part 1:

Instances	1500
Attributes	23

Attribute	Description
DBA	Business Name.
BORO	Name of the Borough.
BUILDING	Building Number.
STREET	Street Name.
ZIPCODE	The zip code of the restaurant.
CUISINE DESCRIPTION	The specializations of the restaurant.
INSPECTION DATE	The date that the inspection occurred.
ACTION	Steps taken after the inspection.
VIOLATION CODE	The code given to that specific violation.
VIOLATION DESCRIPTION	The description of the violation.
CRITICAL FLAG	Whether the restaurant is in critical condition or not.
GRADE DATE	The date that the grade was given.
RECORD DATE	The date that the grade was recorded in the database.
INSPECTION TYPE	The type of inspection that was conducted.
Latitude	The latitude location of the restaurant.
Longitude	The longitude of the location of the restaurant.
Community	The local administrative district that handles neighborhood matters,

Board	potentially impacting restaurant regulations and health inspections.
Council District	The political division represented by a council member, which may influence local policies affecting restaurants.
Census Tract	A small, relatively permanent statistical subdivision of a county used for demographic data collection.
BIN	Building Identification number.
BBL	Parcel numbers that identify the location of buildings and its properties
NTA	Neighborhood Tabulation Area, a statistical geographic unit used to analyze neighborhoods within a city

Class	**Description**
GRADE	The grade that the specific score given to the restaurant is associated with.

Class Proportions

A	68%
B	15%
C	17%

<u>Part 2</u>

Instances	2118
Attributes	23

Class Proportions

A	48%
B	25%
C	27%

Attribute Selection Algorithms and Classifier models:

Since the dataset is small enough, K-fold Cross Validation is used instead of a training, validation, and testing set.

Classifier methods

OneR

A simple algorithm that generates one set of rules to classify data based on a single attribute. It works by evaluating all the attributes in the dataset and selecting the one that produces the best results in terms of accuracy. For each value of this selected attribute, OneR assigns the most frequent class label, thus creating simple if-then rules. The process involves calculating error rates for each attribute and choosing the attribute with the lowest error rate as the basis for classification. Despite its simplicity, OneR can often produce surprisingly accurate models, especially in cases where the data is straightforward or one attribute is significantly more informative than the others. Its simplicity also makes it fast and interpretable.

Random Tree

A type of decision tree algorithm where the model builds multiple decision trees using random subsets of features at each node. Unlike a standard decision tree, which evaluates all attributes to find the best split, Random Tree picks a random subset of attributes, reducing the chance of overfitting. This randomness helps the model generalize better to new data, as it reduces the risk of creating overly complex decision rules that fit only the training data. During the classification process, the Random Tree predicts by passing the instance through the tree's branches based on its attribute values, at the end reaching a leaf node that represents the predicted.

Decision Table

A tabular representation that shows combinations of attribute values and their corresponding class labels. To build this table, the algorithm evaluates subsets of attributes to determine which ones are most important for classification. It then creates rules based on these combinations of attribute values. The decision table simplifies the classification problem by focusing only on the most relevant features and reducing the complexity of the model. Decision Tables are also interpretable, as they provide an easy-to-understand set of rules for classification.

NaiveBayes

A probabilistic classification algorithm based on Bayes' Theorem, which assumes that all attributes are independent of each other given the class label. Although this independence assumption is often unrealistic, Naive Bayes still works surprisingly well for many real-world problems. The algorithm calculates the probability of each class based on the given input attributes, and then predicts the class with the highest probability. Naive Bayes is particularly effective when the dataset has a large number of features or when some attributes are more informative than others. Its simplicity and efficiency make it a good choice for some datasets, but it may struggle with very complex attribute interactions due to its independence assumptions.

My selection:

Attribute	Reason
DBA	The specific restaurant chain may have different health standards. Some chains are known for promoting healthier practices and stricter cleanliness policies, while others might cut corners.
Building	Some buildings have more than one restaurant in them. Restaurants within the same building may share sanitation standards or infrastructure issues (plumbing, ventilation), which can impact overall cleanliness. Some buildings may enforce their own cleanliness regulations, while others may have more relaxed or absent standards.
Street	Higher-end or upscale streets may enforce stricter health standards or attract restaurants that naturally follow better practices. Alternatively, high-traffic areas may result in more wear and tear leading to issues.
Cuisine Description	Certain cuisines may require more specialized preparation and handling, which could affect the likelihood of violations (e.g., sushi or raw foods needing specific storage temperatures). The handling of culturally significant food might also indicate whether extra care is required to prevent contamination.
Action	The type of corrective action can reflect the severity of issues found. Also, the actions taken after the inspection give direct insight into how well the restaurant manages its cleanliness and whether violations are being addressed properly.
Violation code	The violation code specifies which rules were broken, giving direct insight into the nature of the health risks at a restaurant.
Violation description	A violation code isn't always specific enough, various different violations could fall under the same violation code, violation description allows us to go more in depth.
Critical flag	This attribute shows whether any of the violations are severe enough to pose immediate health risks. A critical flag typically leads to lower grades, as it reflects poor sanitary conditions or imminent hazards.
Longitude	The location could influence inspection outcomes due to local regulations or environmental factors like weather.
Latitude	The location could influence inspection outcomes due to local regulations or environmental factors like weather.
Community board	Different boards may have varying degrees of strictness or

	resources for enforcing health and safety regulations.
Council district	Council members can shape regulations, allocate resources for health inspections, or prioritize cleanliness in their district.

OneR

```
   a    b    c   <-- classified as
1007    5    8 |    a = A
 154  344    1 |    b = B
 171    2  426 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
83.8999 %	0.839	0.145	0.872	0.839	0.847

Random Tree

```
  a   b   c   <-- classified as
978  22  20 |   a = A
158 338   3 |   b = B
168   4 427 |   c = C
```

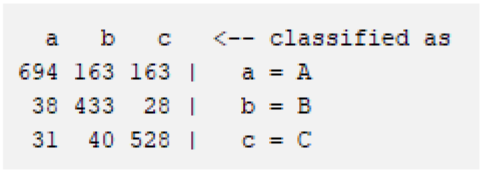
Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
82.2946 %	0.823	0.151	0.848	0.823	0.872

Decision Table

```
   a    b    c   <-- classified as
1011    4    5 |    a = A
 192  306    1 |    b = B
 202    1  396 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
80.8782 %	0.809	0.175	0.857	0.809	0.853

NaiveBayes



```
  a   b   c   <-- classified as
694 163 163 |   a = A
 38 433  28 |   b = B
 31  40 528 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
78.1398 %	0.781	0.095	0.806	0.781	0.947

InfoGainAttributeEval:

InfoGainAttributeEval works by looking at how much knowing the value of an attribute helps us understand the target class. It calculates how much uncertainty, or entropy, is reduced about the class when we know the value of that attribute. If knowing an attribute makes it much clearer what the class is, it gets a high score; if it doesn't help much, it gets a low score. **Attributes with an information gain above 0.5 were selected.**

Weka takes care of the whole process but if we were to do it by hand, the following is what we'd need to know.

Entropy Calculation

$$H(D) = -\sum_{i=1}^{n} p_i log_2(p_i)$$

p_i is the proportion of instances in class.

Information Gain Calculation

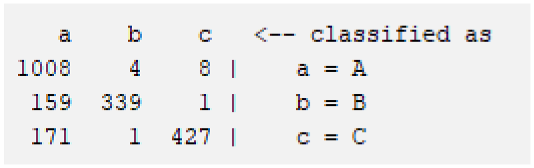
$$IG(A) = H(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} H(D_v)$$

H(D) is the entropy of the entire dataset D
v is a value of attribute A
D_v is the subset of A where A=v
$\frac{|D_v|}{|D|}$ is the proportion of the data where A=v
$H(D_v)$ is the entropy of the subset D_v.

1.2051	Latitude
1.2051	Longitude
1.1905	BBL
1.1861	DBA
1.1711	BIN
0.9961	BUILDING

0.6694	GRADE DATE
0.5982	INSPECTION DATE
0.564	Census Tract
0.5568	STREET
0.2338	VIOLATION DESCRIPTION
0.2061	NTA
0.1501	VIOLATION CODE
0.1449	INSPECTION TYPE
0.1388	ACTION
0.1124	CUISINE DESCRIPTION
0.0828	Council District
0.0777	Community Board
0.0267	CRITICAL FLAG
0.0105	BORO
0	RECORD DATE
0	ZIPCODE

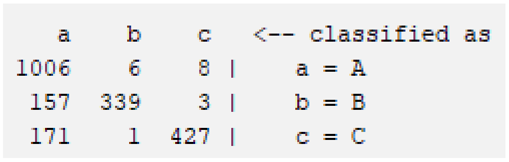
OneR



```
   a    b    c   <-- classified as
1008    4    8 |    a = A
 159  339    1 |    b = B
 171    1  427 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
83.7583 %	0.838	0.147	0.872	0.838	0.845

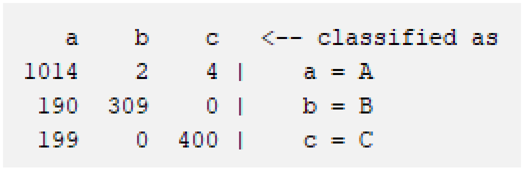
Random Tree



```
   a    b    c   <-- classified as
1006    6    8 |    a = A
 157  339    3 |    b = B
 171    1  427 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
83.6638 %	0.837	0.147	0.870	0.837	0.873

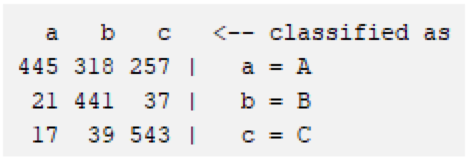
Decision Table



```
   a    b    c   <-- classified as
1014    2    4 |    a = A
 190  309    0 |    b = B
 199    0  400 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
81.3503 %	0.814	0.172	0.862	0.814	0.856

NaiveBayes



```
  a   b   c   <-- classified as
445 318 257 |   a = A
 21 441  37 |   b = B
 17  39 543 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
67.4693 %	0.675	0.123	0.757	0.675	0.938

GainRatioAttributeEval:

GainRatioAttributeEval works similarly to InfoGainAttributeEval but adds a step to account for the number of possible values an attribute can have. In other words, it's an enhancement of the InfoGain, with a normalized score. It measures how much knowing the value of an attribute reduces uncertainty about the class, just like information gain, but then it adjusts that value to prevent attributes with many possible values from being unfairly favored. This way, it helps find attributes that are not only informative but also balanced, making it more reliable for feature selection. **Attributes with an Information Gain above 0.25 were chosen.**

Split Information

Measures the potential information generated by splitting the dataset into multiple subsets based on attribute A. It's essentially the entropy of the split itself.

$$SI(A) = -\sum_{v \in Values(A)} \frac{|D_v|}{|D|} log_2 \left(\frac{|D_v|}{|D|}\right)$$

$|D_v|$ is the number of instances where A = v
|D| is the total number of instances

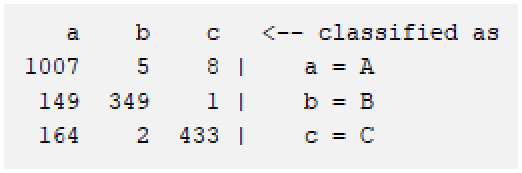
Gain Ratio

$$GR(A) = \frac{IG(A)}{SI(A)}$$

0.28012	ACTION
0.11639	DBA
0.11555	Longitude
0.11555	Latitude
0.11441	BBL
0.11273	BIN
0.10062	BUILDING
0.08119	GRADE DATE
0.07668	INSPECTION DATE
0.06548	STREET
0.0649	Census Tract
0.0447	VIOLATION DESCRIPTION
0.03475	VIOLATION CODE
0.03022	NTA
0.02594	CRITICAL FLAG
0.02348	CUISINE DESCRIPTION
0.0156	Council District
0.01392	Community Board

0.00513	BORO
0	ZIPCODE
0	RECORD DATE

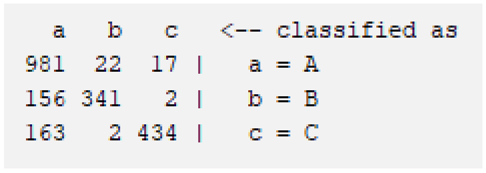
OneR



```
   a    b    c   <-- classified as
1007    5    8 |    a = A
 149  349    1 |    b = B
 164    2  433 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
84.4665 %	0.845	0.140	0.875	0.845	0.852

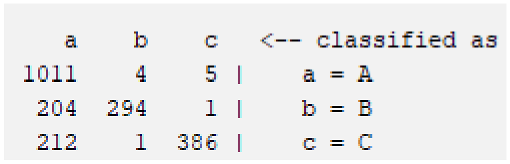
Random Tree



```
  a   b   c   <-- classified as
981  22  17 |   a = A
156 341   2 |   b = B
163   2 434 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
82.9084 %	0.829	0.147	0.854	0.829	0.875

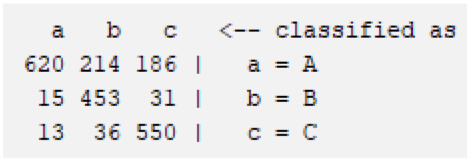
Decision Table



```
   a    b    c   <-- classified as
1011    4    5 |    a = A
 204  294    1 |    b = B
 212    1  386 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
79.8395 %	0.798	0.184	0.851	0.798	0.855

NaiveBayes



```
  a   b   c   <-- classified as
620 214 186 |   a = A
 15 453  31 |   b = B
 13  36 550 |   c = C
```

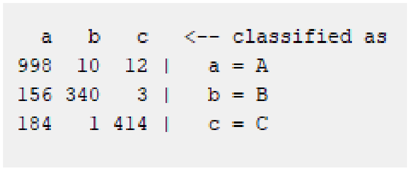
Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
76.6289 %	0.766	0.089	0.815	0.766	0.964

ClassiferAttributeEval:
ClassifierAttributeEval assesses how well each attribute helps a specific classifier make predictions. It uses the classifier's performance, like accuracy or error rate, to determine the importance of the attributes. This means it evaluates attributes based on how they improve the classifier's ability to correctly predict the target class, focusing on their practical impact rather than just statistical measures. **The best 17 attributes were chosen.**

NTA
ACTION
VIOLATION CODE
CUISINE DESCRIPTION
INSPECTION DATE
ZIPCODE
BBL
BORO
BUILDING
STREET
VIOLATION DESCRIPTION
CRITICAL FLAG
GRADE DATE
Census Tract
BIN
RECORD DATE

Council District
Community Board
Longitude
Latitude
INSPECTION TYPE
DBA

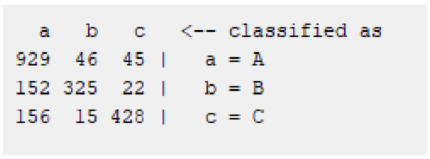
OneR



```
  a   b   c   <-- classified as
998  10  12 |   a = A
156 340   3 |   b = B
184   1 414 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
82.7195 %	0.827	0.154	0.860	0.827	0.837

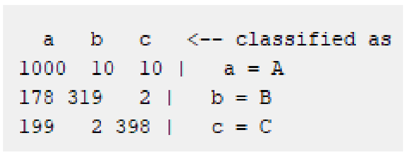
Random Tree



```
  a   b   c   <-- classified as
929  46  45 |   a = A
152 325  22 |   b = B
156  15 428 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
79.4145 %	0.794	0.156	0.805	0.794	0.855

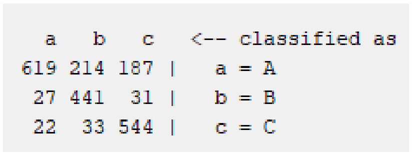
Decision Table



```
   a   b   c   <-- classified as
1000  10  10 |   a = A
 178 319   2 |   b = B
 199   2 398 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
81.067 %	0.811	0.169	0.851	0.811	0.843

NaiveBayes



```
  a   b   c   <-- classified as
619 214 187 |   a = A
 27 441  31 |   b = B
 22  33 544 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
75.7318 %	0.757	0.098	0.799	0.757	0.948

CorrelationAttributeEval:

CorrelationAttributeEval measures the relationship between each attribute and the target class by calculating the correlation coefficient. It looks at how changes in the attribute values relate to changes in the class values; a strong correlation indicates that the attribute is likely helpful for prediction. This method helps identify attributes that have a meaningful relationship with the class, making them more valuable for building a model. **Attributes with correlation 0.025 or higher were chosen.**

The correlation coefficient is calculated by dividing the covariance of x and y by the variance of x multiplied by the variance of y.

$$r_{xy} = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \, \Sigma(y_i - \bar{y})^2}}$$

x_i is an instance of x in the dataset

$\bar{x}$ is the x's mean

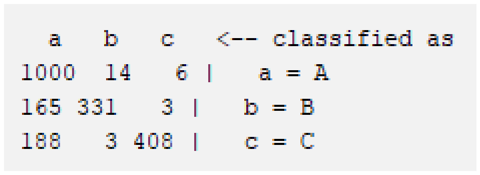
y_i is an instance of y in the dataset

$\bar{y}$ is the y's mean

0.2869	ACTION
0.2457	INSPECTION TYPE
0.1438	CRITICAL FLAG
0.0808	VIOLATION CODE
0.0802	GRADE DATE

0.0669	VIOLATION DESCRIPTION
0.0458	CUISINE DESCRIPTION
0.0361	ZIPCODE
0.0333	BORO
0.0327	Council District
0.0293	Community Board
0.0255	DBA
0.0255	NTA
0.0249	INSPECTION DATE
0.0235	BUILDING
0.0232	BBL
0.0232	Longitude
0.0232	Latitude
0.0232	BIN
0.0225	Census Tract
0.0223	STREET
0	RECORD DATE

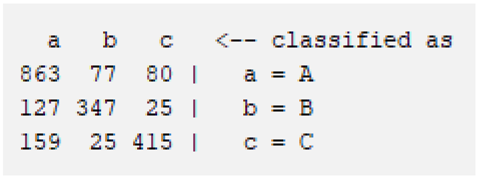
OneR



```
  a   b   c   <-- classified as
1000  14   6 |   a = A
165 331   3 |   b = B
188   3 408 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
82.1058 %	0.821	0.159	0.857	0.821	0.831

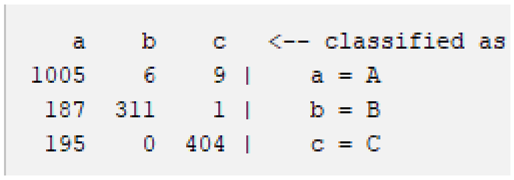
Random Tree



```
  a   b   c   <-- classified as
863  77  80 |   a = A
127 347  25 |   b = B
159  25 415 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
76.7233 %	0.767	0.160	0.769	0.767	0.852

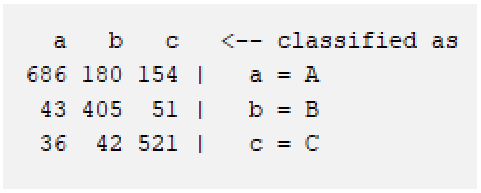
Decision Table



```
    a    b    c   <-- classified as
 1005    6    9 |    a = A
  187  311    1 |    b = B
  195    0  404 |    c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
81.2087 %	0.812	0.170	0.856	0.812	0.849

NaiveBayes



```
   a   b   c   <-- classified as
 686 180 154 |   a = A
  43 405  51 |   b = B
  36  42 521 |   c = C
```

Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
76.1095 %	0.761	0.105	0.787	0.761	0.928

Analysis

Top 5 values in each are highlighted green.

AttributeSelect Classifier	Accuracy	TP Rate	FP Rate	Precision	Recall	ROC Area
My Selection OneR	83.8999 %	0.839	0.145	0.872	0.839	0.847
My Selection	82.2946 %	0.823	0.151	0.848	0.823	0.872

Random Tree						
My Selection Decision Table	80.8782 %	0.809	0.175	0.857	0.809	0.853
My Selection NaiveBayes	78.1398 %	0.781	0.095	0.806	0.781	0.947
InfoGain OneR	83.7583 %	0.838	0.147	0.872	0.838	0.845
InfoGain Random Tree	83.6638 %	0.837	0.147	0.870	0.837	0.873
InfoGain Decision Table	81.3503 %	0.814	0.172	0.862	0.814	0.856
InfoGain NaiveBayes	67.4693 %	0.675	0.140	0.757	0.675	0.938
GainRatio OneR	84.4665 %	0.845	0.123	0.875	0.845	0.852
GainRatio Random Tree	82.9084 %	0.829	0.147	0.854	0.829	0.875
GainRatio Decision Table	79.8395 %	0.798	0.184	0.851	0.798	0.855
GainRatio NaiveBayes	76.6289 %	0.766	0.089	0.815	0.766	0.964
ClassifierAttri OneR	82.7195 %	0.827	0.154	0.860	0.827	0.837
ClassifierAttri Random Tree	79.4145 %	0.794	0.156	0.805	0.794	0.855
ClassifierAttri Decision Table	81.067 %	0.811	0.169	0.851	0.811	0.843
ClassifierAttri NaiveBayes	75.7318 %	0.757	0.098	0.799	0.757	0.948
Correlation OneR	82.1058 %	0.821	0.159	0.857	0.821	0.831
Correlation Random Tree	76.7233 %	0.767	0.160	0.769	0.767	0.852

Correlation Decision Table	81.2087 %	0.812	0.170	0.856	0.812	0.849
Correlation NaiveBayes	76.1095 %	0.761	0.105	0.787	0.761	0.928

Accuracy Trends

Across the different classifiers, OneR consistently performs well, with accuracy ranging between 82% and 84.5%. Among these, GainRatio OneR shows the highest accuracy at 84.47%, making it the most reliable in terms of predicting the correct outcome. Random Tree and Decision Table perform decently, achieving accuracy in the low 80% range. On the other hand, NaiveBayes shows significantly weaker accuracy, especially under the InfoGain evaluator, where its accuracy drops to 67.47%. This indicates that NaiveBayes might struggle with correctly classifying instances, especially with unbalanced datasets such as this one.

Precision and Recall

OneR also excels in both precision and recall, which are critical in ensuring not just accurate but also comprehensive predictions. Under GainRatio, OneR achieves 0.875 for precision, meaning it consistently makes accurate positive predictions. Its recall is also strong at 0.845, indicating it identifies a large proportion of positive instances. Random Tree follows closely behind OneR in these metrics across most evaluators, making it a solid alternative. In contrast, NaiveBayes underperforms in recall, suggesting it misses more positive instances, especially under InfoGain, it drops to 0.675.

False Positive (FP) Rate

When considering the False Positive Rate, NaiveBayes performs exceptionally well, maintaining low FP rates across the board. Under GainRatio, its FP rate is particularly low at 0.089, indicating it is unlikely to misclassify instances as positive when they are not. OneR also shows a strong FP rate, particularly under GainRatio, with 0.123, meaning it too is effective at avoiding false positives. While OneR has a slightly higher FP rate than NaiveBayes, it still maintains solid performance in this area, ensuring the classifier does not misclassify too frequently.

ROC Area

NaiveBayes stands out in terms of ROC Area, which measures the model's ability to distinguish between the classes. It consistently has the highest ROC Area values, particularly under GainRatio, where it scores 0.964, suggesting it is excellent at separating positive from negative cases. However, despite its strength in the ROC Area, NaiveBayes does not perform as well in terms of accuracy and recall, meaning its predictions, while ranked correctly, may still be incorrect more often than those of other classifiers.

Best Classification Model

GainRatio OneR is the best classifier. It leads in accuracy at 84.47%, and its precision of 0.875 and recall of 0.845 make it an excellent choice for balancing correct and comprehensive predictions. Its FP rate of 0.123 is also low, reducing the risk of misclassifications. While its ROC Area isn't the highest, its overall performance makes it a more reliable and well-rounded choice compared to other classifiers. In short, GainRatio OneR offers the best combination of critical metrics, making it the best classifier for predicting health inspection grades.

Reproduce the best model

1. Download the NYC Restaurant Health Inspections csv file.
2. Open the Python file and run it in the same directory as the csv file.
3. Open Weka and load the created cleaned_inspections.csv.
4. Under the PreProcess tab, click Filter > Choose > filters > unsupervised > attribute, select StringtoNominal.
5. Click on the white space and make sure it applies to all of the attributes, front the first one to the last one. Hit Apply.
6. Under the PreProcess tab, click Filter > Choose > filters > supervised> instance, select SMOTE.
7. Click on the white space and then set the index to 2 and set the percentage to 122.22%
8. Repeat steps 6 and 7 with an index of 3 and a percentage of 135.29%.
9. Go to the “Select attributes” tab and choose the correct class – Grade.
10. Select GainRatioAttributeEval as Attribute Evaluator and Ranker as Search Method
11. Click Start.
12. Pick out the attributes that have a ratio of 0.25 or more. Remove the other attributes using Weka’s Remove.
13. Save this dataset as an arff.
14. Click on the Classify tab and click “Cross Validation” and select what should be default 10 folds.
15. Select the OneR model under rules.
16. Click Start.

Conclusion

This lab has been a valuable learning experience in several ways. First, it gave me hands-on practice with Weka, a tool I had never used before. As I explored its features, I became more comfortable using it to apply machine learning algorithms and solve problems like missing values, converting between different data types, or applying filters in general to class imbalances. Now, I feel confident navigating Weka and experimenting with different models.

This lab has also introduced me to good sources for finding useful datasets, which will be important for future projects, both school and personal. Learning how to search for and assess the quality of datasets is a critical skill because choosing the right data is key to building successful models. It might have been a little hard to get Weka to accept the dataset at first but, Weka makes it easy to explore various datasets.

Additionally, this experience helped me understand the general process of creating classification models. I learned about the importance of data preparation, testing different classifiers, and evaluating their performance. Even though this was my first time building a model from scratch, I now have a solid grasp of the steps involved and feel capable of applying these methods to new datasets. This means I can use what I've learned to predict outcomes in different areas that interest me, and I don't have to be worried about not knowing how to do it anymore.

Now that I've been through this process, I know how to approach future projects with a systematic mindset. This gives me the freedom to explore and predict a wide range of outcomes I've always been curious about. Also, since this was a classification project, I think I would like doing something similar with a regression project next time. I would like to explore the differences between the two.

Files Attached with the Report:

Drive Folder
https://drive.google.com/drive/folders/1UlBlW0yRi4AU9tFH2jlHlxMXu--GZyQB?usp=drive_link

Original Dataset
https://www.kaggle.com/datasets/babatundezenith/food-inspection-data
Dr. Yilmaz mentioned that I could find a dataset on Kaggle if mine was too complex because it had some unusual values and attributes. I liked my original idea, so I didn't want to change it much. However, I ended up switching to another dataset, which still focuses on restaurant health inspections but had better and more understandable attributes. This new dataset also has a lot of missing values and other issues that I needed to address, so it wasn't an already clean dataset and still required a lot of preprocessing.

Python Code
https://drive.google.com/file/d/15gf3skn32oyElLqb4-vtVqF6P8DK2ZHy/view?usp=sharing
- Formats the dataset (Moves Grade attribute to the end, indicating that it is the class)
- Fills in the missing values in the Grade Date attribute
- Preforms Stratified Sampling

Final Dataset before Attribute Selection:
https://drive.google.com/file/d/1lT3R4Hj_eL1k7EArelDn4W3_HlkcU1Bg/view?usp=sharing

All of the Datasets after Attribute Selection
- **MySelections**

- **InfoGainAttributeEval**
- **GainRatioAttributeEval**
- **ClassiferAttributeEval**
- **CorrelationAttributeEval**

References

Weka Attribute Selection Algorithms
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/GainRatioAttributeEval.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/ClassifierAttributeEval.html
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/CorrelationAttributeEval.html
https://www.bmj.com/about-bmj/resources-readers/publications/statistics-square-one/11-correlation-and-regression

Classifier Algorithms
https://www.geeksforgeeks.org/learn-one-rule-algorithm/
https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html#:~:text=Random%20Forest%20Trees%20(RFT)%20is,of%20several%20independent%20base%20models.
https://www.geeksforgeeks.org/software-engineering-decision-table/
https://www.ibm.com/topics/naive-bayes#:~:text=Na%C3%AFve%20Bayes%20is%20part%20of,important%20to%20differentiate%20between%20classes.
https://www.geeksforgeeks.org/naive-bayes-classifiers/