Building a Movie Recommendation Engine

with Naïve Bayes

Machine Learning Practice With Python

Siksha 'O' Anushandhan, ITER

Center for Data Science

Table of Contents

Classification in Machine Learning

- **Classification in ML**: A type of supervised learning where the model learns to map **features (input data)** to **labels/classes (categories)** based on a training dataset.
- **Goal**: Learn a **general rule** from given observations and their associated categorical outputs to classify future data correctly.
- **Training Phase**: The model uses **features** + **labels** from known data to build a **trained classification model**.
- **Prediction Phase**: When **new, unseen data** arrives, the trained model predicts the **class membership** based on input features.

Classification in Machine Learning

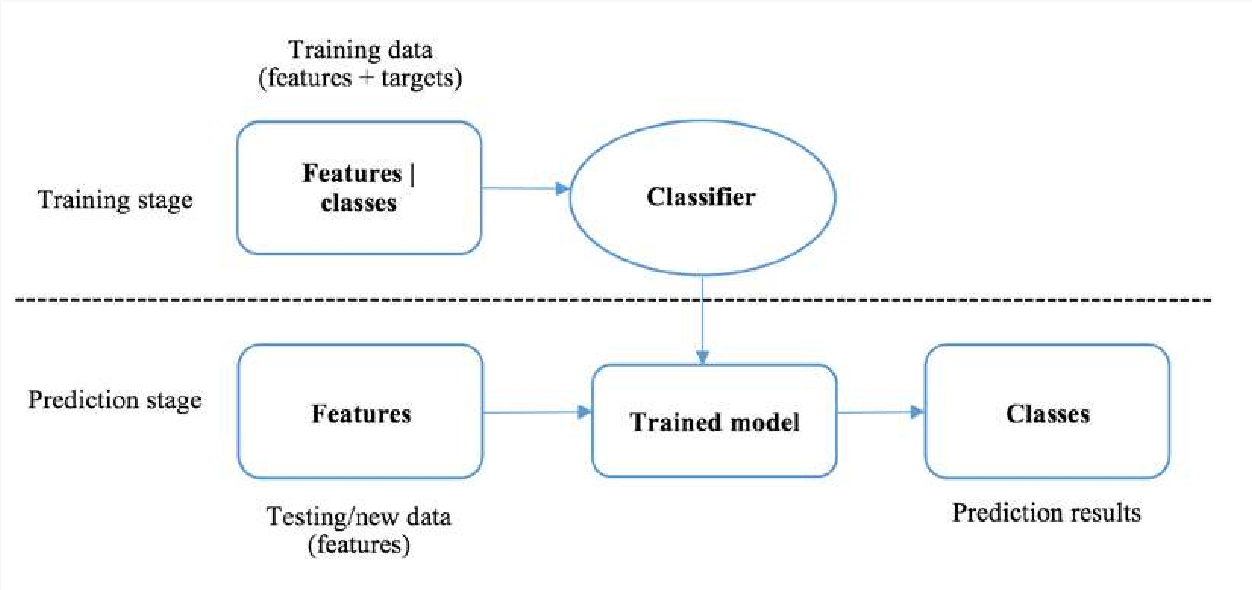


Figure: **The training and prediction stages in classification**

Types of Classification (Binary Classification)

Binary Classification: Categorizes data into **two possible classes** e.g., spam vs. not spam in email filtering, churn vs. retain in customer prediction systems.

Real-World Applications

Marketing/Advertising

Predict if an online ad will be **clicked** using user data (cookies, browsing history).

Customer Analytics

Identify potential **churners** using CRM and behavioral data.

Biomedical Use

Assist early diagnosis e.g., classify patients into **high-risk vs. low-risk** groups for diseases like cancer from MRI images.

Types of Classification (Multiclass Classification)

Multiclass Classification: Assigns observations to **more than two classes**, unlike binary classification with only two.

Classic Example

Handwritten digit recognition (digits **0-9**) historically important and used in automatic **ZIP code reading**.

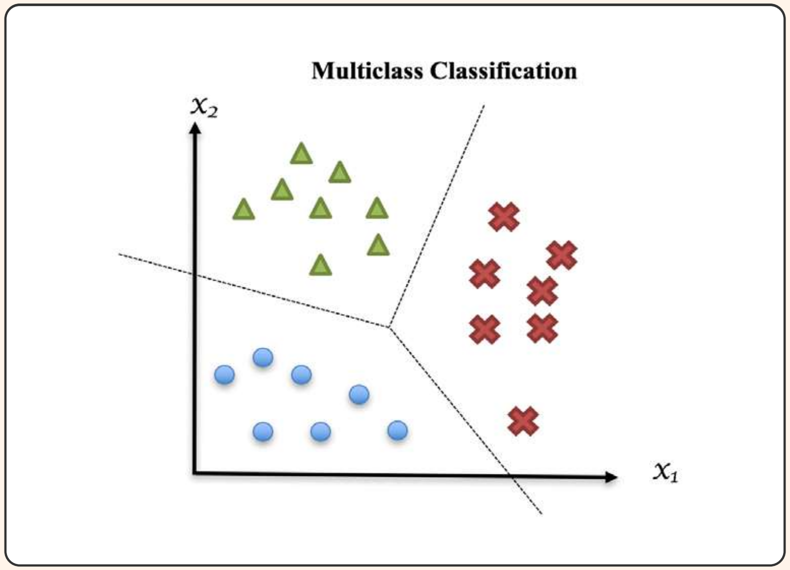
MNIST Dataset

Classic **benchmark** for testing multi-class classifiers on handwritten digits.

60,000 training images

10,000 test images

Visual Example



Exploring Naïve Bayes

- **Naïve Bayes** is a **probabilistic classifier** that computes the probability of a data sample belonging to each class based on its **predictive features** (attributes or signals).
- It produces a **probability distribution** over all classes and selects the **most likely class** for prediction.
- From the resulting probability distribution, we can conclude the **most likely class** that the data sample is associated with.

What Naïve Bayes does specifically, as its name indicates:

Bayes: Uses Bayes' theorem to relate the probability of observed features given a class to the probability of the class given those features.

⋆ Naïve: Assumes all predictive features are mutually independent to simplify probability calculations.

Bayes' Theorem with Examples

Bayes' theorem

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

where, $P(B \mid A)$ is the probability of B given A, while $P(A)$ and $P(B)$ are the probabilities that A and B occur, respectively.

Exercise:

Given two coins, one is unfair, with 90% of flips getting a head and 10% getting a tail, while the other one is fair. Randomly pick one coin and flip it. What is the probability that this coin is the unfair one, if we get a head?

Bayes' Theorem with Examples

Solution

Event Definitions:

- A = event of picking the **unfair** coin
- B = The flip shows head
- To find: $P(A|B)$
- Given: $P(A) = 0.5 \qquad P(\neg A) = 0.5 \qquad P(B \mid A) = 0.9 \qquad P(B \mid \neg A) = 0.5$

$$P(B) = P(B \mid A)P(A) + P(B \mid \neg A)P(\neg A) = 0.70$$
$$\Rightarrow P(A \mid B) \approx 0.64$$

Bayes' Theorem with Examples

Exercise:

Suppose a physician reported the following cancer screening test scenario among 10,000 people:

	Cancer	No Cancer	Total
Test Positive	80	900	980
Test Negative	20	9000	9020
Total	100	9900	10000

If a person test positive, what is the probability that actually have cancer?

Bayes' Theorem with Examples

Exercise:

Suppose a physician reported the following cancer screening test scenario among 10,000 people:

	Cancer	No Cancer	Total
Test Positive	80	900	980
Test Negative	20	9000	9020
Total	100	9900	10000

If a person test positive, what is the probability that actually have cancer?

$$P(Cancer \mid test+ve) = \frac{P(test+ve \mid Cance) \times P(Cancer)}{P(test+ve)} = \frac{\frac{80}{100} \times \frac{100}{10000}}{\frac{980}{10000}}$$

$$= 0.08163$$

Exercise

Exercise:

1. Three machines A, B, and C in a factory account for 35%, 20%, and 45% of bulb production. The fraction of defective bulbs produced by each machine is 1.5%, 1%, and 2%, respectively. A bulb produced by this factory was identified as defective, which is denoted as event D. What are the probabilities that this bulb was manufactured by machines A, B, and C, respectively.
2. At an airport, 1% of passengers carry prohibited items. A security scanner correctly identifies a passenger with a prohibited item 95% of the time, but it also falsely alarms 5% of passengers who dont carry prohibited items. If a passenger triggers the alarm, what is the probability that they actually have a prohibited item?

Naïve Bayes Classifier mechanism overview

Given a data sample $\boldsymbol{x}$ with n features $x_1, x_2, \ldots, x_n$ (where $\boldsymbol{x}$ represents a feature vector and $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$).

The **goal** of Nave Bayes is to determine the probabilities that this sample belongs to each of K possible classes $y_1, y_2, \ldots, y_K$, which is $P(y_k \mid \boldsymbol{x})$ for $k = 1, 2, \ldots, K$.

Bayes Theorem:

$$\boxed{P(y_k \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x} \mid y_k) \times P(y_k)}{P(\boldsymbol{x})}}$$

- $P(y_k)$: **Prior probability** of class k
- $P(\boldsymbol{x} \mid y_k)$, or equivalently $P(x_1, x_2, \ldots, x_n \mid y_k)$, is the **joint distribution** of n features given that the sample belongs to class y_k.
- $P(y_k \mid \boldsymbol{x})$: **Posterior probability** of class given features
- $P(\boldsymbol{x})$: **Evidence** (normalization factor)

Naïve Independence Assumption

Assumption: Features are **conditionally independent**

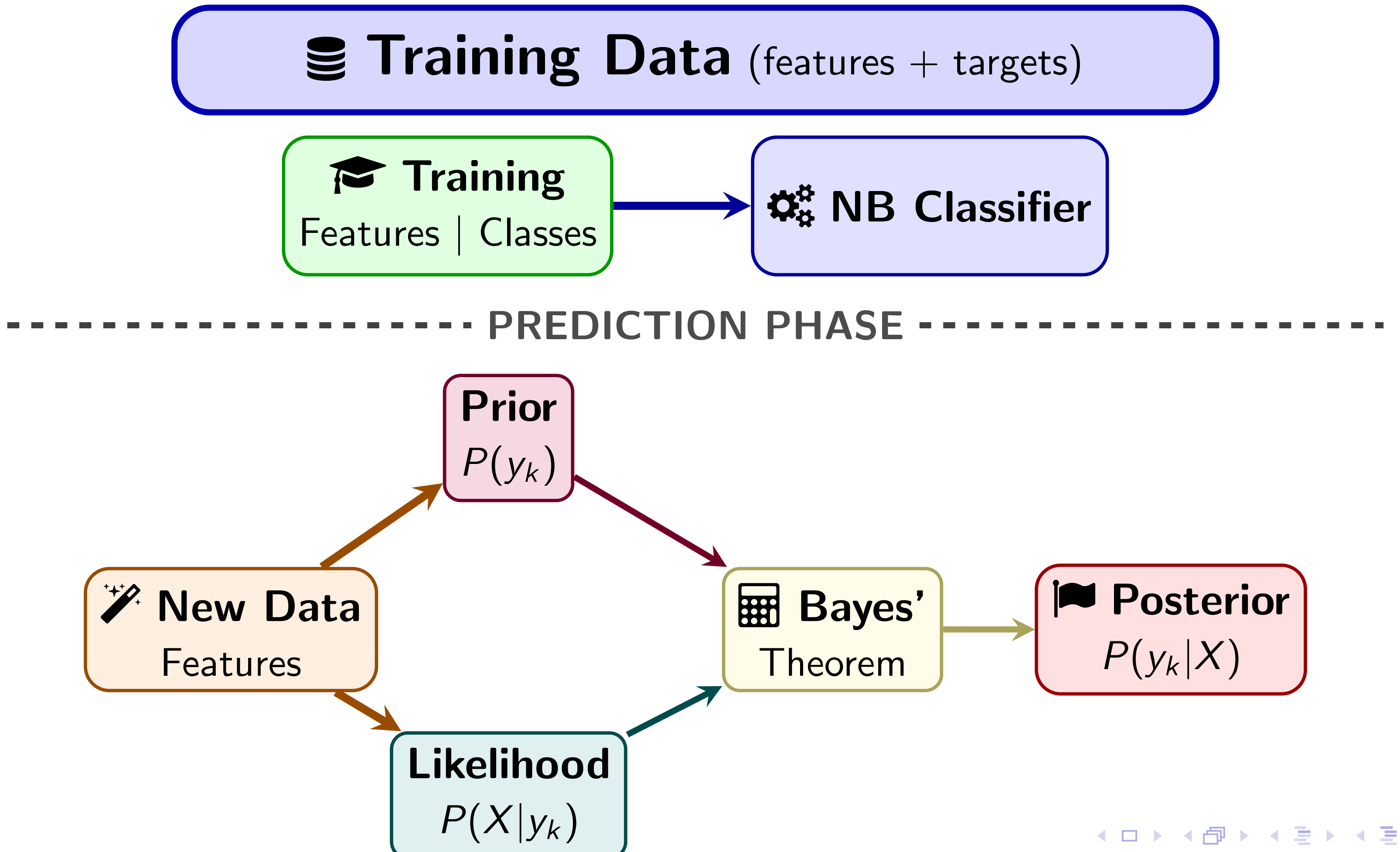
The joint conditional distribution of **n** features can be expressed as:

$$P(\mathbf{x} \mid y_k) = \prod_{i=1}^{n} P(x_i \mid y_k)$$

Hence, the posterior probability is proportional to:

$$P(y_k \mid \mathbf{x}) \propto P(y_k) \prod_{i=1}^{n} P(x_i \mid y_k)$$

Naïve Bayes: Training & Prediction Workflow



Example

Example

Given four (pseudo) users, whether they like each of three movies, m_1, m_2, m_3 (indicated as 1 or 0), and whether they like a target movie (denoted as event Y) or not (denoted as event N), as shown in the following table, we are asked to predict how likely it is that another user will like that movie:

ID	m_1	m_2	m_3	Whether the user likes the target movie
1	0	1	1	Y
2	0	0	1	N
3	0	0	0	Y
4	1	1	0	Y
Testing case				
5	1	0	1	?

Naïve Bayes Classification Example

Training Data

$P(Y) = \frac{3}{4}$, $P(N) = \frac{1}{4}$

Y $=$ {users 1,3,4}, N $=$ {user 2}

Naïve Bayes Classification Example

Training Data

$P(Y) = \frac{3}{4}$, $P(N) = \frac{1}{4}$
Y = {users 1,3,4}, N = {user 2}

Class Y (3 examples)

$P(m_1 = 1|Y) = \frac{1}{3}$ (user 4)
$P(m_2 = 0|Y) = \frac{1}{3}$ (user 3)
$P(m_3 = 1|Y) = \frac{1}{3}$ (user 1)
$P(\mathbf{x}|Y) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{27}$

$$P(Y) \cdot P(\mathbf{x}|Y) = \frac{3}{4} \cdot \frac{1}{27} = \boxed{\frac{1}{36}}$$

Naïve Bayes Classification Example

Training Data

$P(Y) = \frac{3}{4}$, $P(N) = \frac{1}{4}$
Y = {users 1,3,4}, N = {user 2}

Class Y (3 examples)

$P(m_1 = 1|Y) = \frac{1}{3}$ (user 4)
$P(m_2 = 0|Y) = \frac{1}{3}$ (user 3)
$P(m_3 = 1|Y) = \frac{1}{3}$ (user 1)
$P(\mathbf{x}|Y) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{27}$

$$P(Y) \cdot P(\mathbf{x}|Y) = \frac{3}{4} \cdot \frac{1}{27} = \boxed{\frac{1}{36}}$$

Class N (1 example)

Problem: $P(m_1 = 1|N) = 0$
(user 2 has $m_1 = 0$)
$P(\mathbf{x}|N) = 0 \cdot P(m_2 = 0|N) \cdot P(m_3 = 1|N)$
$P(\mathbf{x}|N) = \boxed{0}$

Naïve Bayes Classification Example

Training Data

$P(Y) = \frac{3}{4}$, $P(N) = \frac{1}{4}$
Y = {users 1,3,4}, N = {user 2}

Class Y (3 examples)

$P(m_1 = 1|Y) = \frac{1}{3}$ (user 4)
$P(m_2 = 0|Y) = \frac{1}{3}$ (user 3)
$P(m_3 = 1|Y) = \frac{1}{3}$ (user 1)
$P(\mathbf{x}|Y) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{27}$

$$P(Y) \cdot P(\mathbf{x}|Y) = \frac{3}{4} \cdot \frac{1}{27} = \boxed{\frac{1}{36}}$$

Class N (1 example)

Problem: $P(m_1 = 1|N) = 0$
(user 2 has $m_1 = 0$)
$P(\mathbf{x}|N) = 0 \cdot P(m_2 = 0|N) \cdot P(m_3 = 1|N)$
$P(\mathbf{x}|N) = \boxed{0}$

Final Result

After normalization:
$\boxed{P(Y|\mathbf{x}) = 1}$
$\boxed{P(N|\mathbf{x}) = 0}$
Classification: Y

Implementing Naïve Bayes with scikit-learn

- Define dataset

```
import numpy as np
X_train = np.array([[0, 1, 1],  [0, 0, 1], [0, 0, 0], [1, 1, 0
        ]])
Y_train = ['Y', 'N', 'Y', 'Y']
X_test = np.array([[1, 0, 1]])
```

- Create Naïve-Bayes model from scikit-learn

Implementing Naïve Bayes with scikit-learn

- Define dataset

```
import numpy as np
X_train = np.array([[0, 1, 1],  [0, 0, 1], [0, 0, 0], [1, 1, 0]])
Y_train = ['Y', 'N', 'Y', 'Y']
X_test = np.array([[1, 0, 1]])
```

- Create Naïve-Bayes model from scikit-learn

```
from sklearn.naive_bayes import BernoulliNB
clf = BernoulliNB(alpha=1.0, fit_prior=True)
```

- Train and predict class

Implementing Naïve Bayes with scikit-learn

- Define dataset

```python
import numpy as np
X_train = np.array([[0, 1, 1],  [0, 0, 1], [0, 0, 0], [1, 1, 0]])
Y_train = ['Y', 'N', 'Y', 'Y']
X_test = np.array([[1, 0, 1]])
```

- Create Naïve-Bayes model from scikit-learn

```python
from sklearn.naive_bayes import BernoulliNB
clf = BernoulliNB(alpha=1.0, fit_prior=True)
```

- Train and predict class

```python
clf.fit(X_train, Y_train)
pred = clf.predict(X_test)
print('[scikit-learn] Prediction:', pred)
```

```
[scikit-learn] Prediction: ['Y']
```

Building a Movie Recommender with Naïve Bayes

MovieLens Small Dataset

Download: `ml-latest-small.zip` (1 MB)
files.grouplens.org/datasets/movielens/

Dataset Statistics:

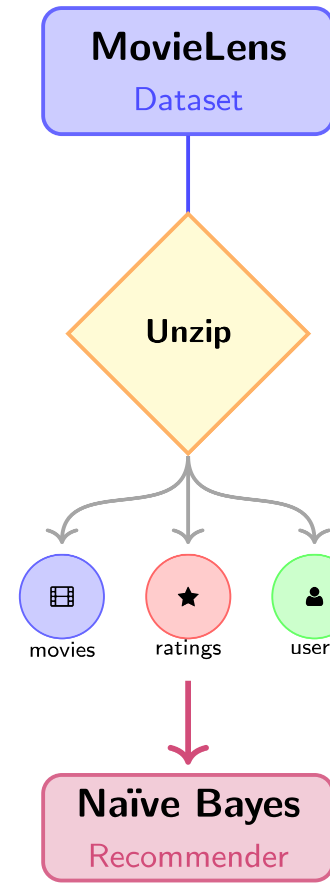
100,836 ratings **6,10** users **9724** movies

Dataset Files

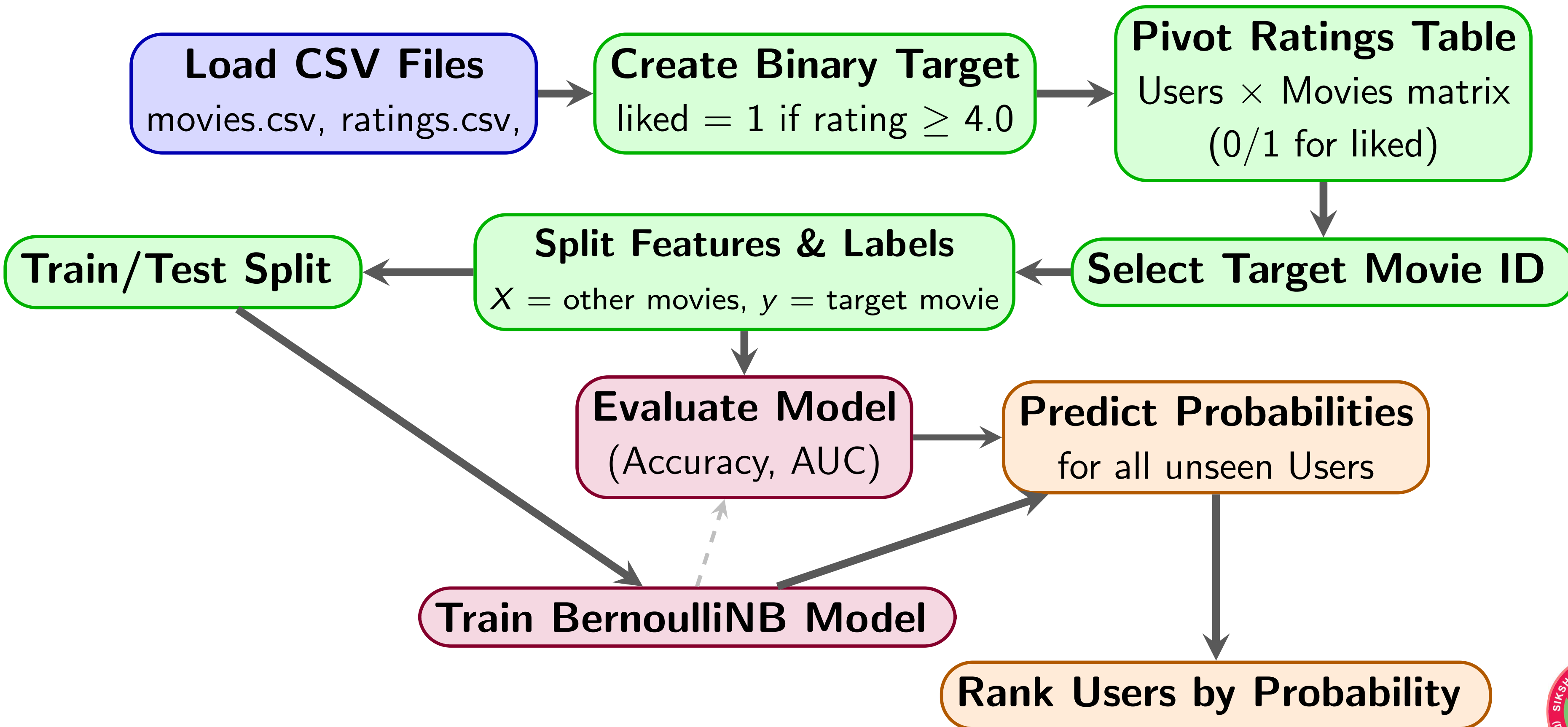
`movies.csv`	movieId::title::genres
`ratings.csv`	userId::movieId::rating::timestamp *(primary file for this model)*
`README.txt`	Documentation

Target

Predict a particular movie will recommend to which user.



Movie Recommendation System Workflow



Building a Movie Recommender with Naïve Bayes

- Import data (ratings.csv, movies.csv)

```
import numpy as np
import pandas as pd
movies = pd.read_csv("movies.csv")
ratings = pd.read_csv("ratings.csv")
print(movies.head())
print(ratings.head())
```

- Merge `movies` and `ratings` on `movieId` [Optional]

Building a Movie Recommender with Naïve Bayes

- Import data (ratings.csv, movies.csv)

```
import numpy as np
import pandas as pd
movies = pd.read_csv("movies.csv")
ratings = pd.read_csv("ratings.csv")
print(movies.head())
print(ratings.head())
```

- Merge `movies` and `ratings` on `movieId` [Optional]

```
df = pd.merge(ratings, movies, on='movieId')
df.head()
```

Building a Movie Recommender with Naïve Bayes

- Now, lets see how many unique users and movies are in the dataset

Building a Movie Recommender with Naïve Bayes

- Now, lets see how many unique users and movies are in the dataset

```
n_users = df['userId'].nunique()
n_movies = df['movieId'].nunique()
print(f"Number of users: {n_users}")
print(f"Number of movies: {n_movies}")
```

```
Number of users: 610
Number of movies: 9724
```

- Check which rating is rated by how many users

Building a Movie Recommender with Naïve Bayes

- Now, lets see how many unique users and movies are in the dataset

```python
n_users = df['userId'].nunique()
n_movies = df['movieId'].nunique()
print(f"Number of users: {n_users}")
print(f"Number of movies: {n_movies}")
```

```
Number of users: 610
Number of movies: 9724
```

- Check which rating is rated by how many users

```python
values, counts = np.unique(df['rating'], return_counts=True)
for value, count in zip(values, counts):
print(f'Number of rating {value}: {count}')
```

```
Number of rating 0.5: 1370
Number of rating 1.0: 2811
Number of rating 1.5: 1791
Number of rating 2.0: 7551
```

```
Number of rating 3.0: 20047
Number of rating 3.5: 13136
Number of rating 4.0: 26818
Number of rating 4.5: 8551
```

Building a Movie Recommender with Naïve Bayes

- Create a new column "`liked`" with values 1 if **rating** is ≥ 4.