



Цифрлық әдіскер

ақпараттық-білім беру ортасы

ДӘРІСТЕР ЖИНАҒЫ

Жасанды интеллект (машиналық оқыту негіздері)

Машиналық оқытудың толық циклі: Python кітапханалары (NumPy, Pandas, Scikit-learn) арқылы деректерді өңдеу, модель құру және бағалау.

Машиналық оқыту циклі

NumPy, Pandas, Scikit-learn

Жіктеу және болжау

Деректерді визуализациялау

ДӘРІС ЖИНАҒЫ

АЛҒЫ СӨЗ

Бұл лекция конспектісі «Жасанды интеллект» пәні бойынша білімдеріңізді тереңдетуге арналған. Қазіргі таңда жасанды интеллект (ЖИ) адамзат дамуының жаңа кезеңіне жол ашып, ғылымнан бастап өндіріс, медицина, қаржы, білім беру, логистика және өзге де салаларда кеңінен қолданыла бастады. Ал машиналық оқыту – осы жасанды интеллекттің негізін құрайтын маңызды бағыттардың бірі. Оқу барысында сіздер Python бағдарламалау тілі мен оның танымал кітапханалары (NumPy, Pandas, Matplotlib, Scikit-learn, т.б.) көмегімен деректерді өңдеу, модель құру, болжау, жіктеу, визуализация және интерпретация әдістерін үйренесіздер. Сонымен қатар, машиналық оқыту жобасының толық циклін (data preprocessing → model building → evaluation → deployment) игеруге мүмкіндік аласыздар. Конспект пәннің әрбір тақырыбы бойынша теориялық мазмұнмен қатар, нақты мысалдар мен графикалық иллюстрацияларды қамтиды. Бұл материал сіздің аналитикалық ойлау қабілетіңізді, алгоритмдік мәдениетіңізді және өздік іздену дағдыларыңызды дамытуға бағытталған. Пәнді табысты меңгеру үшін практикалық тапсырмаларды жүйелі түрде орындап, интерактивті құралдарды (Kaggle, Google Colab, AI-сервистер) белсенді қолдануыңыз ұсынылады.

Лекция № 1

Тақырыбы: Машиналық оқуға кіріспе Жоспары:

- Машиналық оқыту
 - Неліктен машиналық оқыту
 - Машиналық оқытудың түрлері
 - Машиналық оқыту процесінің кезеңдері Лекция мәтіні
 - Машиналық оқыту дегеніміз не? Машиналық оқыту (Machine Learning) – бұл компьютерлердің адам тарапынан нақты программалануынсыз, тәжірибе негізінде (деректерден) үйренуіне мүмкіндік беретін жасанды интеллекттің бағыты. Машиналық оқытуда негізгі мақсат — үлгілерді тану және болжау жасау. Жүйе деректерден заңдылықтарды анықтап, оларды болашақтағы жаңа деректерге қолдана алады. Анықтама: Arthur Samuel (1959):
“Машиналық оқыту – бұл компьютерге бағдарламалаусыз үйренуге мүмкіндік беретін зерттеу саласы.”
 - Неліктен машиналық оқыту қажет? Қиындық Машиналық оқытудың артықшылығы Деректер көлемінің шексіз өсуі
Үлкен деректерді тиімді өңдеу Қолмен жазылған ережелер жеткіліксіз Деректерден өздігінен заңдылық табу Үнемі өзгеретін орта Үлгі жаңарту арқылы бейімделу
1. Машиналық оқытудың түрлері Түрі Сипаттамасы Мысалы Бақылаумен оқыту (Supervised Learning) Енгізу мен шығу деректері белгілі Электрондық хатты спам/спам емес деп анықтау Бақылаусыз оқыту (Unsupervised Learning) Тек енгізу деректері бар, топтарға бөлу Клиенттерді сатып алу қабілетіне қарай кластерлеу Күшейтілген оқыту (Reinforcement Learning) Агенттің ортамен өзара әрекеттесуі Роботтың қозғалысын оқыту, ойын стратегиясы
 1. Машиналық оқыту процесінің кезеңдері $mermaid$ Копировать Редактировать graph TD; A[Мәліметтер жинау] → B[Деректерді алдын-ала өңдеу]; B → C[Модель таңдау]; C → D[Модельді үйрету (train)]; D → E[Бағалау]; E → F[Болжау];
 1. Қолдану салалары Сала Қолданылуы Медицина Диагноз қою, ауруларды болжау Қаржы Несие скорингі, алаяқтықты анықтау Білім Тұлғалық оқу траекториясын жасау Көлік Автопилот, қозғалыс бағытын болжау Сауда Тұтынушыға ұсыныс жасау жүйесі (Recommendation)
 1. Қолданылатын негізгі құралдар Құрал Мақсаты Python ML үшін ең кең таралған тіл NumPy, Pandas Деректерді өңдеу Matplotlib, Seaborn Визуализация Scikit-learn ML алгоритмдерін іске асыру TensorFlow, PyTorch Нейрондық желілер мен терең оқыту
 2. Қарапайым мысал Мәселе: Үй бағасын болжау Деректер: Бөлме саны, аудан, орналасқан жер Шешім: Регрессия моделі көмегімен баға болжау python Копировать Редактировать from sklearn.linear_model import LinearRegression model = LinearRegression() model.fit(X_train, y_train) prediction = model.predict(X_test)

3. Қорытынды Машиналық оқыту – бұл деректерге негізделген болжау жасау, заңдылықтарды үйрену арқылы автоматтандырылған шешім қабылдау жүйелерін құру жолы. Болашақта бұл бағыттың маңыздылығы одан әрі артады, себебі барлық салада деректердің рөлі артып келеді. Өзін-өзі тексеруге арналған сұрақтар
4. Машиналық оқыту мен дәстүрлі бағдарламалаудың айырмашылығы неде?
5. Бақылаумен және бақылаусыз оқытудың айырмашылығы қандай?
6. Машиналық оқыту кезеңдерін атаңыз.
7. Машиналық оқыту қай салаларда қолданылады?
8. Python тілінде қандай кітапханалар қолданылады?

Лекция № 2

Тақырыбы: NumPy кітапханасы Жоспары:

- NumPy
 - NumPy-ды орнату және импорттау
 - Негізгі құрылым: ndarray Лекция мәтіні
 - NumPy дегеніміз не? NumPy (Numerical Python) — Python тілі үшін арналған сандық есептеулерге негізделген ашық кітапхана. Ол көпөлшемді массивтермен (array) жұмыс істеуге және математикалық операцияларды тиімді орындауға мүмкіндік береді. Анықтама: NumPy – векторлық және матрицалық операцияларды жылдам орындайтын, жоғары өнімділігімен ерекшеленетін кітапхана.
 - NumPy-ды орнату және импорттау `bash` КопироватьРедактировать `pip install numpy python` КопироватьРедактировать `import numpy as np`
1. Негізгі құрылым: ndarray NumPy массивінің басты объектісі – ndarray (n-dimensional array), ол біртекті элементтерден тұрады (барлығы бірдей типтегі деректер). Мысал: `python` КопироватьРедактировать `a = np.array([1, 2, 3]) print(a) # [1 2 3]`
 1. Массивтерді жасау әдістері Функция Сипаттамасы Мысал `np.array()` Тізімнен массив жасау `np.array([1, 2, 3])` `np.zeros()` Барлық элементтері 0 массив `np.zeros((2, 3))` `np.ones()` Барлық 1 элементті массив `np.ones((3, 3))` `np.arange()` Аралық бойынша сандар тізбегі `np.arange(0, 10, 2)` `np.linspace()` Белгілі интервалға бөлінген сандар `np.linspace(0, 1, 5)` `np.random.rand()` Кездейсоқ мәндер `np.random.rand(2, 2)`
 1. Массивтің сипаттамалары `python` КопироватьРедактировать `a.shape` # өлшемі (қатар × баған) `a.ndim` # өлшемдер саны (1D, 2D, 3D) `a.size` # барлық элементтер саны `a.dtype` # типі (int, float т.б.)
 1. Массивтермен арифметикалық операциялар `python` КопироватьРедактировать `a = np.array([1, 2, 3]) b = np.array([4, 5, 6]) print(a + b) # [5 7 9] print(a * b) # [4 10 18] print(a ** 2) # [1 4 9]` Ескерту: NumPy операциялары векторлық түрде орындалады (for циклісіз).
 2. Пішін өзгерту және индекстеу `python` КопироватьРедактировать `a = np.array([1, 2, 3], [4, 5, 6]) a.reshape(3, 2) #` Пішінді өзгерту `a[0, 1]` # 1-қатар, 2-баған элементі `a[:, 1]` # Барлық қатарлардан 2-баған
 3. Жиі қолданылатын функциялар Функция Мақсаты `np.mean(a)` Орташа мән `np.std(a)` Стандартты ауытқу `np.sum(a)` Барлық элементтер қосындысы `np.max(a)` / `np.min(a)` Максимум / минимум мән `np.sort(a)` Сұрыптау `np.unique(a)` Бірегей мәндер
 4. Практикалық қолдану Кейс: Машиналық оқытуда NumPy деректерді дайындауға жиі қолданылады. `python` КопироватьРедактировать `from sklearn.linear_model import LinearRegression`

NumPy арқылы дерек

```
X = np.array([[1], [2], [3]])
y = np.array([2, 4, 6])

model = LinearRegression()
model.fit(X, y)
print(model.predict([[4]])) # [8.]
```

1. Визуализациямен бірге қолдану `python` КопироватьРедактировать `import matplotlib.pyplot as plt`

```
x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)

plt.plot(x, y)
plt.title("sin(x) функциясы")
plt.show()
11. Тест сұрақтары (өзіңді тексер)
```

- NumPy массиві мен Python тізімінің айырмашылығы неде?
- np.arange(5) және np.linspace(0, 1, 5) қандай айырмашылығы бар?
- a.shape, a.ndim, a.size нені білдіреді?
- NumPy-де reshape және flatten қайда қолданылады?
- NumPy қай жағдайларда машиналық оқытуда пайдалы?

ҚОРЫТЫНДЫ

NumPy – машиналық оқытудағы деректерді құрылымдаудың, трансформациялаудың және есептеудің негізгі құралы. Бұл кітапхана машиналық оқытуда, ғылыми зерттеулерде және аналитикалық есептерде өте маңызды рөл атқарады.

Лекция № 3

Тақырыбы: Pandas арқылы деректерді өңдеу Жоспары:

- Pandas
- Негізгі құрылымдар
- Деректерді жүктеу және сақтау Лекция мәтіні
- Pandas дегеніміз не? Pandas – Python тіліне арналған ашық бастапқы кодты кітапхана, құрылымды деректерді талдауға, тазартуға, өзгертуге, визуализациялауға және сақтауға арналған. Анықтама: Pandas – кестелік деректерді (CSV, Excel, SQL) оңай өңдеуге арналған жоғары деңгейлі интерфейсті қамтамасыз ететін кітапхана.

1. Негізгі құрылымдар: Series және DataFrame Series – бір өлшемді массив (вектор): python КопироватьРедактировать
import pandas as pd

```
s = pd.Series([10, 20, 30])
print(s)
```

DataFrame – екі өлшемді кесте (қатар × баған):

python КопироватьРедактировать

```
data = {'Аты': ['Асан', 'Ерлан'], 'Жасы': [25, 30]}
df = pd.DataFrame(data)
print(df)
```

1. Деректерді жүктеу және сақтау Функция Сипаттамасы pd.read_csv("файл.csv") CSV-файлды оқу
pd.read_excel("файл.xlsx") Excel файлын оқу df.to_csv("нәтиже.csv") CSV-ға сақтау df.head(n) / df.tail(n) Алғашқы / соңғы n қатарды көру
1. Мәліметтер құрылымын зерттеу python КопироватьРедактировать df.shape # Өлшемі (қатар, баған) df.columns # Баған атаулары df.info() # Мәліметтер типі, бос мәндер df.describe() # Негізгі статистика (орташа, мин, макс)
1. Мәліметтерді өңдеу Баған қосу / өшіру: python КопироватьРедактировать df["Жынысы"] = ["Е", "Ә"] df.drop("Жасы", axis=1, inplace=True) Жолдарды фильтрациялау: python КопироватьРедактировать df[df["Жасы"] > 25] Бағанды қайта атау: python КопироватьРедактировать df.rename(columns={"Аты": "Толық_аты"}, inplace=True)

6. Бос мәндермен жұмыс python КопироватьРедактировать

```
df.isnull().sum()          # Бос мәндерді санау
df.dropna()                # Бос қатарларды алып тастау
df.fillna(value=0)         # Бос мәндерді толтыру
```

7. Топтау және жиынтықтау (groupby)

python КопироватьРедактировать

```
df.groupby("Жынысы")["Жасы"].mean()
```

1. Статистикалық операциялар Функция Сипаттамасы df.mean() Орташа мән df.sum() Барлық мәндердің қосындысы df.min() / df.max() Минимум / максимум df.median() Медиана df.std() Стандартты ауытқу

1. Визуализациямен бірге қолдану python КопироватьРедактировать import matplotlib.pyplot as plt

```
df["Жасы"].hist()
plt.title("Жас бойынша үлестірім")
plt.xlabel("Жас")
plt.ylabel("Саны")
plt.show()
```

1. Pandas + машиналық оқыту Машиналық оқытуда Pandas жиі:
2. Деректерді оқу және зерттеу
3. Мақсатты бағанды бөлу
4. Формат түрлендіру
5. Мәліметті алдын ала өңдеу (preprocessing) үшін қолданылады. python КопироватьРедактировать X = df.drop("Нәтиже", axis=1) y = df["Нәтиже"]
6. Өзін-өзі тексеру сұрақтары
7. Series мен DataFrame айырмашылығы қандай?
8. groupby() әдісі қандай мақсатта қолданылады?
9. Бос мәндерді қалай анықтап, жоюға болады?
10. Қай функция мәліметтер жиынын сипаттайды?
11. Pandas қай кезеңде машиналық оқытуда пайдаланылады?

ҚОРЫТЫНДЫ

Pandas – Python тілінде деректермен жұмыс істеудің ең қуатты және ыңғайлы құралы. Ол арқылы деректерді оңай оқып, өңдеп, талдап және машиналық оқыту моделіне дайындауға болады.

Лекция № 4

Тақырыбы: Matplotlib көмегімен визуализация Жоспары:

- Matplotlib
- Орнату және импорттау
- Негізгі график түрлері Лекция мәтіні
- Matplotlib дегеніміз не? Matplotlib — Python тілі үшін арналған мәліметтер визуализациясы кітапханасы. Ол 2D графиктер мен диаграммалар салуға мүмкіндік береді. Анықтама: Matplotlib — ғылыми есептеулерде қолданылатын құрамдастырылған графикалық құрал, яғни диаграмма, гистограмма, сызықтық график, нүктелік график, т.б. визуализация түрлерін жасауға арналған.

- Орнату және импорттау `bash` КопироватьРедактировать `pip install matplotlib python` КопироватьРедактировать `import matplotlib.pyplot as plt`
- Негізгі график түрлері Визуализация Қолдану мысалы Сызықтық график (Line plot) Уақыттық деректер, тренд
Гистограмма (Histogram) Бөлініс, таралу Диаграмма (Bar plot) Категорияларды салыстыру Нүктелік график (Scatter plot) Екі айнымалы арасындағы байланыс Дөңгелек диаграмма (Pie chart) Пропорцияларды көрсету

1. Қарапайым график мысалы python КопироватьРедактировать `import matplotlib.pyplot as plt`

```
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]

plt.plot(x, y)
plt.title("Сызықтық график")
plt.xlabel("X мәні")
plt.ylabel("Y мәні")
plt.grid(True)
plt.show()

5. Гистограмма (histogram)
```

python КопироватьРедактировать

```
data = [22, 30, 35, 30, 40, 50, 45, 40, 35]

plt.hist(data, bins=5, color='skyblue')
plt.title("Жас бойынша таралу")
plt.xlabel("Жас")
plt.ylabel("Саны")
plt.show()

6. Баған диаграммасы (bar plot)
```

python КопироватьРедактировать

```
cities = ['Алматы', 'Астана', 'Шымкент']
population = [1800000, 1200000, 1000000]

plt.bar(cities, population, color='orange')
plt.title("Қалалар бойынша халық саны")
plt.show()

7. Нүктелік график (scatter plot)
```

python КопироватьРедактировать

```
import numpy as np

x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y, color='purple')
plt.title("Нүктелік график")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()

8. Дөңгелек диаграмма (pie chart)
```

python КопироватьРедактировать

```
labels = ['Python', 'C++', 'Java', 'Others']
sizes = [40, 25, 20, 15]

plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title("Бағдарламалау тілдерінің үлесі")
plt.show()
```

1. Графикті баптау элементтері Элемент Функциясы plt.title(...) Атауы plt.xlabel(...), plt.ylabel(...) Осьтер атауы plt.legend() Аңыз (легенда) plt.grid(True) Торлар plt.style.use('ggplot') Стиль қосу

1. Визуализация және машиналық оқыту
2. Графиктер модельге дейінгі деректі түсіну үшін қажет
3. Outlier (шектен тыс мәндер) мен үлгілерді көзбен тануға көмектеседі
4. Модель нәтижесін интерпретациялау үшін қолданылады (accuracy curve, confusion matrix visualization)
5. Практикалық қолдану мысалы python КопироватьРедактировать import pandas as pd

```
df = pd.read_csv("students.csv")
plt.bar(df['Мамандық'], df['GPA'])
plt.title("GPA көрсеткіші мамандық бойынша")
plt.xticks(rotation=45)
plt.show()
```

1. Өзін-өзі тексеру сұрақтары
2. plot() мен scatter() айырмашылығы неде?
3. hist() функциясы не үшін қолданылады?
4. Графикті баптау үшін қандай командалар бар?
5. Pie chart пен bar chart айырмашылығы қандай?
6. Визуализация дерек ғылымында не үшін маңызды?

ҚОРЫТЫНДЫ

Matplotlib – Python тіліндегі ең кең таралған визуализация құралы. Ол сіздің деректермен жұмыс жасауыңызды жеңілдетіп, ақпаратты көрнекі және интуитивті түрде ұсынуға көмектеседі.

Лекция № 5

Тақырыбы: Seaborn кітапханасы Жоспары:

- Seaborn
 - Негізгі график түрлері
 - Қарапайым мысалдар Лекция мәтіні
 - Seaborn дегеніміз не? Seaborn — бұл Python үшін арналған жоғары деңгейлі визуализация кітапханасы, ол Matplotlib негізінде жасалған және деректермен статистикалық графиктерді оңай және стильді түрде жасауға мүмкіндік береді. Анықтама: Seaborn – мәліметтер арасындағы қатынастарды визуалды түрде көрсетуге арналған кітапхана. Ол DataFrame құрылымын тікелей қолдайды және стильдік өңдеулерді автоматты түрде орындайды.
1. Орнату және импорттау bash КопироватьРедактировать pip install seaborn python КопироватьРедактировать import seaborn as sns import matplotlib.pyplot as plt
 2. Деректермен жұмыс істеу Seaborn көбінесе Pandas DataFrame құрылымымен жұмыс істейді. Ішкі дайын деректер жиыны да бар: python КопироватьРедактировать df = sns.load_dataset("tips") # Кафеде шот бойынша деректер df.head()
 3. Негізгі график түрлері График түрі Функция Мақсаты Сызықтық график sns.lineplot() Уақыттық өзгерістер Жолақ диаграмма sns.barplot() Орташа мәндерді салыстыру Нүктелік график sns.scatterplot() Айнымалылар арасындағы

байланыс Гистограмма `sns.histplot()` Таралу үлгісі Box-график `sns.boxplot()` Мәндердің медианасы мен ауытқулары
Heatmap `sns.heatmap()` Корреляцияны, матрицаны көрсету Pairplot `sns.pairplot()` Бірнеше баған арасындағы қатынасты көрсету

1. Қарапайым мысалдар

Нүктелік график (scatter plot)

python КопироватьРедактировать

```
sns.scatterplot(data=df, x="total_bill", y="tip", hue="sex")
plt.title("Шот пен шайпұл арасындағы байланыс")
plt.show()
Жолақ график (bar plot)
```

python КопироватьРедактировать

```
sns.barplot(data=df, x="day", y="total_bill", ci=None)
plt.title("Күндер бойынша шот сомасы")
plt.show()
Box plot (қорапша графигі)
```

python КопироватьРедактировать

```
sns.boxplot(data=df, x="day", y="tip", hue="sex")
plt.title("Шайпұлдың таралуы")
plt.show()
```

1. Стилль мен баптау

python КопироватьРедактировать

```
sns.set_style("whitegrid") # немесе: darkgrid, white, ticks
sns.set_palette("pastel") # түс схемасы
```

♦

Стилльдер мысалы: python КопироватьРедактировать

```
sns.set_context("notebook") # немесе: talk, paper, poster
```

1. Қосымша мысал: Pairplot және Heatmap Pairplot – толық жұптық қатынасты көрсету python КопироватьРедактировать
`sns.pairplot(df, hue="sex")` `plt.suptitle("Жұптық салыстыру", y=1.02)` `plt.show()` Heatmap – корреляциялық матрица
python КопироватьРедактировать `corr = df.corr()` `sns.heatmap(corr, annot=True, cmap="coolwarm")`
`plt.title("Айнымалылар арасындағы корреляция")` `plt.show()`

1. Seaborn vs Matplotlib Салыстыру Matplotlib Seaborn Интерфейс Төмен деңгейлі Жоғары деңгейлі Деректермен жұмыс
Массивке негізделген DataFrame негізінде Стилль Қарапайым, қолмен баптау Автоматты стилль, кеңейтілген баптау
Пайдалану жеңілдігі Икемді, бірақ күрделі Интуитивті, тез үйренуге болады

1. Машиналық оқытуда қолдану Seaborn көмегімен:

2. Алдын ала зерттеу (EDA) кезінде айнымалылар арасындағы қатынасты анықтау,

3. Модель нәтижелерін визуалдау,

4. Класстардың таралуын тексеру,

5. Outlier-ді табу жүзеге асады.

6. Өзін-өзі тексеру сұрақтары
7. Seaborn мен Matplotlib айырмашылығы қандай?
8. sns.heatmap() қандай жағдайда қолданылады?
9. Жыныс пен шайпұлдың байланысын қай графикпен көрсетуге болады?
10. Seaborn стилін қалай баптауға болады?
11. Pairplot визуализациясы не үшін қажет?

ҚОРЫТЫНДЫ

Seaborn — Python тіліндегі әдемі, кәсіби көрінетін графиктерді оңай жасауға мүмкіндік беретін қуатты визуализация құралы. Ол машиналық оқытудың барлық кезеңдерінде — деректі талдау, модель интерпретациясы және нәтижені ұсынуда кеңінен қолданылады.

Лекция № 6

Тақырыбы: Деректерді алдын ала өңдеу (Data Preprocessing) Жоспары:

- Неліктен деректерді алдын ала өңдеу қажет
- Алдын ала өңдеудің негізгі кезеңдері
- Бос мәндермен (missing values) жұмыс Лекция мәтіні
- Неліктен деректерді алдын ала өңдеу қажет? Машиналық оқытуда модельдің сапасы — берілген деректердің сапасына тікелей байланысты. Көп жағдайда деректер:
 - Бос мәндерден (missing values),
 - Артық/қайталанатын жазбалардан,
 - Бірлік шкалалардан айырмашылығы,
- Сандық емес (категориялық) белгілерден тұрады. Модель тиімді жұмыс істеуі үшін деректерді тазарту, трансформациялау, шкалалау сияқты өңдеу кезеңдері қажет.
- Алдын ала өңдеудің негізгі кезеңдері mermaid КопироватьРедактировать graph TD; A[Деректерді жүктеу] → B[Бос мәндерді өңдеу]; B → C[Категориялық айнымалыларды түрлендіру]; C → D[Шкалалау және нормализация]; D → E[Outlier өңдеу]; E → F[Мәліметті модельге дайындау];
- Бос мәндермен (missing values) жұмыс Тексеру: python КопироватьРедактировать df.isnull().sum() Өшіру: python КопироватьРедактировать df.dropna() Толтыру: python КопироватьРедактировать df.fillna(value=0) # тұрақты мән df["age"].fillna(df["age"].mean()) # орташа мәнмен
- Категориялық айнымалыларды өңдеу Машиналық оқыту алгоритмдері сандық деректермен жұмыс істейді, сондықтан мәтіндік (категориялық) мәндерді түрлендіру қажет. Label Encoding: python КопироватьРедактировать from sklearn.preprocessing import LabelEncoder le = LabelEncoder() df["gender"] = le.fit_transform(df["gender"]) One-Hot Encoding: python КопироватьРедактировать pd.get_dummies(df["city"])
- Мәліметті шкалалау (scaling) Шкалалау — айнымалылардың ортақ шкалаға келтірілуі. Бұл әсіресе градиентті әдістер мен қашықтыққа негізделген алгоритмдерде маңызды. Стандартты шкалалау (StandardScaler): python КопироватьРедактировать from sklearn.preprocessing import StandardScaler sc = StandardScaler() X_scaled = sc.fit_transform(X) Мин-Макс нормализация (MinMaxScaler): python КопироватьРедактировать from sklearn.preprocessing import MinMaxScaler scaler = MinMaxScaler() X_norm = scaler.fit_transform(X)
- Outlier (шектен тыс мәндер) өңдеу
- Boxplot арқылы көріп, алып тастауға немесе шектеуге болады.
- Z-score немесе IQR (Interquartile Range) әдістері қолданылады. python КопироватьРедактировать from scipy import stats z = np.abs(stats.zscore(df["income"])) df = df[(z < 3)]
- Модельге дайын дерек жасау python КопироватьРедактировать X = df.drop("target", axis=1) # Түсіндіруші айнымалылар y = df["target"] # Мақсатты айнымалы

Деректі бөлу

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

1. Қосымша кеңестер Қиындық Шешімі Санаттар саны көп Rare label encoding немесе біріктіру Бос мәндер көп Мәндердің медианасы, арнайы «unknown» санат Ерекше мәндер Логарифмдік түрлендіру (log1p) Мәлімет теңгерілмеген SMOTE немесе класстық баланс техникалары
1. Өзін-өзі тексеру сұрақтары
2. Бос мәндерді өңдеудің қандай жолдары бар?
3. LabelEncoder мен OneHotEncoder айырмашылығы қандай?
4. Қандай жағдайларда шкалалау міндетті?
5. Outlier өңдеу не үшін маңызды?
6. Машиналық оқытуда қандай деректер дайын болады?

ҚОРЫТЫНДЫ

Деректерді алдын ала өңдеу — машиналық оқытудың құрылыс іргетасы. Бұл кезең дұрыс орындалмаса, тіпті ең қуатты модель де дұрыс нәтиже бермейді. Сондықтан тазалық, бірізділік және шкалалық үйлесімділік — басты принцип.

Лекция № 7

Тақырыбы: Feature Engineering – Белгілерді құрастыру Жоспары:

- Feature Engineering
 - Неліктен Feature Engineering маңызды
 - Feature Engineering кезеңдері Лекция мәтіні
 - Feature Engineering дегеніміз не? Feature Engineering – деректерден модельге тиімді, мағыналы белгілерді (features) алу, өңдеу немесе жаңасын құрастыру процесі. Анықтама: Feature Engineering – машиналық оқытуда алгоритмдердің жұмысын жақсарту үшін деректерді түрлендіру, біріктіру, бөліп алу немесе жаңа мәндер құру процесі.
 - Неліктен Feature Engineering маңызды?
 - Деректер өзімен өзі жақсы нәтиже бермейді, маңызды белгілер қажет;
 - Жақсы белгі – нашар модельдің нәтижесін де жақсартып алады;
 - Ол overfitting-ті азайту, ассурасу-ді арттыру, интерпретацияны жеңілдету үшін қажет.
 - Feature Engineering кезеңдері mermaid КопироватьРедактировать graph TD; A[Мәліметті талдау] → B[Маңызды айнымалыларды анықтау]; B → C[Белгілерді түрлендіру]; C → D[Жаңа белгілер құрастыру]; D → E[Белгілерді іріктеу]; E → F[Модельге беру];
1. Feature Engineering әдістері Әдіс Сипаттамасы Мысалы Feature Extraction Қосымша мәндер шығару Мерзімнен: күн, ай, апта Feature Transformation Шкалалау, лог түрлендіру $\log(\text{price})$, $\sqrt{\text{income}}$ Binning Үздіксіз мәнді санаттарға бөлу Жас → бала, жасөспірім, ересек Interaction Features Айнымалылардың өзара көбейтіндісі немесе бөлігі $\text{income_per_member} = \text{income} / \text{family_size}$ Polynomial Features Қуатты мән қосу x , x^2 , x^3 Encoding Санаттық мәндерді түрлендіру One-Hot Encoding, Label Encoding Domain Knowledge Салалық логика негізінде қосымша белгілер Несие тарихына негізделген «тәуекел класы»
 2. Мысал: уақыт белгілері python КопироватьРедактировать `df["date"] = pd.to_datetime(df["date"]) df["year"] = df["date"].dt.year df["month"] = df["date"].dt.month df["dayofweek"] = df["date"].dt.dayofweek`
 3. Жаңа белгі жасау мысалдары Қос айнымалыдан қатынас: python КопироватьРедактировать `df["price_per_sqft"] = df["price"] / df["area"]` Binning: python КопироватьРедактировать `df["age_group"] = pd.cut(df["age"], bins=[0, 18, 35, 60, 100], labels=["бала", "жас", "орта", "қарт"])`
 4. Автоматтандырылған Feature Engineering құралдары Құрал Мүмкіндігі FeatureTools Авто-feature generation Kats (Meta features) Уақыттық белгілерді автоматты шығару Scikit-learn pipelines Preprocessing ішінде трансформация ChatGPT/ AutoML Белгілер туралы логика жасау, ұсыныс беру

8. Feature Selection (іріктеу)

Кейбір белгілер артық болуы мүмкін. Олар noise жасайды немесе коррелирленген. Іріктеу әдістері:

- SelectKBest
- Recursive Feature Elimination (RFE)
- Feature Importance (Tree-based)
- Lasso Regression (L1 Regularization) python КопироватьРедактировать

```
from sklearn.feature_selection import SelectKBest, f_classif
```

```
selector = SelectKBest(score_func=f_classif, k=5)
X_selected = selector.fit_transform(X, y)
```

1. Feature Importance графигі python КопироватьРедактировать

```
from sklearn.ensemble import RandomForestClassifier
```

```
model = RandomForestClassifier()
model.fit(X, y)

importances = model.feature_importances_
sns.barplot(x=importances, y=X.columns)
plt.title("Feature Importance")
plt.show()
```

1. Жақсы белгілердің қасиеттері
2. Мағыналы және интерпретацияға оңай;
3. Модель өнімділігін арттырады;
4. Мәліметтің бастапқы құрылымынан тәуелсіз;
5. Outlier-ге тұрақты.

1. Өзін-өзі тексеру сұрақтары
2. Feature Engineering не үшін қажет?
3. Қандай әдістермен жаңа белгілер жасалады?
4. Санаттық белгілерді қалай өңдейміз?
5. Feature Importance дегеніміз не?
6. Қай кезде белгілерді іріктеу қажет?

ҚОРЫТЫНДЫ

Feature Engineering – деректерді «модель түсінетін» пішінге келтіретін шығармашылық және аналитикалық процесс. Дұрыс құрылған белгілер — мықты модельдің негізі.

Лекция № 8

Тақырыбы: Классификация алгоритмдері Жоспары:

- Классификация
- Классификацияның мақсаты
- Классификация түрлері
- Негізгі классификация алгоритмдері

Лекция мәтіні

1. Классификация дегеніміз не? Классификация (classification) — берілген мәліметтер негізінде нысанды (объектіні) белгілі бір класқа жатқызатын бақылаумен оқыту (supervised learning) түрі. Мысал: Электрондық хатты «спам» немесе «спам емес» деп анықтау.
 2. Классификацияның мақсаты
 3. Нысандарды дұрыс топтарға бөлу;
 4. Жаңа объект үшін дұрыс болжам жасау;
 5. Модельдің нақты, түсінікті және тиімді болуын қамтамасыз ету.
 6. Классификация түрлері Түрі Сипаттамасы Мысалы Бинарлы классификация 2 класс 0 немесе 1, «иә»/«жоқ» Көпклассты классификация 3 немесе одан көп класс «алма», «банан», «апельсин» Көпбелгілік классификация Бір нысан бірнеше класқа жатуы мүмкін Бір фильм бірнеше жанрда болуы
1. Негізгі классификация алгоритмдері Алгоритм Қысқаша сипаттама Қолдану мысалы Logistic Regression Сызықтық шекара негізінде бөледі Хаттың спам екенін анықтау K-Nearest Neighbors (KNN) Ұқсас көршілерге қарап шешім шығарады Картинаны тану Decision Tree Шешім ағашы арқылы бөледі Несие беру шешімі Random Forest Бірнеше ағаштан тұратын ансамбль әдісі Жүрек ауруын болжау Naive Bayes Байес теоремасына негізделген Тілдік жіктеу Support Vector Machine (SVM) Гипержазықтық арқылы бөлу Сурет классификациясы Gradient Boosting (XGBoost, LightGBM) Күшейтілген ансамбль моделдері Kaggle жарыстарында үздік нәтиже береді
 1. Мысал: Logistic Regression python КопироватьРедактировать from sklearn.linear_model import LogisticRegression from sklearn.datasets import load_breast_cancer from sklearn.model_selection import train_test_split

```
X, y = load_breast_cancer(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

1. Модель бағалау метрикалары Метрика Мақсаты Accuracy Жалпы дұрыс болжамдар үлесі Precision Нақты оң нәтижелердің үлесі Recall Барлық оң нысандарды табу деңгейі F1-score Precision мен Recall теңгерімі Confusion Matrix Нақты және болжаған мәндерді салыстыру python КопироватьРедактировать from sklearn.metrics import classification_report, confusion_matrix print(classification_report(y_test, y_pred))
1. Confusion Matrix мысалы

Шын оң	Шын теріс	Болжам: оң	TP	FP	Болжам: теріс	FN	TN	TP:	True Positive
FP:				False					Positive
FN:				False					Negative
TN:	True Negative								

1. Практикалық қолдану: KNN python КопироватьРедактировать from sklearn.neighbors import KNeighborsClassifier

```
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

1. Алгоритм таңдау факторлары Критерий Ұсынылатын алгоритм Жылдамдық Naive Bayes, Logistic Түсіндіру ыңғайлылығы Decision Tree Дәлдік Random Forest, XGBoost Шағын дерек KNN Үлкен дерек SVM, LightGBM
1. Өзін-өзі тексеру сұрақтары
2. Классификация алгоритмдерінің мақсаты қандай?
3. Logistic Regression мен Decision Tree айырмашылығы неде?
4. Классификация метрикалары қандай?

5. KNN алгоритмі қалай жұмыс істейді?
6. Random Forest қандай жағдайларда тиімді?

ҚОРЫТЫНДЫ

Классификация — машиналық оқытудың кең таралған, практикалық маңызы зор бағыты. Оның көмегімен көптеген нақты өмірлік мәселелер (диагностика, қаржы, қауіпсіздік, ұсыныс жүйелері) шешіледі.

Лекция № 9

Тақырыбы: Регрессия алгоритмдері Жоспары:

- Регрессия
- Регрессияның мақсаты
- Регрессия түрлері

Лекция мәтіні

1. Регрессия дегеніміз не? Регрессия (Regression) — үздіксіз (сандық) мәндерді болжауға арналған бақылаумен оқыту (supervised learning) түрі. Мысал: Пәтердің ауданына қарай бағасын болжау.
2. Регрессияның мақсаты
3. Айнымалылар арасындағы сандық тәуелділікті анықтау;
4. Келешектегі нақты мәнді болжап шығару;
5. Болжам нәтижелерін түсіндіру және интерпретациялау.
6. Регрессия түрлері Түрі Қысқаша сипаттама Мысалы Жай сызықтық регрессия Бір тәуелсіз айнымалы → бір тәуелді Білім деңгейі → жалақы Көптік регрессия (Multiple) Бірнеше тәуелсіз айнымалы Аудан + бөлме саны → пәтер бағасы Полиномиалдық регрессия Тәуелсіз айнымалының дәрежесі қолданылады x^2 , x^3 түрінде қисық Ridge/Lasso регрессиясы Регуляризация қосылады (overfitting-ті азайту)

Decision Tree/Random Forest регрессиясы Сызықсыз регрессия Күрделі мәліметтерге

1. Негізгі алгоритмдер Алгоритм Ерекшелігі LinearRegression Сызықтық байланыс PolynomialFeatures + LinearRegression Қисық модельдеу Ridge L2 regularization, overfitting-ке қарсы Lasso L1 regularization, feature selection DecisionTreeRegressor Ережеге негізделген бөлу RandomForestRegressor Бірнеше ағаш, жоғары дәлдік GradientBoostingRegressor Boosted ensemble, мықты нәтиже
2. Жай сызықтық регрессия мысалы python КопироватьРедактировать

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pandas as pd
```

Мәлімет дайындау

```
df = pd.read_csv("housing.csv")
X = df[["area"]]
y = df["price"]

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

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

1. Бағалау метрикалары Метрика Мақсаты MAE (Mean Absolute Error) Орташа абсолюттік қате MSE (Mean Squared Error) Орташа квадраттық қате RMSE (Root Mean Squared Error) MSE түбірі R^2 (R-squared) Модельдің түсіндіру қабілеті (0–1 арасында) python КопироватьРедактировать

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
print(mean_absolute_error(y_test, y_pred))
print(r2_score(y_test, y_pred))
```

1. Полиномиалдық регрессия мысалы python КопироватьРедактировать `from sklearn.preprocessing import PolynomialFeatures`
`poly = PolynomialFeatures(degree=2)`

```
X_poly = poly.fit_transform(X)
model = LinearRegression()
model.fit(X_poly, y)

plt.scatter(X, y)
plt.plot(X, model.predict(X_poly), color='red')
8. Регуляризация (Ridge, Lasso)
```

Мақсаты – overfitting-ті болдырмай және feature selection жасау. python КопироватьРедактировать

```
from sklearn.linear_model import Ridge, Lasso

ridge = Ridge(alpha=1.0)
lasso = Lasso(alpha=0.1)
```

1. Decision Tree және Random Forest регрессиясы python КопироватьРедактировать `from sklearn.ensemble import RandomForestRegressor`

```
model = RandomForestRegressor(n_estimators=100)
model.fit(X_train, y_train)
```

1. Алгоритмді таңдау Жағдай Ұсынылатын әдіс Қарапайым, түсінікті LinearRegression Күрделі үлгі, қисық сызық Polynomial, RandomForest Feature көп, overfitting бар Lasso, Ridge Түсіндіру қажет емес, дәлдік маңызды GradientBoosting
1. Өзін-өзі тексеру сұрақтары
2. Сызықтық регрессия мен классификация айырмашылығы қандай?
3. MSE мен MAE айырмашылығы неде?
4. Overfitting-ті қалай азайтуға болады?
5. R^2 метрикасы не үшін қолданылады?
6. Полиномиалдық регрессия қай кезде қолданылады?

ҚОРЫТЫНДЫ

Регрессия — нақты сандық мәндерді болжаудың негізгі әдісі. Ол экономикалық модельдерде, қаржылық болжамдарда, денсаулық сақтау мен ғылымда кең қолданылады.

Лекция № 10

Тақырыбы: Машиналық оқытуда гиперпараметрлер Жоспары:

- Гиперпараметрлер
- Параметр мен гиперпараметр айырмашылығы
- Гиперпараметрлер мысалдары

Лекция мәтіні

1. Гиперпараметрлер дегеніміз не? Гиперпараметрлер (Hyperparameters) — бұл модельді үйретпес бұрын қолмен таңдалатын параметрлер, яғни олар модельдің құрылымын және оқу процесінің мінез-құлқын анықтайды. Анықтама: Гиперпараметр — бұл модельге берілетін сыртқы параметр, оны оқу процесі кезінде модель өзі үйренбейді.
 2. Параметр мен гиперпараметр айырмашылығы Сипаттама Параметр Гиперпараметр Мәні Үйреніледі (модельмен) Алдын ала беріледі Мысалы Коэффициенттер (θ) `learning_rate`, `max_depth`, `k_neighbors` Қайда қолданылады Модель ішінде Модель құрылымында, оқу процессінде
 3. Гиперпараметрлер мысалдары KNN:
 4. `n_neighbors`: көршілер саны
 5. `weights`: салмақ беру тәсілі Decision Tree:
 6. `max_depth`: ағаш тереңдігі
 7. `min_samples_split`: бөлу үшін минималды үлгілер саны Random Forest:
 8. `n_estimators`: ағаш саны
 9. `max_features`: әр ағашта қаралатын белгілер саны Gradient Boosting:
 10. `learning_rate`: қадам ұзындығы
 11. `n_estimators`: итерация саны SVM:
 12. `C`: қатені жазалау коэффициенті
 13. `kernel`: ядро түрі (`linear`, `rbf`, `poly`)
1. Неліктен гиперпараметрді баптау маңызды?
 2. Дұрыс таңдалмаған гиперпараметрлер `overfitting` немесе `underfitting` әкеледі
 3. Модельдің дәлдігіне және жылдамдығына тікелей әсер етеді
 4. Жақсы баптау арқылы өнімділік пен тұрақтылықты арттыруға болады

5. Гиперпараметрлерді баптау әдістері (Hyperparameter Tuning)

Әдіс Қысқаша сипаттама Grid Search Барлық мүмкін комбинацияны сынайды Random Search Параметр кеңістігінен кездейсоқ сынамалар Bayesian Optimization Математикалық модель арқылы ең тиімді мәндерді іздейді Genetic Algorithms Эволюциялық тәсіл арқылы іздеу Optuna / Hyperopt / Ray Tune Заманауи автоматтандырылған құралдар

6. Grid Search мысалы (Scikit-learn)

python Копировать Редактировать

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier

param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [5, 10, 15]
```

```
}
```

```
model = RandomForestClassifier()
grid_search = GridSearchCV(model, param_grid, cv=5)
grid_search.fit(X_train, y_train)

print(grid_search.best_params_)
```

1. Cross-validation және гиперпараметрлер Гиперпараметрлерді бағалауда k-fold cross-validation әдісі жиі қолданылады:

2. Мәлімет бірнеше бөлікке бөлінеді (әдетте $k=5$ немесе $k=10$);
 3. Әр итерацияда бір бөлік — тексеруге, қалғандары — үйретуге беріледі;
 4. Орташа нәтиже есептеліп, ең жақсы параметрлер таңдалады.
 5. Кеңестер Тек 1 параметрді өзгертпей, бірнешеуін бір уақытта қарау
Модельдің оқу уақытын бақылау – кей гиперпараметрлер есептеуді баяулатуы мүмкін
Алдымен Random Search, кейін Grid Search қолдану – тиімді
 6. Модель сапасына әсер ететін гиперпараметрлер мысалы Алгоритм Гиперпараметр Әсері Decision Tree max_depth
Ағаш күрделілігін анықтайды Random Forest n_estimators Орташа нәтиже, тұрақтылық SVM C Класстарды бөлу дәлдігі
Gradient Boosting learning_rate Оқытудың жылдамдығы, нәтижеге жету уақыты
1. Өзін-өзі тексеру сұрақтары
 2. Параметр мен гиперпараметр айырмашылығы қандай?
 3. Grid Search қалай жұмыс істейді?
 4. learning_rate қандай әсер береді?
 5. Random Forest-те қандай гиперпараметрлер маңызды?
 6. Cross-validation не үшін қолданылады?

ҚОРЫТЫНДЫ

Гиперпараметрлерді дұрыс баптау — машиналық оқытудағы жоғары сапалы модельдің кілті. Қарапайым алгоритмдерді де, гиперпараметр арқылы оңтайлап, қуатты құралға айналдыруға болады.

Лекция № 11

Тақырыбы: Scikit-learn кітапханасы Жоспары:

- Scikit-learn
- Scikit-learn мүмкіндіктері
- Орнату және импорттау

Лекция мәтіні

1. Scikit-learn дегеніміз не? Scikit-learn (sklearn) — Python тіліндегі ашық бастапқы кодты, машиналық оқытуға арналған ең танымал кітапхана. Ол деректерді дайындау, модель құру, баптау, бағалау, болжау секілді бүкіл процесті қамтиды. Анықтама: Scikit-learn — бұл бір интерфейс арқылы көптеген машиналық оқыту алгоритмдерін жүзеге асыруға мүмкіндік беретін платформа.
1. Scikit-learn мүмкіндіктері Мүмкіндік Мысал Деректерді дайындау StandardScaler, Imputer, OneHotEncoder Бақылаумен оқыту Классификация, регрессия Бақылаусыз оқыту Кластерлеу (KMeans), PCA Модель бағалау Cross-validation, метрикалар Гиперпараметрді баптау GridSearchCV, RandomizedSearchCV Pipeline құру Кодты автоматтандыру, қайта қолдану
2. Орнату және импорттау bash КопироватьРедактировать pip install scikit-learn python КопироватьРедактировать import sklearn from sklearn.model_selection import train_test_split from sklearn.linear_model import LinearRegression
3. Жалпы жұмыс ағымы mermaid КопироватьРедактировать graph TD; A[Деректерді жүктеу] → B[Train/Test бөлу]; B → C[Алдын ала өңдеу]; C → D[Модель таңдау]; D → E[Оқыту]; E → F[Бағалау]; F → G[Болжау];
4. Мысал: Регрессия python КопироватьРедактировать from sklearn.linear_model import LinearRegression from sklearn.model_selection import train_test_split

```
X = df.drop("price", axis=1)
y = df["price"]

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

model = LinearRegression()
```

```
model.fit(X_train, y_train)
predictions = model.predict(X_test)
```

1. Бағалау метрикалары python КопироватьРедактировать from sklearn.metrics import mean_absolute_error, r2_score

```
print(mean_absolute_error(y_test, predictions))
print(r2_score(y_test, predictions))
```

1. Классификация мысалы python КопироватьРедактировать from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy_score, classification_report

```
model = RandomForestClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

1. Алдын ала өңдеу құралдары Құрал Қызметі StandardScaler Нормализация (z-score) MinMaxScaler 0–1 аралығына келтіру OneHotEncoder Категорияларды бинарлық түрге ауыстыру SimpleImputer Бос мәндерді толтыру python КопироватьРедактировать from sklearn.preprocessing import StandardScaler scaler = StandardScaler() X_scaled = scaler.fit_transform(X)
2. Pipeline – өңдеу мен модельді біріктіру python КопироватьРедактировать from sklearn.pipeline import Pipeline

```
pipe = Pipeline([
    ("scaler", StandardScaler()),
    ("model", LinearRegression())
])

pipe.fit(X_train, y_train)
predictions = pipe.predict(X_test)
```

1. Гиперпараметрді баптау python КопироватьРедактировать from sklearn.model_selection import GridSearchCV

```
params = {"n_estimators": [50, 100], "max_depth": [5, 10]}
grid = GridSearchCV(RandomForestClassifier(), params, cv=5)
grid.fit(X_train, y_train)
print(grid.best_params_)
```

1. Танымал модельдер Модель Мақсаты LinearRegression Сандық болжам LogisticRegression Бинарлы классификация KNeighborsClassifier Көршілер бойынша топтау DecisionTreeClassifier Ережеге негізделген ағаш RandomForestClassifier Бірнеше ағаш негізінде ансамбль SVM (SVC, SVR) Ядро әдістерімен бөліп жіктеу
1. Өзін-өзі тексеру сұрақтары
2. Scikit-learn кітапханасы не үшін қолданылады?
3. train_test_split функциясы қандай қызмет атқарады?
4. Pipeline қолданудың артықшылығы қандай?
5. StandardScaler мен MinMaxScaler айырмашылығы қандай?
6. GridSearchCV қалай жұмыс істейді? Лекция № 12

Тақырыбы: Anaconda платформасы Жоспары:

- Anaconda дегеніміз не?
- Anaconda-ның артықшылықтары

- Anaconda Navigator интерфейсі Лекция мәтіні
- Anaconda дегеніміз не? Anaconda — деректер ғылымы (Data Science), машиналық оқыту және жасанды интеллект жобаларын дамыту үшін кеңінен қолданылатын Python және R тілдеріне арналған интеграцияланған платформа. Анықтама: Anaconda – бұл Python ортасы, кітапханалар, құралдар мен пакеттерді оңай басқаруға және іске қосуға арналған кросс-платформалық дистрибутив.
- Anaconda-ның артықшылықтары Артықшылық Түсіндірме Барлығы бір жерде Python, Jupyter, кітапханалар, Conda бірге орнатылады Оңай орнату және жаңарту Пакеттерді Conda көмегімен басқару Виртуалды орта Өр жоба үшін жеке кітапхана конфигурациясы JupyterLab және Spyder Интерактивті IDE-лер алдын ала дайын

1. Anaconda орнату
2. <https://www.anaconda.com> ресми сайтынан өз жүйеңізге сәйкес орнатушыны жүктеу
3. Орнату барысында "Add Anaconda to PATH" опциясын таңдау ұсынылады
4. Орнату аяқталғаннан кейін Anaconda Navigator және Anaconda Prompt қолжетімді болады
5. Anaconda Navigator интерфейсі Anaconda Navigator – графикалық интерфейс арқылы:
6. Жаңа жобалар ашу
7. Виртуалды орта жасау
8. JupyterLab, Spyder, VS Code іске қосу
9. Пакеттерді орнату/жаңарту мүмкін
10. Conda – пакет және орта менеджері Пакет орнату: `bash КопироватьРедактировать conda install numpy pandas scikit-learn` Виртуалды орта жасау: `bash КопироватьРедактировать conda create --name ml_env python=3.10` Ортаны іске қосу: `bash КопироватьРедактировать conda activate ml_env` Ортаны өшіру: `bash КопироватьРедактировать conda deactivate` Орта тізімін көру: `bash КопироватьРедактировать conda info --envs`
11. JupyterLab арқылы код жазу Anaconda орнатылғаннан кейін: `bash КопироватьРедактировать jupyter lab` → Браузерде ашылады → Файлдарды .ipynb форматында сақтайды → Визуализация, машиналық оқыту, модель нәтижесін көрсетуге ыңғайлы
12. Практикалық мысал
13. Жаңа орта құру: `bash КопироватьРедактировать conda create -n ai_env python=3.9` `conda activate ai_env`
14. Қажетті кітапханаларды орнату: `bash КопироватьРедактировать conda install numpy pandas matplotlib seaborn scikit-learn`
15. Jupyter Lab ашу: `bash КопироватьРедактировать jupyter lab`
16. Conda vs pip Салыстыру Conda pip Ортаны басқарады ма?

(venv бөлек)

Python-нан бөлек кітапхана орнатады ма?

(C/C++ libs т.б.)

× Кітапхана көздері Anaconda repo PyPI Орнату жылдамдығы Тұрақты Тез, бірақ кейде үйлесімсіздік болады

1. Кеңестер
2. Өрбір жоба үшін жеке орта жасаңыз
3. Орта талаптарын файлда сақтаңыз: `bash КопироватьРедактировать conda env export > environment.yml`
4. Басқа құрылғыға орнату үшін: `bash КопироватьРедактировать conda env create -f environment.yml`
5. Өзін-өзі тексеру сұрақтары
6. Anaconda деген не және оның артықшылығы қандай?
7. Conda көмегімен орта қалай жасалады?
8. JupyterLab қандай жағдайда қолданылады?
9. Pip пен Conda айырмашылығы қандай?

10. Environment.yml не үшін қажет?

ҚОРЫТЫНДЫ

Anaconda — Python негізіндегі жобалар үшін сенімді, тұрақты, оңай басқарылатын экожүйе. Ол әсіресе деректер талдауы мен машиналық оқыту саласында бастапқыдан кәсіби деңгейге дейінгі жұмыстарды ұйымдастыруға мүмкіндік береді.

Лекция № 13

Тақырыбы: Модельді бағалау және интерпретация Жоспары:

- Модельді бағалау дегеніміз не?
- Бақылау түріне қарай бағалау метрикалары
- Классификация метрикалары

Лекция мәтіні

1. Модельді бағалау дегеніміз не? Модельді бағалау (model evaluation) — бұл машиналық оқыту алгоритмінің сапасын, дәлдігін, тұрақтылығын анықтайтын метрикалар мен әдістер жиынтығы. Мақсаты — модельдің нақты өмірдегі деректерге қаншалықты жақсы бейімделетінін тексеру.
2. Бағалау мақсаты
3. Жеткіліксіз немесе артық үйренуді (under/overfitting) анықтау
4. Модель сапасын салыстыру
5. Болжаудың дұрыстығын тексеру
6. Іс жүзіндегі қолдануға жарамдылығын бағалау
7. Бақылау түріне қарай бағалау метрикалары
Модель түрі Метрикалар
Классификация Accuracy, Precision, Recall, F1-score, ROC-AUC
Регрессия MAE, MSE, RMSE, R² score

1. Классификация метрикалары Confusion Matrix (Шатастыру матрицасы):

Нақты оң Нақты теріс Болжам оң TP FP Болжам теріс FN TN Метрикалар: Метрика Формула Түсіндірме Accuracy

$$(TP + TN) / \text{Барлығы}$$

Дұрыс болжамдар үлесі Precision

$$TP / (TP + FP)$$

Болжаған «оңдардың» нақтылығы Recall

$$TP / (TP + FN)$$

Нақты оң нысандарды табу үлесі F1-score

$$2 * (P * R) / (P + R)$$

Precision мен Recall теңгерімі ROC-AUC – Жалған позитивтерге төзімділік

5. Классификация мысалы (Python)

python Копировать Редактировать

```
from sklearn.metrics import confusion_matrix, classification_report

y_true = [0, 1, 1, 0, 1]
y_pred = [0, 1, 0, 0, 1]

print(confusion_matrix(y_true, y_pred))
print(classification_report(y_true, y_pred))
```

1. Регрессия метрикалары Метрика Формула Мақсаты MAE (Mean Absolute Error) $\sum y - \hat{y}$ MSE (Mean Squared Error) $\sum (y - \hat{y})^2 / n$ Квадраттық қате RMSE $\sqrt{\text{MSE}}$ Қатені түбір астында көрсету R^2 (R-squared) $1 - (SS_{\text{res}} / SS_{\text{tot}})$ Модельдің түсіндіру қабілеті (0–1)

7. Регрессия мысалы (Python)

python Копировать Редактировать

```
from sklearn.metrics import mean_absolute_error, r2_score

mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"MAE: {mae}, R2: {r2}")
```

1. Модель интерпретациясы дегеніміз не? Интерпретация (Interpretation) — бұл модельдің неліктен нақты бір шешім қабылдағанын түсіндіру процесі. Бұл әсіресе медицина, қаржы, заң салаларында қажет (модельдің әділеттілігі, сенімділігі үшін).
2. Интерпретация әдістері Әдіс Сипаттамасы Feature Importance Қай айнымалы нәтижеге қаншалықты әсер етті SHAP (SHapley values) Әрбір белгінің үлесін математикалық негізде көрсету LIME (Local Interpretable Model Explanation) Бір болжамды локалды түсіндіру Partial Dependence Plot (PDP) Белгі мен нәтиже арасындағы тәуелділік графигі

10. Feature Importance мысалы (Random Forest)

python Копировать Редактировать

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

importances = model.feature_importances_
features = pd.Series(importances, index=X.columns)
features.sort_values().plot(kind='barh')
plt.title("Feature Importance")
plt.show()

11. SHAP визуализациясы (қосымша)
```

python Копировать Редактировать

```
import shap

explainer = shap.Explainer(model.predict, X)
shap_values = explainer(X)

shap.plots.beeswarm(shap_values)
```

1. Бағалау мен интерпретацияны біріктіру Қадам Мақсат

2. Бағалау Модель қаншалықты дәл?
3. Интерпретация Неге дәл? Қай айнымалылар маңызды?
4. Талдау Модельді сенімді, әділ, жауапты ету
1. Өзін-өзі тексеру сұрақтары
2. Accuracy мен Precision айырмашылығы қандай?
3. R^2 метрикасы не үшін қолданылады?
4. SHAP әдісі не үшін қажет?
5. Confusion Matrix-тен қандай метрикалар алынады?
6. Қай кезде интерпретация қажет?

ҚОРЫТЫНДЫ

Бағалау мен интерпретация — машиналық оқытудағы сенімділік, сапа және этика мәселелерін шешетін маңызды кезең. Тек дәл модель емес, түсінікті модель де қажет.

Лекция № 14

Тақырыбы: Машиналық оқыту жобасының кезеңдері Жоспары:

- Машиналық оқыту жобасы дегеніміз не?
- Жалпы кезеңдері
- Кезеңдер сипаттамасы

Лекция мәтіні

1. Машиналық оқыту жобасы дегеніміз не? Машиналық оқыту жобасы – нақты мәселені шешу үшін деректер жинау, талдау, модель құру және енгізу кезеңдерін қамтитын кешенді процесс. Ол бизнес мақсатты немесе ғылыми міндетті шешуге бағытталады.
2. Жалпы кезеңдері mermaid Копировать Редактировать graph TD; A[1. Мәселені анықтау] → B[2. Деректерді жинау]; B --> C[3. Деректерді өңдеу]; C → D[4. Модель құру]; D → E[5. Модельді бағалау]; E → F[6. Модельді оңтайлау]; F → G[7. Енгізу (deployment)]; G → H[8. Мониторинг және қолдау];
3. Кезеңдер сипаттамасы
4. Мәселені анықтау
5. Нақты мақсатты тұжырымдау
6. Модель қандай сұраққа жауап беруі керек? Мысал: «Клиент несиені қайтара ма?» — бинарлы классификация
7. Деректерді жинау
8. Ішкі немесе сыртқы дереккөздер
9. API, сенсор, формалар, дерекқор Мысал: CRM жүйесінен клиенттік ақпарат Деректерді алдын ала өңдеу (Data preprocessing)
10. Бос мәндерді өңдеу
11. Шкалалау, кодтау (OneHot)
12. Outlier тазалау Бұл кезеңнің сапасы — модель нәтижесіне әсер етеді Модель құру (Modeling)
13. Алгоритм таңдау (логистикалық регрессия, шешім ағашы, Random Forest, т.б.)
14. Модельді үйрету Jupyter немесе Google Colab-та код арқылы іске асырылады Модельді бағалау (Evaluation)
15. Accuracy, Recall, F1-score, RMSE, R^2 т.б.
16. Confusion matrix, ROC curve, кросс-валидация Мақсат – нақты нәтиже мен болжамды салыстыру Модельді оңтайлау (Tuning)
17. Гиперпараметрді баптау (GridSearchCV, RandomizedSearchCV)
18. Overfitting/Underfitting балансын табу Ең жақсы нәтижеге жету үшін қайталама оқыту Модельді енгізу (Deployment)
19. Модельді өнімге енгізу (веб-сервер, API, мобильдік қолданба)

20. Flask, FastAPI, Docker, CI/CD құралдары Пайдаланушы жүйесімен әрекеттесу Мониторинг және қолдау
21. Уақыт өте модельдің сапасы төмендеуі мүмкін (Data Drift)
22. Жаңарту, қайта үйрету
23. Автоматты мониторинг жүйесін қосу (MLflow, Evidently) CRISP-DM стандарты CRISP-DM – машиналық оқыту жобасының құрылымдық моделі (жобалау стандарты): Кезең Мазмұны
24. Бизнес мақсатын анықтау Мәселенің мәнін түсіну
25. Деректерді түсіну Алғашқы зерттеу
26. Деректерді дайындау Тазалау, трансформация
27. Модельдеу Алгоритм таңдау және оқыту
28. Бағалау Салыстыру, дәлдік тексеру
29. Қолдану Модельді жүйеге енгізу

Практикалық жоба мысалы: Студент үлгерімін болжау Кезең Мысал Мақсат Студенттің сессиядан өте ме, өтпей ме – соны болжау Деректер Ж attendance, GPA, пән бойынша балл Модель Logistic Regression Бағалау Accuracy, F1-score Нәтиже Ұсыным: «тәуекелі жоғары» студенттерге кеңес беру

Құралдар Кезең Құралдар Жинау SQL, CSV, API Өңдеу Pandas, NumPy Модель Scikit-learn, XGBoost Бағалау Matplotlib, Seaborn, SHAP Енгізу Flask, FastAPI, Streamlit Мониторинг MLflow, Neptune, DVC

Өзін-өзі тексеру сұрақтары

- Машиналық оқыту жобасының негізгі кезеңдерін атаңыз.
- CRISP-DM не үшін қолданылады?
- Модельді енгізу кезінде қандай қиындықтар туындайды?
- Data Drift деген не?
- Неліктен модельді қайта оқыту қажет?

ҚОРЫТЫНДЫ

Машиналық оқыту жобасы – бұл тек модель құру емес, толық циклді процесті басқару. Сәтті жоба үшін техникалық дағдымен қатар, логика, құрылым және жауапкершілік қажет.

Лекция № 15

Тақырыбы: Жобалар портфолиосы және презентация Жоспары:

- Жоба портфолиосы дегеніміз не?
- Портфолио не үшін қажет?
- Жобаның негізгі құрылымы Лекция мәтіні Жоба портфолиосы дегеніміз не? Жоба портфолиосы (Project Portfolio) — бұл студенттің немесе маманның машиналық оқыту бойынша орындаған жобаларының жинағы, ол оның білімін, тәжірибесін және дағдысын көрсетеді. Портфолио – сіздің практикалық қабілетіңізді дәлелдейтін құжат, жұмыс берушіге немесе ғылыми жетекшіге көрінетін нәтиже. Портфолио не үшін қажет? Мақсат Сипаттамасы Жұмысқа орналасу HR мен техникалық мамандарға дағдыларды көрсету Академиялық бағалау Сабақ соңындағы жобаны қорғау Ғылыми жобалар Грантқа немесе зерттеу тобына кіру үшін Технологиялық стартап Жоба идеясын инвестор алдында қорғау Жобаның негізгі құрылымы Жақсы жоба портфолиосы келесі бөлімдерден тұрады: Бөлім Мазмұны
- Мәселені сипаттау Не үшін және қандай мәселені шешеді?
- Деректер сипаттамасы Қайдан алынған? Қандай айнымалылар бар?
- Алдын ала өңдеу Тазалау, шкалалау, кодтау
- Модельдер Қандай алгоритм қолданылды, неге?
- Бағалау нәтижелері Accuracy, F1-score, RMSE, т.б.
- Интерпретация Қай айнымалы маңызды?
- Қорытынды/Ұсыныс Нәтиже мен оның қолданылуы
- Визуализация Графиктер, диаграммалар

- Сілтемелер GitHub, Colab, Dataset, презентация слайдтары

Портфолио форматы Формат Артықшылықтары PDF есеп Құжат ретінде жинақталған, басып шығаруға болады GitHub репозиторий Код, README, дерек – барлығы бір жерде Google Colab / Jupyter Notebook Интерактивті код+графика Streamlit / Gradio web app Веб-интерфейс арқылы демонстрация PowerPoint / Google Slides Қорғауға арналған қысқа таныстырылым

Презентация жасаудың құрылымы Бөлім Сипаттамасы

1. Кіріспе Жобаның тақырыбы мен мақсаты
2. Мәселе Не үшін бұл проблема маңызды?
3. Деректер Қайдан алынды, қалай өңделді
4. Модель Таңдалған алгоритм мен неге сол
5. Нәтижелер Метрикалар, кестелер, графиктер
6. Қорытынды Ұсыныстар, болашағы, шектеулер Ұсынылатын уақыт – 5–7 минут Фокус – мәселе мен шешім, қысқа әрі нақты Үлгі жоба тақырыптары (студенттер үшін) Тақырып Түрі Студент үлгерімін болжау Классификация Пәтер бағасын болжау Регрессия Фильм жанрын болжау Көпклассты классификация Клиенттің кету ықтималдығы (churn prediction) Бизнес-аналитика Твиттердегі пікірді анықтау NLP (Natural Language Processing) Спорт нәтижесін болжау Уақыттық деректер (time series)

Презентация құралдары Құрал Мүмкіндігі PowerPoint / Google Slides Дәстүрлі слайдтар Canva / Beautiful.ai Әдемі дизайнмен презентация Tome.app ЖИ арқылы презентация генерациясы Streamlit / Gradio Веб-интерфейс түрінде жобаны көрсету

Өзін-өзі тексеру сұрақтары

- Жобалық портфолио қандай бөлімдерден тұруы керек?
- Презентацияда қандай ақпарат ең маңызды?
- Жобаны кімге және қалай көрсету қажет?
- Қай формат кәсіби портфолио үшін ыңғайлы?
- Модель нәтижесін қалай интерпретациялауға болады?

ҚОРЫТЫНДЫ

Жақсы портфолио — сіздің дағдыңыздың айнасы. Тіпті шағын жоба болса да, оны түсінікті, құрылымды, кәсіби түрде көрсету — болашақ мансабыңыз үшін үлкен қадам.

Ұсынылатын әдебиеттер тізімі Негізгі әдебиеттер

1. Баймұхамедов, М.Ф. Жасанды интеллект: қазіргі заманғы теория және тәжірибе = Artificial Intelligence Modern Theory and Practice. 1-бөлім: Оқу құралы / М.Ф. Баймұхамедов, А.М. Баймұхамедов, С.Н. Боранбаев.- Алматы: Бастау, 2020.- 248с.- (студентам и магистрантам). МОН РК
2. Баймұхамедов, М.Ф. Жасанды интеллект: қазіргі заманғы теория және тәжірибе= Artificial Intelligence Modern Theory and Practice. 2-бөлім: Оқу құралы / М.Ф. Баймұхамедов, А.М. Баймұхамедов, С.Н. Боранбаев.- Алматы: Бастау, 2020.- 240с.- (Тұрмыстық робототехника). МОН РК студентам и магистрантам
3. Нурпеисова, Т.Б. Прикладная робототехника: Учебное пособие / Т.Б. Нурпеисова, Д.В. Панюкова, К.В. Панюков.- Алматы: Бастау, 2021.- 227б.- (студентам).
4. Зарубин, М.Ю.Зияткерлік өмірді қамтамасыз ету жүйелер = Интеллектуальные системы жизнеобеспечения. 3-том: Оқулық / М.Ю. Зарубин, О.В. Баяк, А.Т. Ескожина.- Алматы: Бастау, 2019.- 480б.- (студентам и магистрантам). МОН РК-5
5. Дэнни, С. Устройство и программирование автономных роботов: Учебное пособие / Стейпл Дэнни; пер. с англ. Е.В. Шевчук; науч.ред. В.С. Яценков.- Москва: ДМК Пресс, 2022.- 520с Қосымша әдебиеттер
6. Жомартова, Ш.Ө., Тергеусізова, Ә.С. Желілік технологиялар : оқу құралы. . - Алматы: Қазақ университеті, 2020. <http://rmebrk.kz/book/1178944>

7. Баймұхамедов, М.Ф. Жасанды интеллект: қазіргі заманғы теория және тәжірибе = Artificial Intelligence Modern Theory and Practice. 1-бөлім: Электронный ресурс / М.Ф. Баймұхамедов, А.М. Баймұхамедов, С.Н. Боранбаев.- Алматы: Бастау, 2020.- 9, 72 Мб.- (студентам и магистрантам). МОН РК
8. Баймұхамедов, М.Ф. Жасанды интеллект: қазіргі заманғы теория және тәжірибе = Artificial Intelligence Modern Theory and Practice. 2-бөлім: Электронный ресурс / М.Ф. Баймұхамедов, А.М. Баймұхамедов, С.Н. Боранбаев.- Алматы: Бастау, 2020.- 13, 0 Мб: Тұрмыстық робототехника.- (студентам и магистрантам). МОН РК
9. Нурпеисова, Т.Б. Прикладная робототехника: Электронный ресурс / Т.Б. Нурпеисова, Д.В. Панюкова, К.В. Панюков.- Алматы: Бастау, 2021.- 16, 9 Мб.- (студентам). МОН РК
10. Зарубин, М.Ю. Зияткерлік өмірді қамтамасыз ету жүйелер = Интеллектуальные системы жизнеобеспечения. 3-том: Электронный ресурс / М.Ю. Зарубин, О.В. Баяк, А.Т. Ескожина.- Алматы: Бастау, 2019.- 34, 2 Мб.- (студентам и магистрантам). МОН РК
11. Дейтел П., Дейтел, Х. Python: Искусственный интеллект, большие данные и облачные вычисления. - СПб.: Питер, 2020. - 864 с.

Мультимедиялық қамтамасыз ету және электрондық ресурстар

1. Шынжігіт Б.Б. Жасанды интеллект технологиялары және машиналық оқыту пәні бойынша лекция жинағы-2023, 48 б. <https://drive.google.com/file/d/1ExzAIVQhb4InV8uLEL8Fnss3CZMLtCeT/view>
2. Шынжігіт Б.Б. Сборник лекций по дисциплине «Технологии искусственного интеллекта и машинное обучение» 2023г 48 стр. https://drive.google.com/file/d/1h4ojrW5G2RQyWrp2GoDfpDqAdTpF_1BF/view
3. https://drive.google.com/drive/folders/16bIGPOMn_RvKi2mhxAzsVOu7TQ_pok9d%20II%20%D0%A2%D0%9E%D0%9C-2.%20%D0%9E%D0%90%D0%98%D0%A3,%202022%20260%20%D0%B1