13. NAÏVE BAYES (NB) CLASSIFIER

- NB is a supervised machine learning algorithm used for classification and it has a wide range of applications such as sentiment analysis, text categorization, recommendation systems, etc.

  Algorithms that learn $P(y|x)$, the conditional probability of the target variable given the input features, directly by using a discriminative function (such as logistic regression using sigmoid function) or algorithms that try to learn mappings from the input space to the set of labels (such as perceptron) are called *discriminative* learning algorithms.

  Instead of modeling $P(y|x)$, *generative* learning algorithms model $P(x|y)$. NB is an example of such an algorithm.

- **Conditional Probabilities, Independent Events, Bayes' Theorem**

  - *Definition of Conditional Probability*:
    If A and B are events, such that $P(B) \neq 0$, then the conditional probability of A given B is defined by

    $$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

  - *Definition of Independent Events*:
    Events A and B are independent if

    $$P(A|B) = P(A) \quad \text{or, equivalently,} \quad P(B|A) = P(B)$$

  - *Theorem (Multiplicative Rule):*
    If A and B are independent events, then

    $$P(A \cap B) = P(A) \cdot P(B)$$

    Proof:

    $$\begin{aligned} P(A \cap B) &= P(A|B) \cdot P(B) && \text{from the definition of conditional probability} \\ &= P(A) \cdot P(B) && \text{from the definition of independence} \end{aligned}$$

  - *Bayes' Theorem:*
    If A and B are events, such that $P(B) \neq 0$, then

    $$\boxed{P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}}$$

    where $P(A)$ and $P(B)$ are probabilities of observing A and B without any given conditions (they are known as the marginal probabilities or prior probabilities) and $P(A|B)$ and $P(B|A)$ are conditional probabilities.

    Proof: Recall that $P(A \cap B) = P(A|B)\, P(B)$ as well as that $P(A \cap B) = P(B|A)\, P(A)$. Using the definition of conditional probability we get

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

- *Definition of Conditional Independence:*

  Events A and B are conditionally independent given an event C is

$$P(A|B \cap C) = P(A|C), \quad \text{or, equivalently,} \quad P(B|A \cap C) = P(B|C)$$

- *Remark*: Conditional independence neither implies (nor is it implied by) independence. Consider flipping a coin two times and

$$\begin{aligned} A = &\ \text{the first outcome is a head} \\ B = &\ \text{the second outcome is a head} \\ C = &\ \text{two outcomes are same} \end{aligned}$$

  A and B are independent, but they are not conditionally independent given C.

- *Proposition:*

  If A and B are conditionally independent given an event C, then

$$\boxed{P(A \cap B|C) = P(A|C) \cdot P(B|C)}$$

  Proof: Since $P(A|B \cap C) = P(A|C)$ and the left-hand side of this expression is

$$\frac{P(A \cap B \cap C)}{P(B \cap C)}$$

  we have

$$P(A \cap B \cap C) = P(A|C) \cdot P(B \cap C)$$

  Therefore,

$$\begin{aligned} P(A \cap B|C) &= \frac{P(A \cap B \cap C)}{P(C)} && \text{by the definition of conditional probability} \\ &= \frac{P(A|C) \cdot P(B \cap C)}{P(C)} && \text{by the above formula} \\ &= P(A|C) \cdot P(B|C) && \text{by the definition of conditional probability} \end{aligned}$$

- Therefore we have

$$\boxed{P(y \mid x_1, \ldots, x_d) = \frac{P(x_1, \ldots, x_d \mid y) \cdot P(y)}{P(x_1, \ldots, x_d)}}$$

  - Terminology:
    - $P(y)$ is the class *prior*, i.e., the probability that a randomly chosen data instance is from class y
    - $P(x_1, \ldots, x_d)$ is called the *evidence*
    - $P(x_1, \ldots, x_d \mid y)$ is the *likelihood*
    - $P(y \mid x_1, \ldots, x_d)$ is the *posterior* probability; i.e, the probability that a data instance with specific features $(x_1, \ldots, x_d)$ is from class y

Note that to predict the class given fixed values of features $(x_1, \dots, x_d)$, it sufficies to predict the class y that has the highest numerator. Therefore, our goal is to estimate $P(x_1, \dots, x_d \,|\, y)$ and $P(y)$. In order words,

$$\begin{aligned} \text{argmax}_y \, P(y \,|\, x_1, \dots, x_d) &= \text{argmax}_y \, \frac{P(x_1, \dots, x_d \,|\, y) \cdot P(y)}{P(x_1, \dots, x_d)} \\ &= \text{argmax}_y \, P(x_1, \dots, x_d \,|\, y) \cdot P(y) \end{aligned}$$

If the features are discrete, finding $P(x_1, \dots, x_d \,|\, y)$ and $P(y)$ is done simply by looking at the table of values for features $(x_1, \dots, x_d)$ and labels y and counting. The problem with this method is that it might be hard to find the occurrence of $(x_1, \dots, x_d)$ in our training data set. One way to go around this is to assume that
$\underline{\text{features } x_1, \dots, x_n \text{ are conditionally independent given } y}$. In that case we have

$$P(x_1, \dots, x_d \,|\, y) = P(x_1 \,|\, y) \cdot \ldots \cdot P(x_d \,|\, y)$$

Since in the real world, we cannot assume that all variables satisfy this assumption, this approach is called *naïve.*

Therefore, given a data instance $(x_1, \dots, x_d)$, we find y that maximizes

$$\boxed{P(x_1 \,|\, y) \cdot \ldots \cdot P(x_d \,|\, y) \cdot P(y)}$$

Example: (data set partially taken from https://www.youtube.com/watch?v=l3dZ6ZNFjo0)

Day	Discount	Free Delivery	Purchase
weekday	yes	yes	yes
weekday	yes	yes	yes
weekday	yes	no	no
holiday	yes	yes	yes
weekend	yes	yes	yes
holiday	no	no	no
weekend	yes	no	yes
weekday	yes	yes	yes
weekend	yes	yes	yes
holiday	yes	yes	yes
holiday	no	yes	yes
holiday	no	yes	no

Predict whether the customer will make a purchase given features (holiday, yes, no).

Solution: We want to find

$$P(y \,|\, \text{Day} = \text{holiday, Discount} = \text{yes, Free Delivery} = \text{no})$$

for each of the two classes y=yes and y=no.

We use the Bayes rule and assume that features are independent conditionally given Y to get

$$\begin{aligned}
&P(Y = y \,|\, X_1 = \text{holiday},\, X_2 = \text{yes},\, X_3 = \text{no}) \\
&\qquad\qquad = \frac{P(X_1 = \text{holiday},\, X_2 = \text{yes},\, X_3 = \text{no} \,|\, Y = y) \cdot P(Y = y)}{P(X_1 = \text{holiday},\, X_2 = \text{yes},\, X_3 = \text{no})} \\
&\qquad\qquad \propto P(X_1 = \text{holiday},\, X_2 = \text{yes},\, X_3 = \text{no} \,|\, Y = y) \cdot P(Y = y) \\
&\qquad\qquad = P(X_1 = \text{holiday} \,|\, Y = y) \cdot P(X_2 = \text{yes} \,|\, Y = y) \cdot P(X_3 = \text{no} \,|\, Y = y) \cdot P(Y = y)
\end{aligned}$$

Note that $P(Y = \text{yes}) = 9/12$ and $P(Y = \text{no}) = 3/12$.
Therefore,

$$\begin{aligned}
P(Y = \text{yes} \,|\, X_1 = \text{holiday},\, X_2 = \text{yes},\, X_3 = \text{no}) &\propto 3/9 \cdot 8/9 \cdot 1/9 \cdot 9/12 \\
&= \boxed{0.025}
\end{aligned}$$

and

$$\begin{aligned}
P(Y = \text{no} \,|\, X_1 = \text{holiday},\, X_2 = \text{yes},\, X_3 = \text{no}) &\propto \cdot 2/3 \cdot 1/3 \cdot 2/3 \cdot 3/12 \\
&= \boxed{0.037}
\end{aligned}$$

Hence, we predict that this customer will not make a purchase.

Remark:

- In case $P(x_1, \ldots, x_d \,|\, y) = 0$, for any class y, *Laplace smoothing* is used [1].
- In case of a continuous feature, we can:
  - *use discretization*

    For example, if we have a continuous feature age $\in [11, 56]$, we can discretize it using categories $1, \ldots, 5$ and assume the feature value is in category 1 if age is between 11 and 20, it is in category 2 is age is between 21 and 30, etc.

categories	age
1	11-20
2	21-30
3	31-40
4	41-50
5	51-60

  - *fit a probability distribution* on the feature data values, given a label y, and use the probability mass/density function to compute the probabilities.

    For example, if given a specific label y_k, the values of the feature X_i follow a normal distribution, compute $\mu_{i,k}$ and $\sigma_{i,k}$, and use

    $$P(x|y_k) = \frac{1}{\sqrt{2\pi}\sigma_{i,k}} \cdot \exp\left(-\frac{(x - \mu_{i,k})^2}{2\sigma_{i,k}^2}\right)$$

- Types of `sklearn NB Classifiers`:
  - *Bernoulli NB* – features are binary (0 and 1 values)

    For example, we can use Bernoulli NB in text classification where features are words, 0 means the word does not appear in the document and 1 means the word appears in the document.
  - *Multinomial NB* – features are discrete/categorical

    For example, we can use it if a feature describes movie ratings ranging 0 – 5. Also, in text classification where features are words and values is the number of occurrences of each word in the document.
  - *CategoricalNB* – features are discrete/categorical

    Assumes that each feature has its own categorical distribution.
  - *Gaussian NB* – features are continuous and follow the Gaussian distribution

    If a continuous feature does not follow a Gaussian distribution, use some transformation to convert it to Gaussian distribution.
- **Summary:**
  - Advantages:
    - Very simple and easy to implement.
    - Fast.
    - Performs well in multi-class prediction.
    - When assumption of conditional independence holds, NB performs better compared to other models.
    - Needs less training data.
  - Disadvantages:
    - If there is a value of a categorical variable in test data which was not observed in training data set, then model will assign a 0 probability and will be unable to make a prediction. This is often known as ”Zero Frequency” and to solve this, we can use Laplace smoothing technique.
    - Assumes conditionally independent features. In real life, it is almost impossible that we get a set of features which are completely conditionally independent.
    - Another assumption is that all the features have an equal effect on the label, which may not be true.
  - Applications:
    - text classification (spam/ham) and sentiment analysis (positive/negative)
    - recommendation systems

Python code: Lecture_13_NaiveBayes.ipynb

Homework 8:

- *Part I:* Explain Laplace smoothing (for example, see reference [1]).
- *Part II:* Build a Naïve Bayes algorithm on the `titanic` data set attached to OneDrive to predict whether a passenger survived or not.

  This data set provides information on the fate of passengers on the fatal maiden voyage of the ocean liner ”Titanic”, summarized according to survival (target variable with 1=survived and 0=died) and explanatory variables: `Name`, `Pclass` (passenger class), `Sex`, `Age`, `SibSp` (total number of siblings including the spouse traveling with the passenger), `Parch` (total number of parents and children traveling with the passenger), `Ticket`, `Fare`, `Cabin`, and `Embarked` (where the traveler mounted from: Southampton, Cherbourg, or Queenstown).

  1. Import the data set into pandas data frame.
  2. Split the data into training and test sets.
  3. Select one or more explanatory variables you would like to use.
  4. Figure out if there are any missing values in the explanatory variables you want to use and either delete those passengers from the data set or fill in the missing values. If a numerical variable has missing values, you might fill those in with the average or median of that variable. If a categorical variable has missing values, you might fill those in using the most common value. You can create your own script for missing values or your can use `sklearn SimpleImputer`.
  5. Convert the categorical variables to numerical using encoding. You can create your own script or use `sklearn LabelEncoder`.
  6. Build a model on the training data. You can create your own code or use `sklearn NaïveBayes`. If you use a mix of continuous and categorical explanatory variables, think of how you can build the model.
  7. Inspect the evaluation measures (accuracy score, confusion matrix, classification report).
  8. Take some values for the explanatory variables and use your model to predict if that person would have survived or not.

References and Reading Material:

[1] Andrew Ng's notes (Part IV: Generative Learning Algorithms), pages 31-42

[2] https://www.youtube.com/watch?v=l3dZ6ZNFjo0

[3] https://github.com/Suji04/ML_from_Scratch/blob/master/naive%20bayes.ipynb