

Системи штучного інтелекту.

Лабораторна робота 6. Федорович Дмитро ІПЗ-21-3

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

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

2, 7, 12	Outlook = Overcast Humidity = High Wind = Strong	Перспектива = Похмуро Вологість = Висока Вітер = Сильний
----------	--	--

```
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("\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)))
```

```
{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))
```

C:\Users\dimad\AppData\Local\Programs\Python\Python39\python.exe "F:/4 курс/СШІ/lab6/LR_6_task_3.py"

	День	Прогноз	Вологість	Вітер	Гра
0	D1	Сонячно	Висока	Слабкий	Ні
1	D2	Сонячно	Висока	Сильний	Ні
2	D3	Хмарно	Висока	Слабкий	Так
3	D4	Дощ	Висока	Слабкий	Так
4	D5	Дощ	Нормально	Слабкий	Так
5	D6	Дощ	Нормально	Сильний	Ні
6	D7	Хмарно	Нормально	Сильний	Так
7	D8	Сонячно	Висока	Слабкий	Ні
8	D9	Сонячно	Нормально	Слабкий	Так
9	D10	Дощ	Нормально	Слабкий	Так
10	D11	Сонячно	Нормально	Сильний	Так
11	D12	Хмарно	Висока	Сильний	Так
12	D13	Хмарно	Нормально	Слабкий	Так
13	D14	Дощ	Висока	Сильний	Ні

Ймовірність того, що гра відбувається: 9/14 = 0.643
Ймовірність хмарності під час гри: 4/9 = 0.444
Ймовірність високої вологості під час гри: 3/9 = 0.333
Ймовірність сильного вітру під час гри: 3/9 = 0.333

Ймовірність того, що гра відбувається з умовами Хмарно, Висока Вологість, Сильний Вітер:
0.643 * 0.444 * 0.333 * 0.333 = 0.0317

Process finished with exit code 0

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()

```

C:\Users\dimad\AppData\Local\Programs\Python\Python39\python.exe "F:/4 курс/СШІ/lab6/LR_6_task_4.py"

Звіт:

	precision	recall	f1-score	support
10-20	0.15	1.00	0.25	9
100-110	0.97	0.42	0.59	262
110-120	0.16	0.53	0.24	19
120-130	0.89	0.71	0.79	48
130-140	0.00	nan	0.00	0
140-150	0.00	0.00	0.00	5
150-160	0.00	0.00	0.00	2
160-170	0.04	1.00	0.07	1
180-190	0.93	0.93	0.93	15
20-30	0.58	0.51	0.55	370
210-220	0.00	nan	0.00	0
30-40	0.57	0.36	0.44	334
40-50	0.14	0.02	0.04	444
50-60	0.35	0.04	0.07	439
60-70	0.38	0.48	0.42	512
70-80	0.36	0.60	0.45	485
80-90	0.36	0.78	0.49	342
90-100	0.35	0.54	0.42	121
accuracy			0.41	3408
macro avg	0.35	0.50	0.32	3408
weighted avg	0.43	0.41	0.37	3408

Результати перехресної перевірки: [0.4159331 0.41316311 0.41052168 0.41294299 0.41624477]

Середній бал перехресної перевірки: 0.41376112995383707

