Лабораторна робота №6
Наївний Байєс в Python

Мета роботи: набути навичок працювати з даними і опонувати роботу у Python з використанням теореми Байэса.

Хід роботи

Завдання 1 - було ознайомлено з теоретичним матеріалом нижче.

Завдання 2 – було ретельно переглянуто шлях дій

Завдання 3. Використовуя данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні

Варіант №24
Варіант 4,9, 14

Outlook = Sunny
Humidity = Normal
Wind = Strong

Ймовірність Yes в цей день = P(Outlook = Sunny|Yes) * P(Humidity = Normal|Yes) * P(Wind = Strong|Yes) * P(Yes) = 3/10 * 6/9 * 3/9 * 9/14 = 0.0428

Ймовірність No в цей день = P(Outlook = Sunny|No) * P(Humidity = Normal|No) * P(Wind = Strong|No) * P(No) = 2/4 * 1/5 * 3/5 * 5/14 = 0.0214

Після нормалізації маємо:
P(Yes) = 0.0428 / (0.0428 + 0.0214) = 0.6(6)
P(No) = 0.0214 / (0.0428 + 0.0214) = 0.3(3)

Ймовірність того що матч відбудеться близько 67%.

Завдання 4. Застосуйте методи байєсівського аналізу до набору даних про ціни на квитки на іспанські високошвидкісні залізниці.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

```python
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report

url = "https://raw.githubusercontent.com/susanli2016/Machine-Learning-with-Python/master/data/renfe_small.csv"
df = pd.read_csv(url)
df = df.dropna(subset=["price"])
df['price'] = pd.to_numeric(df['price'], errors='coerce')
df = df.dropna(subset=['price']) # Уникаємо NaN після перетворення
df['price_category'] = pd.cut(df['price'], bins=[0, 40, 70, float('inf')], labels=["low", "medium", "high"])
le = LabelEncoder()
categorical_columns = ["origin", "destination", "train_type", "train_class", "fare"]
for col in categorical_columns:
df[col] = le.fit_transform(df[col])
X = df[["origin", "destination", "train_type", "train_class", "fare"]]
y = df["price_category"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
model = MultinomialNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
~/Desktop/zp/ai/laba6 $ python3 main.py
              precision    recall  f1-score   support

        high       0.71      0.72      0.71      2565
         low       0.58      0.35      0.44      1508
      medium       0.53      0.65      0.58      2742

    accuracy                           0.61      6815
   macro avg       0.61      0.57      0.58      6815
weighted avg       0.61      0.61      0.60      6815
```

Опис роботи:

- Зчитуємо файл
- Очищуємо рядки в яких нема ціни
- Категоризуємо дані. Ціна від 0 до 40 - Low. Ціна від 40 до 70 - Medium. Ціна вище 70 - High
- Ділимо дані на тренувальні та тестові
- Навчаємо Баєсівський класифікартор MultinomialNB
- Виводимо звіт класифікації

Точність моделі 61%. Також бачимо що вона краще справляється з прогнозуванням високих цін. З малими цінами є певна проблема.

Посилання на GitHub: https://github.com/missShevel/SHI_Shevel_Olha_IPZ-21-1/tree/master/Lab6