Системи штучного інтелекту.

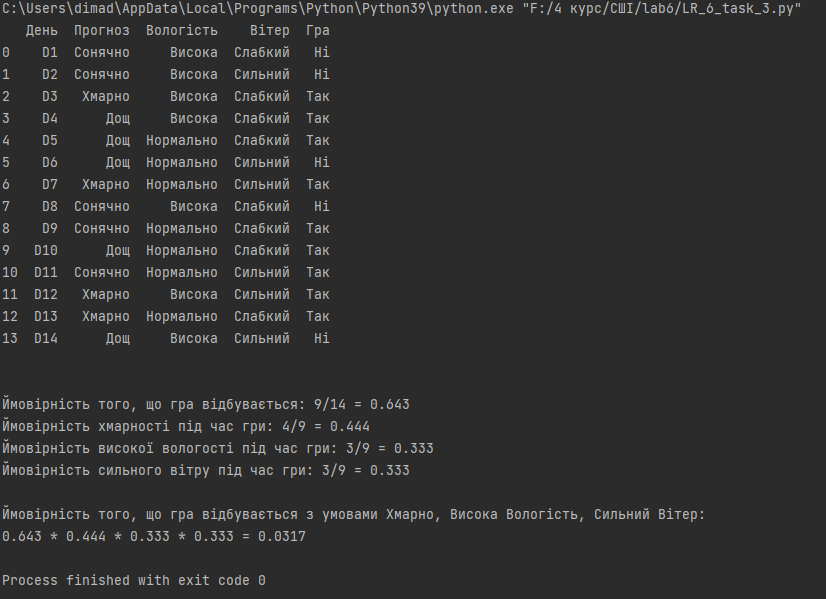
Лабораторна робота 6.Федорович Дмитро ІПЗ-21-3

<https://github.com/Dmitrij3/lab6AI>

**Завдання №3.** Використовуя данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні: Розрахунки провести з використанням Python.

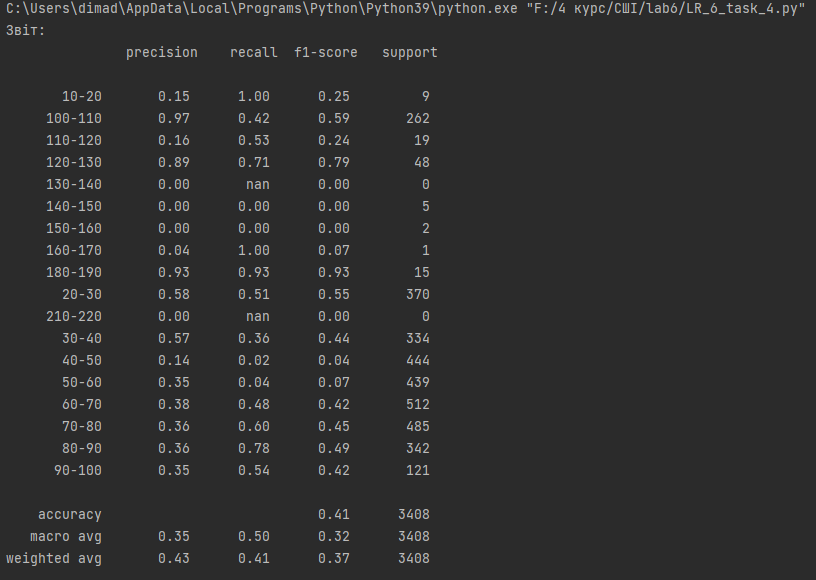


from collections import Counter  
import pandas as pd  
  
data = [  
 {"День": "D1", "Прогноз": "Сонячно", "Вологість": "Висока", "Вітер": "Слабкий", "Гра": "Ні"},  
 {"День": "D2", "Прогноз": "Сонячно", "Вологість": "Висока", "Вітер": "Сильний", "Гра": "Ні"},  
 {"День": "D3", "Прогноз": "Хмарно", "Вологість": "Висока", "Вітер": "Слабкий", "Гра": "Так"},  
 {"День": "D4", "Прогноз": "Дощ", "Вологість": "Висока", "Вітер": "Слабкий", "Гра": "Так"},  
 {"День": "D5", "Прогноз": "Дощ", "Вологість": "Нормально", "Вітер": "Слабкий", "Гра": "Так"},  
 {"День": "D6", "Прогноз": "Дощ", "Вологість": "Нормально", "Вітер": "Сильний", "Гра": "Ні"},  
 {"День": "D7", "Прогноз": "Хмарно", "Вологість": "Нормально", "Вітер": "Сильний", "Гра": "Так"},  
 {"День": "D8", "Прогноз": "Сонячно", "Вологість": "Висока", "Вітер": "Слабкий", "Гра": "Ні"},  
 {"День": "D9", "Прогноз": "Сонячно", "Вологість": "Нормально", "Вітер": "Слабкий", "Гра": "Так"},  
 {"День": "D10", "Прогноз": "Дощ", "Вологість": "Нормально", "Вітер": "Слабкий", "Гра": "Так"},  
 {"День": "D11", "Прогноз": "Сонячно", "Вологість": "Нормально", "Вітер": "Сильний", "Гра": "Так"},  
 {"День": "D12", "Прогноз": "Хмарно", "Вологість": "Висока", "Вітер": "Сильний", "Гра": "Так"},  
 {"День": "D13", "Прогноз": "Хмарно", "Вологість": "Нормально", "Вітер": "Слабкий", "Гра": "Так"},  
 {"День": "D14", "Прогноз": "Дощ", "Вологість": "Висока", "Вітер": "Сильний", "Гра": "Ні"},  
]  
  
print(pd.DataFrame(data))  
  
play\_yes\_count = sum(1 for row in data if row["Гра"] == "Так")  
play\_count = len(data)  
play\_yes\_prob = play\_yes\_count / play\_count  
print("\n\nЙмовірність того, що гра відбувається: {0}/{1} = {2}".format(play\_yes\_count, play\_count, round(play\_yes\_prob, 3)))  
  
overcast\_yes\_count = sum(1 for row in data if row["Гра"] == "Так" and row["Прогноз"] == "Хмарно")  
overcast\_yes\_prob = overcast\_yes\_count / play\_yes\_count  
print("Ймовірність хмарності під час гри: {0}/{1} = {2}".format(overcast\_yes\_count, play\_yes\_count, round(overcast\_yes\_prob, 3)))  
  
humidity\_high\_yes\_count = sum(1 for row in data if row["Гра"] == "Так" and row["Вологість"] == "Висока")  
humidity\_high\_yes\_prob = humidity\_high\_yes\_count / play\_yes\_count  
print("Ймовірність високої вологості під час гри: {0}/{1} = {2}".format(humidity\_high\_yes\_count, play\_yes\_count, round(humidity\_high\_yes\_prob, 3)))  
  
wind\_strong\_yes\_count = sum(1 for row in data if row["Гра"] == "Так" and row["Вітер"] == "Сильний")  
wind\_strong\_yes\_prob = wind\_strong\_yes\_count / play\_yes\_count  
print("Ймовірність сильного вітру під час гри: {0}/{1} = {2}".format(wind\_strong\_yes\_count, play\_yes\_count, round(wind\_strong\_yes\_prob, 3)))  
  
overall\_probability = play\_yes\_prob \* overcast\_yes\_prob \* humidity\_high\_yes\_prob \* wind\_strong\_yes\_prob  
print("\nЙмовірність того, що гра відбувається з умовами Хмарно, Висока Вологість, Сильний Вітер:"  
 " \n{:.3f} \* {:.3f} \* {:.3f} \* {:.3f} = {:.4f}".format(play\_yes\_prob, overcast\_yes\_prob, humidity\_high\_yes\_prob, wind\_strong\_yes\_prob, overall\_probability))

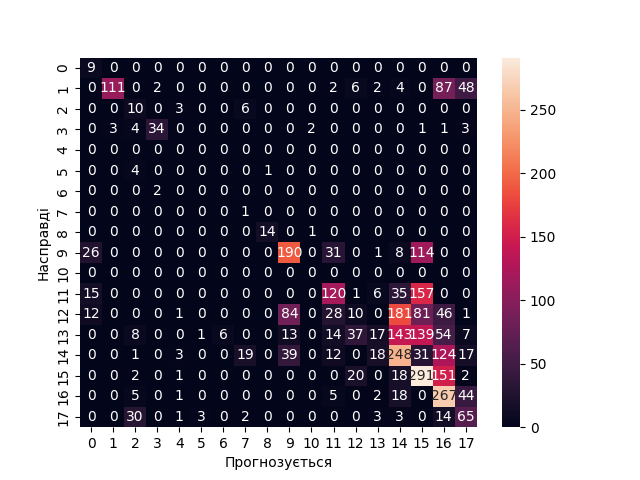
****

**Завдання №4.** Застосуєте методи байєсівського аналізу до набору даних про ціни на квитки на іспанські високошвидкісні залізниці

from sklearn.model\_selection import train\_test\_split  
from sklearn.naive\_bayes import GaussianNB  
import pandas as pd  
from sklearn.model\_selection import cross\_val\_score  
from sklearn.metrics import classification\_report  
from sklearn.metrics import confusion\_matrix  
from sklearn.preprocessing import LabelEncoder  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
  
df = pd.read\_csv('renfe\_small.csv')  
df = df.dropna()  
  
max\_val = round(df['price'].max()) + 10  
bins = range(0, max\_val, 10)  
labels = [f"{i}-{i+10}" for i in range(0, max\_val-10, 10)]  
df['price\_range'] = pd.cut(df['price'], bins=bins, right=False, labels=labels)  
y = df['price\_range']  
  
label\_encoders = {}  
for column in ['origin', 'destination', 'train\_type', 'train\_class', 'fare']:  
 le = LabelEncoder()  
 df[column] = le.fit\_transform(df[column])  
 label\_encoders[column] = le  
  
df['start\_day\_of\_week'] = pd.to\_datetime(df['start\_date']).dt.dayofweek  
df['duration'] = pd.to\_datetime(df['end\_date']) - pd.to\_datetime(df['start\_date'])  
df['duration\_minutes'] = (df['duration'].dt.total\_seconds()//60).astype(int)  
X = df[['origin', 'destination', 'start\_day\_of\_week', 'duration\_minutes', 'train\_type', 'train\_class', 'fare']]  
y = df['price\_range']  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.15, random\_state=5)  
  
model = GaussianNB()  
model.fit(X\_train, y\_train)  
  
y\_pred = model.predict(X\_test)  
  
print("Звіт:\n", classification\_report(y\_test, y\_pred, zero\_division=np.nan, ))  
scores = cross\_val\_score(model, X, y, cv=5, scoring='accuracy')  
print("Результати перехресної перевірки:", scores)  
print("Середній бал перехресної перевірки:", scores.mean())  
cm = confusion\_matrix(y\_test, y\_pred)  
sns.heatmap(cm, annot=True, fmt='d')  
plt.xlabel('Прогнозується')  
plt.ylabel('Насправді')  
plt.show()

****

****

****