

5 – dars (Report)

Machine Learning project qilish uchun asosiy ketma ketliklar

- Muammo aniqlash →
- Ma'lumot yig'ish →
- Ma'lumotni tahlil qilish va tozalash →
- Ma'lumotlarni tayyorlash →
- Model tanlash →
- Modelni o'qitish →
- Modelni baholash →
- Optimallashtirish →
- Modelni joylashtirish →
- Kuzatish va yaxshilash.

1. Muammo aniqlash (Define the Problem)

- ML loyihasining maqsadini aniqlash:
 - **Nima qilishni xohlayabman?**
 - Misol: Sotuvlarni bashorat qilish, tasvirlarni aniqlash, matnni tasniflash va boshqalar.
 - **Natija qanday ko'rinishi kerak?**
 - Regression, klassifikatsiya yoki klasterlash vazifalari.

2. Ma'lumotlarni yig'ish (Data Collection)

- Tahlil qilish va modellashtirish uchun ma'lumotlarni yig'ish:
 - Ma'lumotni yig'ish usullari:
 - API orqali.
 - CSV, Excel yoki ma'lumotlar bazasi (SQL).
 - Ochiq ma'lumotlar platformalari (Kaggle, UCI, va boshqalar).
 - Ma'lumotlar hajmi va sifati haqida o'ylash.

3. Ma'lumotlarni tahlil qilish va tozalash (Exploratory Data Analysis and Cleaning)

- **Ma'lumotlarni tahlil qilish (EDA):**
 - Missing (yo'q) ma'lumotlarni aniqlang.
 - Outlier (me'yordan chetga chiqqan qiymatlar) ni tekshiring.
 - Ma'lumotlarning taqsimotini ko'rib chiqing.
- **Tozalash:**
 - Yo'q ma'lumotlarni to'ldiring yoki olib tashlash.
 - Noto'g'ri qiymatlarni tuzatish.

4. Ma'lumotlarni tayyorlash (Data Preprocessing)

- **Ma'lumotlarni kodlash:**
 - Kategorik ma'lumotlarni **One-Hot Encoding** yoki **Label Encoding** bilan raqamlarga aylantiring.
- **Normalizatsiya yoki Standartizatsiya:**
 - Ma'lumotlarni bir xil miqyosga keltiring.
- **Ma'lumotlarni bo'lish:**
 - Ma'lumotlarni o'qitish va test uchun ajrating.
 - Masalan: 80% - trening, 20% - test uchun.

Kod misoli:

python

Copy code

```
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, random_state=42)
```

5. Model tanlash (Model Selection)

- **Modelni tanlash:**

- Regression uchun: Linear Regression, Decision Tree, Random Forest.
- Klassifikatsiya uchun: Logistic Regression, SVM, Random Forest, Neural Networks.
- Klasterlash uchun: K-Means, DBSCAN.

- **Modelni sinovdan o'tkazish:**

- Har xil modellarni sinab ko'ring va eng yaxshi ishlaydiganini tanlasg.

6. Modelni o'qitish (Model Training)

- Tanlangan modelni ma'lumotlar bo'yicha o'qiting.
- Giperparametrlarni sozlang.

Kod misoli:

```
python
```

```
Copy code
```

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

```
model = RandomForestClassifier(n_estimators=100)
```

```
model.fit(X_train, y_train)
```

7. Modelni baholash (Model Evaluation)

- O'qitilgan modelni test ma'lumotlarida baholang:
 - Regression uchun: MSE, RMSE, R2-score.
 - Klassifikatsiya uchun: Accuracy, Precision, Recall, F1-score, ROC-AUC.

Kod misoli:

```
python
```

```
Copy code
```

```
from sklearn.metrics import accuracy_score
```

```
y_pred = model.predict(X_test)
```

```
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
```

8. Giperparametrlarni sozlash (Hyperparameter Tuning)

- Modelni optimallashtirish uchun giperparametrlarni sozlang:
 - GridSearchCV yoki RandomizedSearchCV yordamida.

9. Modelni joylashtirish (Model Deployment)

- Modelni real dunyo uchun foydalanishga tayyorlang:
 - Flask yoki FastAPI yordamida API yaratish.
 - Modelni Docker yoki cloud platformalariga yuklash (AWS, Azure, Google Cloud).

10. Modelni kuzatish va yaxshilash (Monitoring and Updating)

- Model ishlashini kuzatib boring:
 - Doimiy yangi ma'lumotlarni qo'shib o'rgating.

- Model eskirganligini tekshiring va yangilang.

Kod misoli:

python

Copy code

```
from sklearn.model_selection import GridSearchCV
```

```
param_grid = {'n_estimators': [100, 200], 'max_depth': [10, 20]}
```

```
grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=3)
```

```
grid.fit(X_train, y_train)
```

Data Preprocessing: Missing - values

1. Drop
2. Mean
3. Median
4. Mode
5. Fixed

Usul	Qachon ishlatiladi?	Mos keladigan ma'lumot turi
Drop	Yo'q qiymatlar kam bo'lsa	Har qanday
Mean	Yo'q qiymatlarni o'rtacha qiymat bilan to'ldirish kerak bo'lsa	Raqamli (Numerical)
Median	Me'yordan chetga chiqqan qiymatlar mavjud bo'lsa	Raqamli (Numerical)
Mode	Kategorik ma'lumotlarni to'ldirish kerak bo'lsa	Kategorik (Categorical)
Fixed	Aniq belgilangan qiymat bilan almashtirish kerak bo'lsa	Har qanday

Inner Conditionals (inner if)

- Ichki conditionals
- Uzunroq muammoi tartibli va qisqaroq shaklda bajarishga yordam beradi

Inner conditional — bu ichma-ich joylashgan (nested) if shartlarini ishlatish orqali murakkab muammolarni tartibli va qisqaroq shaklda hal qilish usuli. Bu dasturda biror shart bajarilganda, uning ichida boshqa shartlarni tekshirishga imkon beradi.

2. Talabani bahosini tekshirish

Vazifa: Talabani bahosi (0-100 oralig'ida) aniqlanadi. Bahoga qarab uning darajasini ko'rsatish:

- 90 va undan yuqori: "A'lo"
- 75-89: "Yaxshi"
- 60-74: "Qoniqarli"
- 59 va undan past: "Yomon"