**Artificial Intelligence & Machine Learning | Batch 1 - Day 25 Overview**

**DAY 25 AGENDA**

* Naive Bayes Theorem
* Classification Technique
* Hands-On in Python Of Naive Bayes Theorem
* Training and Testing

**Naive Bayes Theorem :**

1. This is a classification technique.
2. We use probability to classify the records.
3. We can apply this theorem only when we have dependent and independent variables categorical.
4. It is useful for classifying large datasets.
5. We use probability to classify the records and with proper formula and structure.
6. In the example dataset, we have a dependent variable is if the company is truthful or not.
7. The independent variable is the size of the company or the company is legally charged or not.
8. We will check how many companies are fraud and then out of those we will check according to the size and those fraud ones we have 3 companies with charge yes.
9. We will find the probability of the companies fraud out of that how many frauds, small in size and charged or not, and then multiply the probability of each.
10. p(fraud|charges,small) =(4/10)\*(1/4)\*(3/4)=0.075/(0.075+0.067)=0.53
11. p(truthful|charges,small)=(1/6)\*(4/6)\*(6/10)=0.067/(0.075+0.067)=0.471
12. Looking at data we calculated we can say that if the company charges and it is small we can classify it is a fraud company.
13. In the next example, we have 14 days of data of the badminton series which has all the weather data and the dependent variable says if the match was played or not.
14. The Independent variable here is the outlook, temperature, humidity, wind and the dependent variable is if the match was played or not.
15. We will then look for the number of records and check how many days match was played and in those days how many days the outlook is sunny and further specific details.
16. We will find the probability for different scenarios.
17. Finding the probability for different conditions is called the conditional probability.
18. We have to predict on a day sunny, cool temp, high humidity, and strong wind.
19. To find the prediction we multiply yes for all conditions and no for all the conditions and we find if yes has a higher probability the match will be played or else it will no be played.
20. It finds its application in Gmail for spam classification.

**Classification Technique**

|  |  |  |  |
| --- | --- | --- | --- |
| Technique | Dependent Variable | Independent variable | Purpose |
| Naive Bayes Theorem | Categorical | Categorical | Classification technique used to classify the records using probability |
| K-nearest neighbor(KNN) | Categorical | Categorical and continuous | Classification technique used to classify record with the help of Euclidean Distance |
| Support Vector Machine(SVM) | Categorical | Categorical and continuous | Classification technique and used to classify record with the help of Hyperplane and is applicable for over-dimensional data |

**Hands-On in Python Of Naive Bayes Theorem :**

1. *Import pandas*
2. *load the data set*
3. *from sklearn import p reprocessing*
4. *from sklearn.cross\_validation import train\_test\_split (if*cross-validation*is not supported we can use sklearn.model\_selection for loading train\_test\_split function)*
5. *from sklearn.naive\_bayesimport GaussianNB*
6. *from sklearn.metrics import accuracy\_score #*Tells the model accuracy.
7. *From sklearn.matrics import confusion\_matrix*#Confusion matrix tells us how many reocrds are classified accurately .
8. *le=preprocessing.LabelEncoder()*
9. [*le.fit*](http://le.fit/)*(dataset[“Sec”]*
10. *print(le.classes\_)*#Gives out classes
11. *dataset[“Sex’]=le.transform(dataset[“Sex”]) #*Coverting categorical to Continuous data
12. *y=dataset[“Survived”] #*Defining dependent varible
13. *X=dataset.drop([“Survived”,”PassengerId”],axis=1)* #axis=1 defines columns and 0 defines rows
14. *y.count()* #Gives the number of record
15. Independent variable is denoted by X and dependent by y.

**Training and Testing :**

1. *X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=0)  #*Splitting in training and testing dataset
2. *X\_train.head()*
3. *from sklearn.naive\_bayes import \**
4. *clf=BernoulliNB()*
5. *y\_pred-clf.fit(X\_train,y\_train).predict(X\_test)*
6. *accuracy\_score(y\_test,y\_pred,normalize=True)*#Gives out Accuracy score
7. *confusion\_matrix(y\_test,y\_pred)* #checks how many records were classified correctly

**Reading Confusion Matrix :**

1. Training dataset is in the column and test or predicted in the rows .
2. The diagonals are correctly classified.
3. Other than diagonal others are no classified correctly

**Assignment:**

1. First take pclass , gender, SibSP , ParCh, Embarked as dependent variable one by one and others as the independent variable and find the confusion matrix. Using looping or function how can you minimize the code.