Experiment 5 - Naive Bayes Classifier

April 30, 2023

1 Experiment Details

1.1 Submitted By

Desh Iyer, 500081889, Year III, AI/ML(H), B5

1.2 Problem Statement

Write a program to implement the Naïve Bayesian classifier for a sample training data set stored as a `.csv` file. Compute the accuracy of the classifier, considering few test data sets.

1.3 Theory

The Naive Bayes Classifier is a probabilistic algorithm used for classification tasks. It is based on Bayes' theorem, which states that the probability of a hypothesis H given some observed evidence E is proportional to the probability of the evidence given the hypothesis times the prior probability of the hypothesis.

The Naive Bayes Classifier assumes that the features are independent of each other given the class label. That is, it assumes that the probability of observing a particular combination of feature values is the product of the probabilities of observing each individual feature value given the class label.

1.4 Steps

Here are the steps to implement the Naive Bayes Classifier:

1. Load the training data from the CSV file.
2. Preprocess the data, if necessary. This may involve steps such as removing missing values, converting categorical variables to numerical ones, and scaling numerical variables.
3. Split the data into training and testing sets.
4. Calculate the prior probabilities of each class label in the training set.
5. For each feature, calculate the conditional probabilities of each possible value given each class label in the training set.
6. For each test instance, calculate the posterior probability of each class label given the feature values using Bayes' theorem.
7. Assign the test instance to the class with the highest posterior probability.
8. Evaluate the accuracy of the classifier on the testing set by comparing the predicted class labels to the true class labels.

1.5 Pseudocode

Here's the pseudocode for the Naive Bayes Classifier:

```
# Load the data
data = load_csv('data.csv')

# Preprocess the data, if necessary
data = preprocess_data(data)

# Split the data into training and testing sets
train_set, test_set = split_data(data)

# Calculate the prior probabilities of each class label
priors = calculate_priors(train_set)

# Calculate the conditional probabilities of each feature given each class label
cond_probs = calculate_conditional_probs(train_set)

# Classify the test set
predictions = []
for instance in test_set:
    posterior_probs = calculate_posterior_probs(instance, priors, cond_probs)
    predicted_class = get_max_class(posterior_probs)
    predictions.append(predicted_class)

# Evaluate the accuracy of the classifier
accuracy = calculate_accuracy(test_set, predictions)
print('Accuracy:', accuracy)
```

2 Import Libraries

[ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

import pickle

import pandas as pd
```

3 Load Data

[ ]:
```
data = pd.read_csv('naive-bayes-classification-data.csv')
```

4 Train-test Split

```
[ ]: X_train, X_test, y_train, y_test = train_test_split(
         data[['glucose', 'bloodpressure']], data['diabetes'], test_size=0.3,␣
      ↪random_state=42)

     print(f'X_Train shape: {X_train.shape}\ny_train shape: {y_train.shape}')
```

```
X_Train shape: (696, 2)
y_train shape: (696,)
```

5 Define Gaussian Naive-Bayes Classifier

```
[ ]: classifier = GaussianNB()
     y_predicted = classifier.fit(X_train, y_train).predict(X_test)
```

6 Print Accuracy

```
[ ]: print("Number of mislabeled points out of a total %d points : %d" %
           (X_test.shape[0], (y_test != y_predicted).sum()))
     print("The resultant accuracy of the Gaussian Naive Bayes classifier is: %f" %
           accuracy_score(y_test, y_predicted))
```

```
Number of mislabeled points out of a total 299 points : 20
The resultant accuracy of the Gaussian Naive Bayes classifier is: 0.933110
```

7 Save Model using Pickle

```
[ ]: with open('model.pickle', 'wb') as f:
         pickle.dump(classifier, f)
```