1. Show examples from DBSCAN and EM Models in code
2. Finish filling out clustering grid from last time

##

STARTER

Here is a small dataset showing the weather and whether Tiger Woods plays golf

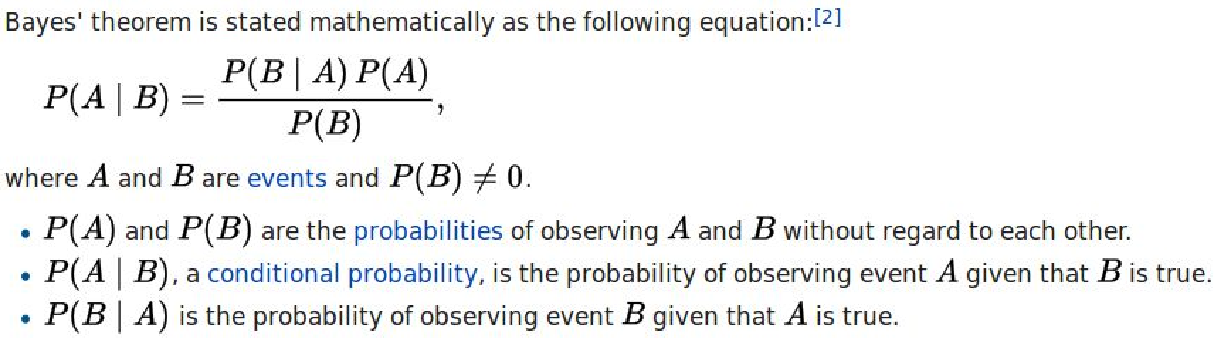
Given this data estimate the probability he would play if it was rainy day with high humidity and windy weather?

How did you think about the problem?

##

Naive Bayes

1. For each item, calculate probability he plays, multiply them together, including the probability he plays at all
2. Use Bayes Rule:
    a. Does anyone remember it from yesterday:



Bayes' theorem is stated mathematically as the following equation:[2]

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

where A and B are events and $P(B) \neq 0$.

- $P(A)$ and $P(B)$ are the probabilities of observing A and B without regard to each other.
- $P(A \mid B)$, a conditional probability, is the probability of observing event A given that B is true.
- $P(B \mid A)$ is the probability of observing event B given that A is true.

3. Example: P(Yes|Sunny) = P(Sunny|Yes) * P(Yes) / P(Sunny)
    a. Have students calculate
    b. P(Yes|Sunny) = .6
4. Build a frequency table:

Weather	No	Yes
Overcast		4
Rain	3	2
Sunny	2	3
Total	5	9

5. Example: P(Yes|Rainy) = P(Rainy|Yes) * P(Yes) / P(Rainy)
   a. (2/9) * (9/14) / (5/14) = .4
   b. P(Yes|Rainy) = 0.4
6. Example: P(Yes|Windy) = P(Windy|Yes) * P(Yes) / P(Windy)

Wind	No Golf	Yes Golf
Yes	3	3
No	2	6
Total	5	9

   a. (3/9) * (9/14) / (6/14) = .5
   b. P(Yes|Windy) = .5
7. Example: P(No|Windy) = P(Windy|No) * P(No) / P(Windy)
   a. (3/5) * (5/14) / (6/14) = .5
   b. P(No|Windy) = .5
8. What about if multiple: Rain, High Humidity, Weak winds?
   a. P(Yes|Rain, High Humidity, Weak Winds) = P(Rain|Yes)*P(HighHumidity|Yes)*P(WeakWinds|Yes)*P(Yes)
   b. (2/9)*(3/9)*(6/9)*(9/14) = 0.031746032
   c. P(No|Rain, High Humidity, Weak Winds) = P(Rain|No)*P(HighHumidity|No)*P(WeakWinds|No)*P(No)
   d. (3/5)*(4/5)*(2/5)*(5/14) = 0.068571429
   e. NORMALIZE:
      i. P(Yes|Rain, High Humidity, Weak Winds) = 0.0317/ (0.0317 + 0.0685) = 31.6%
      ii. P(No|Rain, High Humidity, Weak Winds) = 0.0685/ (0.0317 + 0.0685) = 68.4%
9. Pros:
   a. It is easy and fast to predict class of test data set. It also perform well in multi class prediction
   b. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
   c. It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).
10. Cons:
   a. If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we

can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.

   b. On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
   c. Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

11. 4 Applications of Naive Bayes Algorithms
    a. Real time Prediction: Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
    b. Multi class Prediction: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
    c. Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
    d. Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

##

Decision Tree algorithms

1. try to understand the system
   a. split into subsets
   b. For each subset
      i. Is it pure? (all yes or no)
         1. if yes, stop
         2. if not split into subsets again
2. **Everyone try it!**
3. Start with outlook, create a decision tree
4. Follow decision tree: Rain High Weak
   a. What is his decision?

ID3 Algorithm builds the dataset

Split(node, {examples}):

1. Find best attribute to split on (A)
2. Decision attribute for this node (A)
3. For each value of A, create a new child node

4. Split training {examples} to child nodes
5. For each child node:

```
    if subset is pure: stop
    else: split(node, {examples})
```

How to know which attribute to split on?

Example:

Look at original data, which attribute is best to split on?

We like splits with pure subsets, or have a higher certainty

4/0 -- completely certain (100%)

3/3 -- completely uncertain (50%)

Use Entropy:

$H(S) = -p_+ * \log_2(p_+) - p_- * \log_2(p_-)$

S .. subset of training examples

p_+/p_- .. % of positive / negative examples in S

Interpretation: assume item X belongs to S

-how many bits need to tell if X is positive or negative

-lets try this with different values like .5,.5 and 1,0

-log2(x) ... the power you need to raise 2 to get x

GainSplit = Gain(s,a) = H(S) - $\sum(\frac{|Sv|}{|S|} * H(S_v)$

V ... possible values of A

S ... set of examples {X}

S_v .. Subset where X_a = V

Explain with golf example

-Splitting on entropy, bias towards attributes with lots of values

-Large trees with many branches prefered

-Don't use ID as an attribute!