№11 ЗЕРТХАНАЛЫҚ ЖҰМЫС
Машиналық оқыту моделіне арналған деректерді дайындау

Пән: Жасанды интеллектке арналған Python (Python for AI)
Ұзақтығы: 2 академиялық сағат
Құралдар: Google Colab, Python, Pandas, scikit-learn

1. Жұмыстың мақсаты
Машиналық оқытудың негізгі ұғымдарымен танысу, көлік жүйелерінің деректерін кіріс белгілері (Features) мен мақсатты айнымалыға (Target) бөлу, санаттық деректерді сандық түрге келтіру, деректерді оқыту және тестілеу жиындарына бөлу, сондай-ақ машиналық оқыту моделіне арналған деректерді дайындауды үйрену.

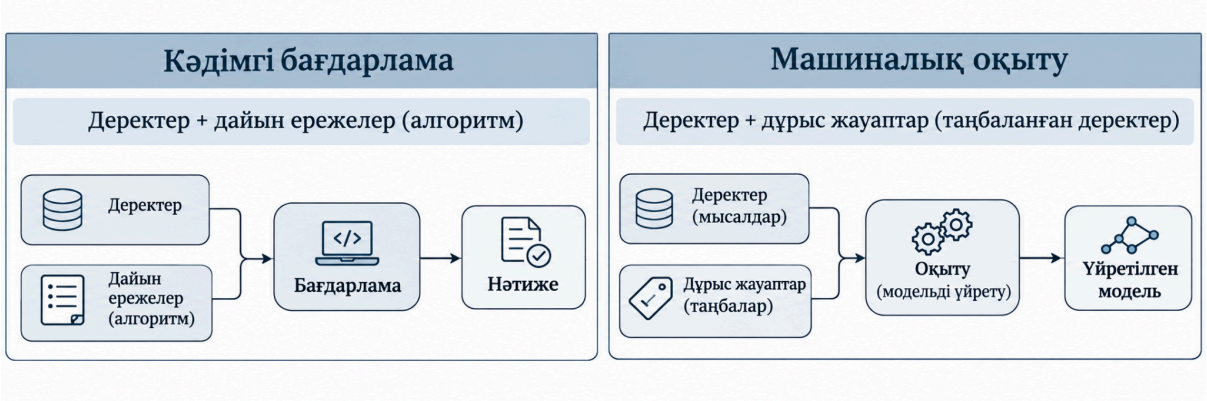
2. Оқыту нәтижелері
Зертханалық жұмысты орындағаннан кейін білім алушы:
- Машиналық оқытудың (Machine Learning) негізгі жұмыс принципін түсіндіре алады;
- Features және Target ұғымдарын ажырата алады;
- Машиналық оқытуға арналған деректер жиынын анықтай алады;
- Санаттық деректерді сандық түрге түрлендіре алады;
- train_test_split() көмегімен деректерді бөле алады;
- StandardScaler көмегімен сандық белгілерді масштабтай алады;
- Дайындалған деректерді келесі зертханалық жұмыста модель құру үшін қолдана алады.

3. Қысқаша теориялық мәліметтер
3.1. Машиналық оқыту дегеніміз не?
Машиналық оқыту (Machine Learning, ML) — компьютердің дайын деректерден заңдылықтарды анықтап, сол заңдылықтар негізінде жаңа деректерге болжам жасауына мүмкіндік беретін әдістер жиынтығы.
Бұрынғы зертханалық жұмыстарда біз шарттарды өзіміз жаздық.
Мысалы:

```
temperature = 50
if temperature > 45:
   print("HIGH")
else:
   print("NORMAL")
```

Бұл бағдарламада температураны бағалау ережесін бағдарламашы өзі анықтайды.

Ал машиналық оқытуда компьютерге көптеген мысалдар беріледі. Модель сол деректердегі байланысты үйреніп, жаңа мәндер үшін болжам жасайды.



Бұл жұмыста біз модельді әлі үйретпейміз. Алдымен оған қажетті деректерді дайындаймыз.

3.2. Features және Target дегеніміз не?
Мысалы, электрмобильдің энергия тұтынуын болжағымыз келеді.

Speed (км/сағ)	Temperature (°C)	SOC (%)	Mode	Energy (кВт·сағ/100 км)
40	20	85	Eco	12.5
60	25	75	Normal	15.2
80	30	65	Sport	19.1
100	35	55	Sport	23.4
50	22	80	Eco	13.8

Мұнда:

- Features (X) — модельге берілетін кіріс параметрлері: Speed, Temperature, SOC, Mode.
- Target (y) — болжағымыз келетін нәтиже: Energy.

4. Жұмысты орындау

Тапсырма №1. Google Colab ортасын дайындау

1. Google Colab сайтына өтіңіз.
2. Жаңа Notebook құрыңыз.
3. Файлды Lab11_Surname деп атаңыз.
4. Кітапханаларды импорттаңыз.

```
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

Тапсырма №2. Деректер жиынын оқу
Алдымен жұмыстың орындалуын үйрену үшін шағын мысалдық деректер жиынын пайдаланамыз. Жеке тапсырмада оқытушы берген CSV файлы қолданылады.

```
data = {
   "Speed": [40, 60, 80, 100, 50, 70, 90, 55, 75, 95],
   "Temperature": [20, 25, 30, 35, 22, 28, 33, 24, 29, 34],
   "SOC": [85, 75, 65, 55, 80, 70, 60, 78, 68, 58],
   "Mode": ["Eco", "Normal", "Sport", "Sport", "Eco",
         "Normal", "Sport", "Eco", "Normal", "Sport"],
   "Energy": [12.5, 15.2, 19.1, 23.4, 13.8,
           16.7, 21.3, 14.1, 17.8, 22.5]
}

df = pd.DataFrame(data)
display(df)
```

Тапсырма №3. Features және Target бөлу

```
X = df.drop(columns=["Energy"])
y = df["Energy"]

print("Features:")
display(X.head())

print("Target:")
display(y.head())
```

Тапсырма №4. Санаттық деректерді сандық түрге түрлендіру
Mode бағанында Eco, Normal, Sport мәтіндік мәндері бар. Оларды бөлек сандық белгілерге айналдырамыз.

```
X = pd.get_dummies(X, columns=["Mode"], dtype=int)

display(X.head())
```

Нәтижесінде Mode_Eco, Mode_Normal, Mode_Sport сияқты жаңа бағандар пайда болады. Әр жолда тиісті режим 1, қалғандары 0 мәнін алады.

Тапсырма №5. Деректерді оқыту және тестілеу жиындарына бөлу
Модельді оқыту және оның жаңа деректермен жұмысын тексеру үшін деректер екі бөлікке бөлінеді.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42
)

print("Оқыту деректері:", X_train.shape)
print("Тестілеу деректері:", X_test.shape)
```

10 жолдың 8-і оқытуға, 2-і тестілеуге бөлінеді. Бұл тек әдісті көрсетуге арналған шағын мысал; нақты модельді сенімді бағалау үшін көбірек дерек қажет.

random_state параметрі дегеніміз не?
train_test_split() функциясы деректерді оқыту және тестілеу жиындарына кездейсоқ түрде бөледі.
Егер random_state параметрі көрсетілмесе, бағдарламаны әр іске қосқан сайын деректер әртүрлі бөлінуі мүмкін. Яғни бір өлшеу бірінші іске қосқанда оқыту жиынына, ал келесі іске қосқанда тестілеу жиынына түсуі мүмкін.
random_state — кездейсоқ бөлудің нәтижесін қайталап алуға мүмкіндік беретін параметр.
test_size=0.2 — деректердің 20%-ы тестілеуге, қалған 80%-ы оқытуға бөлінеді.
random_state=42 — бағдарлама қайта іске қосылғанда, бастапқы деректер мен параметрлер өзгермесе, деректердің дәл солай бөлінуін қамтамасыз етеді.
Неліктен 42 саны қолданылады?
42 саны міндетті емес. Оның машиналық оқытуда арнайы математикалық артықшылығы жоқ.
Бұл — бағдарламашылар арасында кеңінен танымал сан. Ол «Галактикаға автостоппен саяхат» (*The Hitchhiker's Guide to the Galaxy*) атты кітаптың

арқасында танымал болды. Бұл шығармада 42 саны өмірдің, Ғаламның және барлық нәрсенің басты сұрағына жауап ретінде көрсетіледі.
Оның орнына кез келген теріс емес бүтін санды қолдануға болады. Мысалы:
random_state=10
random_state=100
random_state=2026

random_state параметрі не үшін қажет?
Бұл параметр әсіресе зертханалық жұмыстарда пайдалы. Мысалы, оқытушы мен студент бірдей CSV файлын және random_state=42 мәнін қолданса, деректер бірдей бөлінеді. Осылайша алынған нәтижелерді салыстыру және тексеру жеңілдейді.
Қорытынды: test_size деректердің қанша бөлігі тестілеуге бөлінетінін анықтайды, ал random_state кездейсоқ бөлудің қайталануын қамтамасыз етеді.

Тапсырма №6. Сандық белгілерді масштабтау
Speed, Temperature, SOC мәндерінің өлшем бірліктері мен диапазондары әртүрлі. StandardScaler оларды орташа мәні 0-ге, стандартты ауытқуы 1-ге жуық болатындай түрлендіреді.

```
numeric_cols = ["Speed", "Temperature", "SOC"]

scaler = StandardScaler()

X_train = X_train.copy()
X_test = X_test.copy()

X_train[numeric_cols] = scaler.fit_transform(
    X_train[numeric_cols]
)

X_test[numeric_cols] = scaler.transform(
    X_test[numeric_cols]
)

display(X_train.head())
```

Маңызды: fit_transform() тек оқыту деректеріне қолданылады. Тест деректеріне transform() қолданылады. Бұл тест жиынындағы ақпараттың оқыту кезеңіне өтіп кетуін болдырмайды.

Тапсырма №7. Нәтижелерді тексеру

```
print("X_train:", X_train.shape)
print("X_test:", X_test.shape)
print("y_train:", y_train.shape)
print("y_test:", y_test.shape)

display(X_train.head())
```

Жеке тапсырма

Әр студент оқытушы берген жеке CSV нұсқасымен жұмыс істейді.
Жеке тапсырмада:

1. CSV файлын Google Colab ортасына жүктеңіз.
2. Features және Target анықтаңыз.
3. Санаттық белгілерді кодтаңыз.
4. Деректерді 80% және 20% үлесімен бөліңіз.
5. Сандық белгілерді масштабтаңыз.
6. Оқыту және тестілеу жиындарының өлшемдерін көрсетіңіз.
7. Қысқаша қорытынды жазыңыз.

Бақылау сұрақтары

1. Машиналық оқыту дегеніміз не?
2. Кәдімгі бағдарлама мен машиналық оқытудың айырмашылығы қандай?
3. Features дегеніміз не?
4. Target дегеніміз не?
5. Неліктен деректерді оқыту және тестілеу жиындарына бөлеміз?
6. train_test_split() қандай қызмет атқарады?
7. Санаттық деректерді не үшін сандық түрге айналдырамыз?
8. StandardScaler не үшін қажет?
9. fit_transform() және transform() айырмашылығы қандай?
10. Неліктен тест деректерін модельді оқыту кезінде пайдалануға болмайды?