GOVT OF INDIA CERTIFIED SARVA EDUCATION ISO 9001:2015 CERTIFIED Start New Computer Centre Sarvaindia.com Join TODAY

Chapter-1

GOVT OF INDIA CERTIFIED SARVA EDUCATION ISO 9001:2015 CERTIFIED Start New Computer Centre Sarvaindia.com Join TODAY

कृत्रिम बुद्धिमत्ता (Artificial Intelligence - AI) भूमिका

कृत्रिम बुद्धिमत्ता (Artificial Intelligence - AI) आज के समय की सबसे प्रगतिशील और परिवर्तनकारी तकनीकों में से एक है। यह मशीनों को सोचने, समझने, सीखने और निर्णय लेने की क्षमता देती है।

हम सीखेंगे:

- AI क्या है?
- AI का इतिहास और विकास
- AI के प्रकार
- AI के अनुप्रयोग
- मशीन लर्निंग और डीप लर्निंग का परिचय

AI क्या है?

आर्टिफिशियल इंटेलिजेंस (AI) क्या है?

कृत्रिम बुद्धिमत्ता (Artificial Intelligence / AI) कंप्यूटर विज्ञान की एक शाखा है, जिसका उद्देश्य ऐसे मशीनों या प्रणालियों का निर्माण करना है जो **मानव जैसी बुद्धिमत्ता का अनुकरण** कर सकें।

AI वाली मशीनें **सीख सकती हैं, सोच सकती हैं, निर्णय ले सकती हैं**, और समस्याओं को हल कर सकती हैं — ठीक उसी तरह जैसे इंसान करता है।

सरल शब्दों में:
AI = मशीनों को सोचने और समझदारी से काम करने योग्य बनाना।

AI का इतिहास और विकास (History & Evolution)

काल / युग	महत्वपूर्ण घटनाएं और विवरण
1940–1950 का दशक	**AI की मूल कल्पना**: एलन ट्यूरिंग ने ऐसी मशीनों की कल्पना की जो किसी भी मानव कार्य की नकल कर सकें। 1950 में "ट्यूरिंग टेस्ट" का प्रस्ताव रखा गया।
1956	**AI शब्द की उत्पत्ति**: अमेरिका के डार्टमाउथ सम्मेलन में **John McCarthy** ने "Artificial Intelligence" शब्द का पहली बार उपयोग किया। इसे AI का जन्म माना जाता है।
1950–1970 के दशक	**प्रारंभिक AI प्रोग्राम**: साधारण समस्याएं हल करने वाले प्रोग्राम बनाए गए, जैसे गणितीय समस्या हल करना और खेल (जैसे चेस) खेलना।
1970–1980 का दशक	**AI विंटर**: अपेक्षाएं बहुत अधिक थीं लेकिन तकनीकी प्रगति धीमी रही। रिसर्च और

	फंडिंग में गिरावट आई। हालांकि, "एक्सपर्ट सिस्टम" जैसे MYCIN विकसित हुए।
1990 का दशक	**AI में नई ऊर्जा**: कंप्यूटर की ताकत और नई तकनीकों के साथ मशीन लर्निंग और डेटा माइनिंग में विकास हुआ। 1997 में **IBM का Deep Blue** शतरंज चैंपियन गैरी कास्परोव को हराता है।
2000 का दशक	**बिग डेटा का युग**: इंटरनेट और डेटा की भरमार के कारण स्पीच रिकग्निशन, इमेज प्रोसेसिंग और NLP में तेज़ विकास हुआ।
2010–2020 का दशक	**डीप लर्निंग क्रांति**: **Neural Networks** में बड़ी प्रगति। GPU की मदद से कंप्यूटर विज़न, NLP, और सेल्फ-ड्राइविंग कार जैसी टेक्नोलॉजी में बड़ा बदलाव। उदाहरण: **AlexNet (2012), AlphaGo (2016)**
2020–अब तक	**जनरेटिव AI का युग**: GPT-3, GPT-4, ChatGPT, DALL·E जैसी AI तकनीकें इंसानी जैसी भाषा और चित्र बना रही हैं। AI नीति, नैतिकता और पारदर्शिता पर भी ज़ोर।
भविष्य (2025 के बाद)	**जनरल AI** और **AI + मानव सहयोग** पर केंद्रित काम। स्वास्थ्य, शिक्षा, न्याय, और साइंस जैसी फील्ड में AI का विस्तार।

सारांश

- **AI** का मतलब: मशीनों को **मानव जैसी सोच और निर्णय लेने की शक्ति** देना।
- **शुरुआत**: 1956, डार्टमाउथ सम्मेलन से।
- **प्रमुख विकास**: लॉजिक बेस्ड → लर्निंग बेस्ड → जनरेटिव और अडवांस AI।
- **प्रमुख क्षेत्र**: मशीन लर्निंग, डीप लर्निंग, कंप्यूटर विजन, NLP, रोबोटिक्स।
- **AI के प्रकार**: नैरो AI, जनरल AI (भविष्य), सुपरइंटेलिजेंस (सैद्धांतिक)।

AI की मूल अवधारणाएँ

शब्द	अर्थ
AI	मशीन को बुद्धिमान बनाना
ML	मशीन को डेटा से सीखाना
DL	मानव मस्तिष्क जैसी सोच विकसित करना (न्यूरल नेटवर्क द्वारा)

एलन ट्यूरिंग और ट्यूरिंग टेस्ट

AI की नींव **एलन ट्यूरिंग** ने 1950 में रखी। उन्होंने एक टेस्ट प्रस्तावित किया जिसमें यदि एक मशीन इंसान की तरह संवाद कर सके तो उसे बुद्धिमान माना जाएगा।

ट्यूरिंग टेस्ट:

यदि कोई इंसान यह न पहचान पाए कि सामने इंसान है या मशीन, तो मशीन को बुद्धिमान माना जाएगा।

AI का ऐतिहासिक विकास

- **1950s**: एलन ट्यूरिंग का विचार
- **1956**: डार्टमाउथ सम्मेलन में AI शब्द का उपयोग
- **1997**: IBM का Deep Blue ने शतरंज चैंपियन को हराया
- **2011**: IBM Watson ने Jeopardy! गेम जीता
- **2016**: AlphaGo ने विश्व चैम्पियन को हराया
- **2020s**: ChatGPT और अन्य जनरेटिव AI

AI के प्रकार

1. संकीर्ण AI (Narrow AI):

- एक ही कार्य में कुशल
- उदाहरण: गूगल असिस्टेंट, सिरी

2. सामान्य AI (General AI):

- किसी भी कार्य को इंसानों की तरह करने में सक्षम
- अभी विकासशील अवस्था में

3. सुपर AI (Super AI):

- इंसानों से कहीं अधिक बुद्धिमान
- भविष्य की परिकल्पना

AI प्रकारों की तुलना

AI प्रकार	कार्यक्षमता	अवस्था
संकीर्ण AI	एक विशेष कार्य में कुशल	वर्तमान में उपलब्ध
सामान्य AI	बहु-कार्य प्रणाली	प्रयोगात्मक
सुपर AI	इंसानों से भी आगे	सैद्धांतिक

AI के अनुप्रयोग क्षेत्र

AI कई क्षेत्रों में क्रांति ला रहा है:

- स्वास्थ्य सेवा
- वित्त
- शिक्षा
- कृषि
- परिवहन
- मनोरंजन

स्वास्थ्य क्षेत्र में AI

- **डायग्नोसिस**: एक्स-रे और स्कैन की पहचान
- **वर्चुअल असिस्टेंट**: रोगी सहायता
- **ड्रग रिसर्च**: दवाइयों की खोज
- **रोबोटिक सर्जरी**: सटीकता में वृद्धि

वित्त क्षेत्र में AI

- **धोखाधड़ी पहचान**
- **क्रेडिट स्कोरिंग**
- **स्वचालित ट्रेडिंग**
- **चैटबॉट्स द्वारा ग्राहक सेवा**

शिक्षा में AI

- व्यक्तिगत अध्ययन
- आटोमैटिक मूल्यांकन
- भाषा सीखने वाले ऐप्स
- ऑनलाइन ट्यूटर

कृषि और उद्योग में AI

कृषि:

- फसल रोग की भविष्यवाणी
- उपज का अनुमान
- ड्रोन से खेतों की निगरानी

उद्योग:

- मशीनों की मरम्मत पूर्वानुमान
- गुणवत्ता नियंत्रण
- स्वचालित रोबोट

AI को अपनाने में चुनौतियाँ

- डेटा की गोपनीयता

- नैतिक मुद्दे
- महंगी तकनीक
- प्रशिक्षित लोगों की कमी
- पक्षपाती एल्गोरिद्म

मशीन लर्निंग (ML) क्या है?

मशीन लर्निंग एक AI की शाखा है जिसमें मशीनें अनुभव से सीखती हैं और भविष्यवाणी करती हैं।

कैसे काम करता है:

1. डेटा एकत्र करना
2. मॉडल बनाना
3. भविष्यवाणी करना
4. सुधार करना

मशीन लर्निंग के प्रकार

1. **सुपरवाइज्ड लर्निंग** – लेबल वाला डेटा
2. **अनसुपरवाइज्ड लर्निंग** – पैटर्न की खोज
3. **रीइन्फोर्समेंट लर्निंग** – प्रयास और पुरस्कार से सीखना

मशीन लर्निंग के एल्गोरिद्म

- लीनियर रिग्रेशन
- निर्णय वृक्ष (Decision Trees)
- K-मीन्स क्लस्टरिंग
- नाइव बेयस
- रैंडम फॉरेस्ट

ML के उपयोग

- वॉयस असिस्टेंट
- नेटफ्लिक्स/अमेज़न सिफारिशें
- धोखाधड़ी की पहचान
- बिक्री और स्टॉक पूर्वानुमान
- ग्राहक व्यवहार विश्लेषण

डीप लर्निंग क्या है?

डीप लर्निंग, मशीन लर्निंग की एक उपशाखा है जो आर्टिफिशियल न्यूरल नेटवर्क का प्रयोग करती है।

विशेषताएँ:

- गहरे (multi-layered) नेटवर्क
- बड़ी मात्रा में डेटा की आवश्यकता
- इमेज, ऑडियो, और वीडियो डेटा पर कार्य

डीप लर्निंग के अनुप्रयोग

- **इमेज पहचान**: फेसबुक टैगिंग
- **वॉयस रिकग्निशन**: गूगल असिस्टेंट
- **भाषा अनुवाद**: गूगल ट्रांसलेट
- **चैटबॉट्स**: ग्राहक सेवा

Chapter-2

Python for AI (पायथन फॉर एआई)

AI के लिए पायथन का परिचय

पायथन एक लोकप्रिय प्रोग्रामिंग भाषा है, जिसका उपयोग कृत्रिम बुद्धिमत्ता (AI) और मशीन लर्निंग (ML) में बड़े पैमाने पर होता है। इसकी सरल सिंटैक्स, विशाल पुस्तकालयें (libraries), और उपयोग में आसानी इसे AI के लिए आदर्श बनाती हैं।

इस ईबुक में आप सीखेंगे:

- पायथन प्रोग्रामिंग की बुनियादी बातें
- डेटा हैंडलिंग के लिए NumPy और Pandas
- डेटा विज़ुअलाइज़ेशन के लिए Matplotlib और Seaborn

AI में पायथन क्यों?

पायथन की खासियतें:

- पढ़ने में आसान और साफ़ सिंटैक्स
- विशाल पुस्तकालय जैसे NumPy, Pandas, TensorFlow
- प्लेटफार्म स्वतंत्रता
- बड़ी कम्युनिटी सहायता
- वेब और डेटा टूल्स से आसान एकीकरण

पायथन कैसे इंस्टॉल करें

1. https://python.org से डाउनलोड करें
2. कोड लिखने के लिए इस्तेमाल करें:
    - Jupyter Notebook
    - VS Code
    - Anaconda (सुझावित)

पायथन वर्शन चेक करें:

```
python --version
```

पायथन सिंटैक्स और इंडेंटेशन

पायथन में इंडेंटेशन (space/tab) कोड का हिस्सा होता है:

```
print("AI में आपका स्वागत है")

if True:
    print("यह ब्लॉक सही तरीके से इंडेंटेड है")
```

वेरिएबल्स और डेटा टाइप्स

सामान्य डेटा प्रकार:

- int: पूर्णांक (जैसे 10)
- float: दशमलव (जैसे 3.14)
- str: स्ट्रिंग (जैसे "AI")
- bool: True या False
- list: [1, 2, 3]
- dict: {"नाम": "AI", "उम्र": 5}

कंट्रोल स्ट्रक्चर

if-else कंडीशन:

```
if age > 18:
    print("वयस्क")
else:
    print("नाबालिग")
```

लूप:

```
for i in range(5):
    print(i)
```

फ़ंक्शन क्या होते हैं

```
def greet(name):
    return f"नमस्ते, {name}"

print(greet("विद्यार्थी"))
```

लिस्ट, टपल और डिक्शनरी

List:

```
fruits = ["सेब", "केला"]
print(fruits[0])
```

Tuple:

```
nums = (1, 2, 3)
```

Dictionary:

```
info = {"नाम": "AI", "प्रकार": "टूल"}
print(info["नाम"])
```

NumPy का परिचय

NumPy का उपयोग संख्यात्मक गणना (numerical computing) में होता है।

इंस्टॉल करें:

```
pip install numpy
```

उदाहरण:

```
import numpy as np
arr = np.array([1, 2, 3])
print(arr.mean())
```

NumPy ऐरे और फ़ंक्शन

```
a = np.array([[1, 2], [3, 4]])
print(a.shape)
```

महत्वपूर्ण फ़ंक्शन:

- `.mean()`
- `.sum()`
- `.reshape()`
- `.flatten()`

NumPy ऑपरेशन्स

```
a = np.array([1, 2])
b = np.array([3, 4])

print(a + b)
print(a * b)
print(np.dot(a, b))  # डॉट प्रोडक्ट
```

Pandas का परिचय

Pandas का उपयोग डेटा को टेबल के रूप में प्रोसेस करने के लिए होता है।

इंस्टॉल करें:

```
pip install pandas
```

उदाहरण:

```
import pandas as pd
data = {"नाम": ["राम", "श्याम"], "उम्र": [20, 22]}
df = pd.DataFrame(data)
print(df)
```

DataFrame का विश्लेषण

```
df.head()          # पहले 5 पंक्तियाँ
df.describe()      # आंकड़ों का सारांश
df["उम्र"].mean()  # औसत उम्र
```

डेटा क्लीनिंग

```
df.dropna()        # खाली मान हटाएं
df.fillna(0)       # खाली मान को 0 से भरें
df["उम्र"] = df["उम्र"].astype(int)
```

Matplotlib का उपयोग

इंस्टॉल करें:

```
pip install matplotlib
```

उदाहरण:

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3], [4, 5, 6])
plt.show()
```

ग्राफ्स के प्रकार

- लाइन ग्राफ
- बार चार्ट
- स्कैटर प्लॉट
- हिस्टोग्राम

```
plt.bar(["A", "B", "C"], [5, 7, 3])
plt.title("अंक")
plt.show()
```

Seaborn का परिचय

Seaborn सांख्यिकीय ग्राफ्स के लिए है।

इंस्टॉल करें:

```
pip install seaborn
```

उदाहरण:

```
import seaborn as sns
sns.histplot([10, 20, 20, 30])
```

Seaborn विज़ुअलाइजेशन

```
# हीटमैप
sns.heatmap(df.corr(), annot=True)

# लाइन प्लॉट
sns.lineplot(x="उम्र", y="अंक", data=df)
```

मिनी AI प्रोजेक्ट उदाहरण

स्टेप्स:

1. डेटा लोड करें (CSV)
2. साफ़ करें और विश्लेषण करें

3. Matplotlib/Seaborn से विजुअलाइज करें
4. अगली स्टेज में मॉडल बनाएं

Chapter-3

मशीन लर्निंग का परिचय (Basics of Machine Learning)

मशीन लर्निंग क्या है?

मशीन लर्निंग (ML) एक ऐसी तकनीक है जिसमें कंप्यूटर बिना स्पष्ट प्रोग्रामिंग के **डेटा से सीखते हैं** और भविष्यवाणी या निर्णय लेना सीखते हैं।

उदाहरण:

- गूगल का सर्च इंजन
- अमेज़न की प्रोडक्ट सिफारिशें
- चेहरा पहचानने वाले सिस्टम

मशीन लर्निंग के प्रकार

मुख्यतः मशीन लर्निंग के तीन प्रकार होते हैं:

1. Supervised Learning (पर्यवेक्षित अधिगम)

- डेटा लेबल होता है (इनपुट → आउटपुट पता होता है)।
- मॉडल उदाहरणों से सीखता है।

2. Unsupervised Learning (गैर-पर्यवेक्षित अधिगम)

- डेटा में कोई लेबल नहीं होता।
- एल्गोरिद्म डेटा में छिपे पैटर्न को पहचानता है।

3. Reinforcement Learning (पुनर्बलन अधिगम)

- मॉडल इनाम और दंड के माध्यम से सीखता है। (इस पुस्तक में संक्षेप में)

Supervised Learning क्या है?

इस प्रकार की लर्निंग में हमारे पास **इनपुट और उससे संबंधित आउटपुट** होते हैं।

उदाहरण:

- छात्र की पढ़ाई के घंटों के आधार पर अंक की भविष्यवाणी
- ईमेल को स्पैम या नॉन-स्पैम में वर्गीकृत करना

यह दो भागों में बंटा होता है:

- **Regression (प्रत्यावर्तन)**
- **Classification (वर्गीकरण)**

Regression (प्रत्यावर्तन) क्या है?

Regression का उपयोग तब किया जाता है जब आउटपुट **संख्यात्मक (continuous)** होता है।

उदाहरण:

- घर की कीमत की भविष्यवाणी
- तापमान का पूर्वानुमान

साधारण रेखीय प्रत्यावर्तन (Simple Linear Regression):

```
y = mx + c
```

जहाँ,
x = इनपुट
y = भविष्यवाणी
m = ढलान
c = इंटरसेप्ट

Classification (वर्गीकरण) क्या है?

Classification का उपयोग तब होता है जब आउटपुट एक **श्रेणी (category/label)** होती है।

उदाहरण:

- ईमेल: स्पैम या नॉन-स्पैम
- बीमारी का निदान: सकारात्मक या नकारात्मक

लोकप्रिय एल्गोरिद्म:

- Logistic Regression
- Decision Tree
- KNN (K-Nearest Neighbors)
- SVM (Support Vector Machine)

Unsupervised Learning क्या है?

इस विधि में डेटा लेबल रहित होता है, और मॉडल **स्वतः पैटर्न या ग्रुप** खोजता है।

मुख्य तकनीक:

- **Clustering (समूह बनाना)** – जैसे K-Means
- **Dimensionality Reduction** – जैसे PCA

उदाहरण:

- ग्राहक वर्गीकरण
- सोशल मीडिया यूजर एनालिसिस

Scikit-learn क्या है?

Scikit-learn एक लोकप्रिय Python लाइब्रेरी है जिसका उपयोग मशीन लर्निंग मॉडल को बनाने, ट्रेन करने और परीक्षण करने के लिए किया जाता है।

विशेषताएं:

- Supervised और Unsupervised एल्गोरिद्म
- Data Preprocessing टूल्स
- Model Evaluation तकनीकें

इंस्टॉलेशन:

```
pip install scikit-learn
```

मशीन लर्निंग वर्कफ़्लो

1. लाइब्रेरी इम्पोर्ट करें
2. डेटा लोड करें
3. प्रशिक्षण और परीक्षण सेट में विभाजन करें
4. मॉडल चुनें और ट्रेन करें
5. भविष्यवाणी करें
6. प्रदर्शन का मूल्यांकन करें

उदाहरण: Linear Regression का उपयोग

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

# डेटा लोड करें
data = pd.read_csv("student_scores.csv")
X = data[["Hours"]]
y = data["Scores"]

# डेटा विभाजित करें
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

# मॉडल ट्रेन करें
model = LinearRegression()
model.fit(X_train, y_train)

# भविष्यवाणी करें
predictions = model.predict(X_test)
print(predictions)
```

उदाहरण: Classification KNN से

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

# डेटासेट लोड करें
iris = load_iris()
X = iris.data
y = iris.target

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

# मॉडल बनाएं
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)

# भविष्यवाणी करें
preds = model.predict(X_test)
print(preds)
```

मॉडल मूल्यांकन मापदंड

Regression के लिए:

- MAE (Mean Absolute Error)
- MSE (Mean Squared Error)
- R^2 Score

Classification के लिए:

- Accuracy
- Confusion Matrix
- Precision, Recall, F1 Score

शुरुआती के लिए सुझाव

- डेटा को हमेशा विज़ुअलाइज़ करें
- आउटपुट की जांच करें
- मॉडल की तुलना करें
- डेटासेट जैसे Iris, Titanic पर अभ्यास करें

Chapter-4

मिनी प्रोजेक्ट और टूल्स

भूमिका (Introduction)

मशीन लर्निंग और आर्टिफिशियल इंटेलिजेंस (AI) की वास्तविक शक्ति तभी समझ आती है जब आप उसे प्रायोगिक रूप से करते हैं। इस अध्याय में आप सीखेंगे:

- कैसे एक बेसिक मशीन लर्निंग मॉडल बनाएँ
- ChatGPT और Google Teachable Machine जैसे टूल्स का उपयोग
- एक अंतिम क्विज़ और स्वयं मूल्यांकन

एक साधारण मशीन लर्निंग मॉडल बनाना

हम दो आसान प्रोजेक्ट बनाएँगे:

1. **स्पैम ईमेल डिटेक्टर (Spam Detector)**
2. **घर की कीमत का पूर्वानुमान (House Price Predictor)**

प्रोजेक्ट 1: स्पैम ईमेल डिटेक्टर

उद्देश्य:
ईमेल को "स्पैम" या "नॉन-स्पैम" के रूप में वर्गीकृत करना।

प्रयोग की जाने वाली तकनीक:

- Naive Bayes एल्गोरिदम
- Python, Pandas, Sklearn

उदाहरण कोड:

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

# डेटा लोड करें
data = pd.read_csv("spam.csv", encoding='latin-1')
data = data[['v1', 'v2']]
data.columns = ['label', 'message']
```

```
data['label'] = data['label'].map({'ham': 0, 'spam': 1})

# डेटा विभाजन
X_train, X_test, y_train, y_test =
train_test_split(data['message'], data['label'],
test_size=0.2)

# फीचर एक्सट्रैक्शन
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# मॉडल ट्रेनिंग
model = MultinomialNB()
model.fit(X_train_vec, y_train)

# मूल्यांकन
preds = model.predict(X_test_vec)
print("Accuracy:", accuracy_score(y_test, preds))
```

प्रोजेक्ट 2: घर की कीमत का पूर्वानुमान

उद्देश्य:
दिए गए एरिया और बेडरूम की संख्या के आधार पर मकान की कीमत की भविष्यवाणी करना।

उदाहरण कोड:

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

data = pd.read_csv('housing.csv')
X = data[['Area', 'Bedrooms']]
y = data['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)

preds = model.predict(X_test)
print(preds[:5])
```

ChatGPT क्या है?

ChatGPT OpenAI द्वारा विकसित एक **कृत्रिम बुद्धिमत्ता आधारित चैटबॉट** है। यह प्राकृतिक भाषा में संवाद कर सकता है।

विशेषताएं:

- कंटेंट लेखन
- कोडिंग में सहायता
- प्रश्नों के उत्तर
- भाषा अनुवाद
- शिक्षा और सहायता

कैसे उपयोग करें:

1. https://chat.openai.com पर जाएँ
2. लॉगिन करें
3. कोई भी प्रश्न या कमांड टाइप करें

उदाहरण:
"Naive Bayes को उदाहरण के साथ समझाओ"
ChatGPT कुछ ही सेकंड में समझा देगा।

Teachable Machine (टीचेबल मशीन)

Teachable Machine गूगल का एक वेब टूल है जो आपको कोड लिखे बिना मॉडल बनाने की सुविधा देता है।

आप क्या बना सकते हैं:

- इमेज क्लासिफायर
- साउंड पहचानने वाला मॉडल
- शरीर की मुद्रा (Pose) डिटेक्टर

कैसे प्रयोग करें:

1. https://teachablemachine.withgoogle.com पर जाएँ
2. Image/Sound/Pose में से एक चुनें
3. उदाहरण रिकॉर्ड करें या अपलोड करें
4. ट्रेन करें और मॉडल डाउनलोड करें

उदाहरण:

- अपने चेहरे की मुस्कान पहचानने वाला मॉडल
- लाइव कैमरा से टेस्ट करना

अंतिम क्विज़ (Final Quiz)

प्रश्न 1:
कौन-सा एल्गोरिदम स्पैम पहचानने के लिए उपयुक्त है?
A. KNN
B. Naive Bayes □
C. PCA
D. Random Forest

प्रश्न 2:
ChatGPT को किसने बनाया है?
A. Google
B. Amazon
C. OpenAI □
D. Facebook

प्रश्न 3:
Teachable Machine का उपयोग क्या है?
A. डेटा विश्लेषण
B. मॉडल कोडिंग
C. बिना कोडिंग मॉडल बनाना □
D. क्लाउड स्टोरेज

प्रश्न 4:
Linear Regression का आउटपुट कैसा होता है?
A. Category
B. Number □
C. Text
D. Graph

(कुल 10-15 प्रश्न डालें अभ्यास के लिए)

स्वयं मूल्यांकन और फीडबैक

□ **चेकलिस्ट:**

- ☐ मैं Supervised Learning और Unsupervised Learning में अंतर समझता/समझती हूँ
- ☐ मैं एक सरल ML मॉडल बना सकता/सकती हूँ

- ☐ मैंने ChatGPT और Teachable Machine का प्रयोग किया है
- ☐ मैंने क्विज़ में भाग लिया

फीडबैक प्रश्न:

1. आपको सबसे मज़ेदार प्रोजेक्ट कौन-सा लगा?
2. आपने किस टूल से सबसे ज़्यादा सीखा?
3. क्या आप AI में आगे और गहराई से पढ़ना चाहेंगे?

Chapter-5

एडवांस्ड मशीन लर्निंग एल्गोरिद्म्स

परिचय

इस ईबुक में हम उन मशीन लर्निंग एल्गोरिद्म्स को समझेंगे जो वास्तविक परियोजनाओं और पेशेवर कामों में सबसे अधिक उपयोग होते हैं:

- निर्णय वृक्ष (Decision Tree)
- रैंडम फॉरेस्ट (Random Forest)
- के-नियरस्ट नेबर (K-Nearest Neighbors, KNN)
- मॉडल मूल्यांकन (Confusion Matrix, Precision, Recall)

निर्णय वृक्ष (Decision Tree)

क्या है निर्णय वृक्ष?

यह एक ग्राफ़ जैसा ढाँचा होता है जो "हां/ना" आधारित निर्णय लेकर डेटा को वर्गीकृत करता है।

मुख्य घटक:

- **रूट नोड** – प्रारंभिक निर्णय बिंदु
- **इंटरनल नोड** – विशेषताओं पर आधारित प्रश्न
- **लीफ नोड** – अंतिम निर्णय या वर्ग

विभाजन मानदंड:

- Gini Index
- Entropy (Information Gain)

उदाहरण (Python कोड):

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
```

लाभ:

✔ सरल और समझने में आसान
✔श्रेणीबद्ध और संख्यात्मक दोनों डेटा के लिए उपयुक्त

हानि:

- ओवरफिटिंग की संभावना
- हल्के बदलाव से अस्थिर हो सकता है

रैंडम फॉरेस्ट (Random Forest)

क्या है रैंडम फॉरेस्ट?

यह कई निर्णय वृक्षों का समूह होता है। हर वृक्ष अलग-अलग डेटा सैंपल पर प्रशिक्षित होता है।

कार्यविधि:

1. अलग-अलग डेटा सैंपल पर कई Decision Tree बनाए जाते हैं
2. प्रत्येक पेड़ अपनी भविष्यवाणी करता है
3. अंतिम निर्णय बहुमत के आधार पर होता है

कोड उदाहरण:

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
```

लाभ:

✔ अधिक सटीकता
✔ ओवरफिटिंग में कमी
✔ आउटपुट स्थिर रहता है

हानि:

- जटिल और धीमा
- विज़ुअलाइज़ करना मुश्किल

के-नियरस्ट नेबर (KNN)

परिचय:

KNN एक सरल लेकिन शक्तिशाली एल्गोरिद्म है जो किसी नए डेटा पॉइंट को उसके पास के 'K' सबसे नजदीकी डेटा पॉइंट्स के आधार पर वर्गीकृत करता है।

मुख्य बातें:

- **K का चुनाव महत्वपूर्ण**
- दूरी का मापन (Euclidean या Manhattan Distance)
- वर्ग सबसे अधिक मिलने वाले लेबल के आधार पर होता है

कोड उदाहरण:

```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
```

लाभ:

✔ बिना प्रशिक्षण काम करता है
✔ सरल और व्याख्यायोग्य

हानि:

□ बड़े डेटा पर धीमा
□ आउटलेयर के प्रति संवेदनशील

मॉडल मूल्यांकन की आवश्यकता

केवल Accuracy देखना पर्याप्त नहीं है, खासकर जब क्लास असंतुलित हों। इसके लिए हमें अन्य मापदंडों की जरूरत होती है:

- Confusion Matrix
- Precision
- Recall
- F1 Score

कन्फ्यूजन मैट्रिक्स (Confusion Matrix)

संरचना:

वास्तविक भविष्यवाणी	भविष्यवाणी: सकारात्मक	भविष्यवाणी: नकारात्मक
वास्तविक: सकारात्मक	True Positive (TP)	False Negative (FN)
वास्तविक: नकारात्मक	False Positive (FP)	True Negative (TN)

कोड उदाहरण:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

प्रिसीजन और रिकॉल (Precision & Recall)

Precision (सटीकता)

Precision = TP / (TP + FP)
इससे यह पता चलता है कि भविष्यवाणी किए गए पॉजिटिव्स में से कितने सही थे।

Recall (पुनः प्राप्ति)

Recall = TP / (TP + FN)
यह दर्शाता है कि सभी वास्तविक पॉजिटिव्स में से कितनों को सही पहचाना गया।

उदाहरण:

- **Precision ज़रूरी होता है** स्पैम डिटेक्शन जैसे मामलों में
- **Recall ज़रूरी होता है** कैंसर डिटेक्शन जैसे मामलों में

एफ1 स्कोर (F1 Score)

Precision और Recall के संतुलन के लिए F1 Score का उपयोग किया जाता है।

F1 = 2 * (Precision * Recall) / (Precision + Recall)

यह विशेष रूप से उपयोगी होता है जब डेटा असंतुलित होता है।

प्रैक्टिकल अभ्यास

Dataset:

- Iris
- Breast Cancer

कार्य:

1. Decision Tree, Random Forest, KNN मॉडल ट्रेन करें
2. Accuracy और Confusion Matrix निकालें
3. Precision, Recall, F1 Score प्रदर्शित करें

क्विज़ (Quiz)

1. Decision Tree का सबसे बड़ा नुकसान क्या है?
   A. बहुत जटिल
   B. ओवरफिटिंग □
   C. कम Accuracy
   D. स्लो
2. Random Forest क्या है?
   A. एक Neural Network
   B. एक Decision Tree
   C. कई Decision Trees का समूह □
   D. कोई एल्गोरिद्म नहीं
3. Precision किसका अनुपात है?
   A. TP / (TP + FN)
   B. TP / (TP + FP) □
   C. TN / (TN + FP)
   D. FN / TP

और प्रश्न जोड़े जा सकते हैं...

निष्कर्ष

एल्गोरिद्म	उपयोग	लाभ	सीमा
Decision Tree	वर्गीकरण	सरल और स्पष्ट	ओवरफिटिंग
Random Forest	वर्गीकरण/रिग्रेशन	अधिक सटीक	व्याख्या में जटिलता
KNN	वर्गीकरण	कोई प्रशिक्षण नहीं	बड़े डेटा पर धीमा

Chapter-6

न्यूरल नेटवर्क का परिचय

(Perceptron, ANN, Activation Functions और TensorFlow/Keras)

न्यूरल नेटवर्क क्या हैं?

न्यूरल नेटवर्क एक कंप्यूटेशनल मॉडल है जो इंसानी मस्तिष्क की कार्यप्रणाली से प्रेरित होता है। ये मशीन लर्निंग की सबसे उन्नत शाखा — **Deep Learning** — का आधार हैं।

जैविक बनाम कृत्रिम न्यूरॉन

जैविक न्यूरॉन	कृत्रिम न्यूरॉन
शरीर के न्यूरॉन्स से प्रेरित	कंप्यूटेशनल यूनिट
संकेतों को भेजता और प्राप्त करता है	डेटा इनपुट लेता है और आउटपुट देता है
विद्युत आवेगों से कार्य करता है	गणना द्वारा कार्य करता है

परसेप्ट्रॉन (Perceptron) क्या है?

परसेप्ट्रॉन न्यूरल नेटवर्क का सबसे सरल रूप है। यह 1958 में फ्रैंक रोसेनब्लैट द्वारा विकसित किया गया था।

घटक:

- इनपुट्स (x_1, x_2, …)
- वेट्स (w_1, w_2, …)
- बायस (bias)
- एक्टीवेशन फंक्शन
- आउटपुट (y)

समीकरण:

$$y = \text{Activation}(w_1x_1 + w_2x_2 + \ldots + b)$$

परसेप्ट्रॉन की सीमाएँ

- केवल **रेखीय रूप से विभाज्य** समस्याओं को हल कर सकता है।
- XOR जैसी समस्याओं को नहीं सुलझा सकता।
- छिपे हुए लेयर का अभाव।

आर्टिफिशियल न्यूरल नेटवर्क (ANN) का परिचय

ANN में तीन प्रमुख लेयर होते हैं:

- **इनपुट लेयर**
- **हिडन लेयर (छुपे हुए परतें)**
- **आउटपुट लेयर**

हर लेयर में कई न्यूरॉन्स होते हैं, जो एक-दूसरे से जुड़े होते हैं।

फीड फॉरवर्ड और बैकप्रोपेगेशन

- **Feedforward**: इनपुट से आउटपुट तक डेटा का प्रवाह।
- **Backpropagation**: त्रुटियों के आधार पर वेट्स को अपडेट करना।

वेट्स और बायस का महत्व

- वेट्स इनपुट की अहमियत तय करते हैं।
- बायस आउटपुट को शिफ्ट करता है।
- सही वेट्स सीखने के लिए हमें लॉस फंक्शन की जरूरत होती है।

लॉस फंक्शन क्या होता है?

लॉस फंक्शन, मॉडल की भविष्यवाणी और वास्तविक मूल्य के बीच का अंतर मापता है।

उदाहरण:

- रिग्रेशन के लिए: Mean Squared Error
- क्लासिफिकेशन के लिए: Cross Entropy

एक्टीवेशन फंक्शन का परिचय

एक्टीवेशन फंक्शन तय करता है कि न्यूरॉन "फायर" करेगा या नहीं। ये नॉन-लिनियरिटी को मॉडल में जोड़ते हैं।

सामान्य एक्टीवेशन फंक्शन्स

1. **Sigmoid**
   सूत्र: `σ(x) = 1 / (1 + e^-x)`
   आउटपुट: 0 से 1 के बीच
2. **Tanh**
   आउटपुट: -1 से 1 के बीच
   केंद्रित आउटपुट देता है
3. **ReLU (Rectified Linear Unit)**
   सूत्र: `ReLU(x) = max(0, x)`
   आजकल सबसे अधिक उपयोगी

उन्नत एक्टीवेशन फंक्शन

- **Leaky ReLU**: ReLU के डेड न्यूरॉन समस्या को सुलझाता है
- **Softmax**: मल्टी-क्लास क्लासिफिकेशन के लिए उपयोगी

न्यूरल नेटवर्क कैसे बनाएं?

1. डेटा तैयार करें
2. लेयर और न्यूरॉन्स तय करें
3. एक्टीवेशन चुनें
4. लॉस और ऑप्टिमाइज़र तय करें
5. प्रशिक्षण (Training)
6. मूल्यांकन (Evaluation)

TensorFlow/Keras का परिचय

TensorFlow: गूगल द्वारा विकसित ओपन-सोर्स डिप लर्निंग लाइब्रेरी
Keras: TensorFlow के ऊपर एक सरल और उपयोगकर्ता-अनुकूल API

क्यों उपयोग करें TensorFlow/Keras?

✔ कम कोड में शक्तिशाली मॉडल
✔ GPU/TPU सपोर्ट
✔ Visualization Tools
✔ व्यापक दस्तावेज और समुदाय समर्थन

TensorFlow/Keras इंस्टॉल करें

```
pip install tensorflow
```

Python में उपयोग:

```
import tensorflow as tf
from tensorflow import keras
```

Keras से एक सिंपल मॉडल बनाएँ

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([
    Dense(16, activation='relu', input_shape=(10,)),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32)
```

मॉडल का सारांश और विज़ुअलाइज़ेशन

```
model.summary()
```

विज़ुअलाइज़ करने के लिए:

```
from tensorflow.keras.utils import plot_model
plot_model(model, show_shapes=True)
```

उपयोग के क्षेत्र

- चेहरा पहचानना
- आवाज़ पहचानना
- रोग पहचान
- ऑब्जेक्ट डिटेक्शन
- चैटबॉट्स

Chapter-7

AI टूल्स और एप्लिकेशन – एक व्यावहारिक गाइड

भूमिका

कृत्रिम बुद्धिमत्ता (AI) ने तकनीक की दुनिया को पूरी तरह से बदल दिया है। आज हम जिन स्मार्ट फीचर्स का उपयोग करते हैं—जैसे कि चेहरा पहचानना, ऑटोमेटिक उत्तर देना, या भाषाओं का अनुवाद—इन सबके पीछे AI तकनीक है।

इमेज रिकग्निशन क्या है?

इमेज रिकग्निशन एक AI तकनीक है जो तस्वीरों या वीडियो में वस्तुओं, चेहरों, स्थानों, या गतिविधियों की पहचान करती है।

उदाहरण:

- स्मार्टफोन में फेस अनलॉक
- सुरक्षा कैमरों में चेहरा पहचान
- सेल्फ-ड्राइविंग कारों में सिग्नल और बाधा पहचान

इमेज रिकग्निशन कैसे काम करता है?

1. इमेज को पिक्सेल डाटा में बदला जाता है
2. विशेषताएं निकाली जाती हैं (जैसे रंग, किनारे, आकृति)
3. मॉडल इन विशेषताओं की तुलना सीखें गए उदाहरणों से करता है
4. आउटपुट प्रदर्शित किया जाता है (उदाहरण: "यह एक बिल्ली है")

OpenCV टूल का परिचय

OpenCV (Open Source Computer Vision) एक ओपन-सोर्स लाइब्रेरी है जिसका उपयोग इमेज प्रोसेसिंग और कंप्यूटर विजन के लिए किया जाता है।

✔ चेहरा पहचानना
✔ वस्तु ट्रैकिंग
✔ वीडियो स्ट्रीम पर प्रोसेसिंग
✔ फिल्टर लगाना

OpenCV कोड उदाहरण (चेहरा पहचान)

```
import cv2
face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
img = cv2.imread('image.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.1, 4)
```

इमेज रिकग्निशन के अनुप्रयोग

- **स्वास्थ्य:** MRI/CT स्कैन में बीमारी पहचान
- **रिटेल:** स्वचालित बिलिंग सिस्टम
- **ट्रैफिक:** वाहन पहचान, नंबर प्लेट रीडिंग
- **सुरक्षा:** फेस डिटेक्शन सिस्टम

NLP (नेचुरल लैंग्वेज प्रोसेसिंग) क्या है?

NLP वह AI क्षेत्र है जो मानव भाषा को समझने, विश्लेषण करने और उत्पन्न करने में मदद करता है।

प्रमुख कार्य:

- भाव विश्लेषण (Sentiment Analysis)
- अनुवाद (Translation)
- चैटबॉट
- टेक्स्ट जनरेशन

NLP कैसे काम करता है?

1. **टोकनाइज़ेशन:** वाक्य को शब्दों में तोड़ना
2. **स्टॉप वर्ड हटाना:** "है", "क्या" जैसे सामान्य शब्द हटाना
3. **वेक्टराइज़ेशन:** शब्दों को अंकों में बदलना
4. **मॉडलिंग:** भावना विश्लेषण, श्रेणी निर्धारण आदि

HuggingFace का परिचय

HuggingFace एक प्रमुख NLP प्लेटफ़ॉर्म है जो प्री-ट्रेंड मॉडल्स प्रदान करता है जैसे कि BERT, GPT, RoBERTa आदि।

डेमो टूल्स:

- टेक्स्ट क्लासिफिकेशन
- भाषा अनुवाद
- टेक्स्ट जनरेशन

HuggingFace का उपयोग कैसे करें?

स्टेप 1: वेबसाइट खोलें – https://huggingface.co
स्टेप 2: कोई मॉडल चुनें (जैसे GPT-2)
स्टेप 3: टेक्स्ट इनपुट करें और आउटपुट देखें
स्टेप 4: Python कोड से भी उपयोग करें:

```
from transformers import pipeline
classifier = pipeline("sentiment-analysis")
print(classifier("AI बहुत उपयोगी है"))
```

चैटबॉट क्या हैं?

चैटबॉट एक सॉफ्टवेयर है जो इंसानों की तरह बातचीत करता है। यह NLP और AI की मदद से सवालों का उत्तर देता है।

प्रकार:

- नियम-आधारित चैटबॉट
- AI आधारित चैटबॉट (जैसे ChatGPT)

ChatGPT का परिचय

ChatGPT एक उन्नत AI चैटबॉट है जो OpenAI द्वारा विकसित किया गया है। यह भाषा समझने, उत्तर देने, ईमेल लिखने, अनुवाद करने आदि में सक्षम है।

वेबसाइट: https://chat.openai.com

चैटबॉट के उपयोग

- ग्राहक सहायता (Customer Support)
- बैंकिंग सेवाएं

- ऑनलाइन शिक्षा
- हेल्पडेस्क सिस्टम

मल्टीमॉडल AI – इमेज + टेक्स्ट

AI अब इमेज और टेक्स्ट दोनों को एक साथ समझने में सक्षम है।

उदाहरण: एक ऐसा मॉडल जो चित्र देखकर उसका वर्णन दे (Image Captioning)।

प्रसिद्ध टूल्स:

- OpenAI CLIP
- BLIP (Bootstrapped Language Image Pretraining)

अन्य उपयोगी AI टूल्स

टूल	उपयोग
TensorFlow	डीप लर्निंग मॉडल बनाना
Keras	न्यूरल नेटवर्क डिजाइन करना
FastAI	हाई-लेवल ML APIs
Scikit-learn	बेसिक मशीन लर्निंग एल्गोरिद्म
Teachable Machine	बिना कोड के मॉडल बनाना

AI एप्लिकेशन – उद्योग अनुसार

क्षेत्र	अनुप्रयोग
स्वास्थ्य	बीमारी की पहचान, हेल्थ रिपोर्टिंग
शिक्षा	ऑटो-ग्रेडिंग, पर्सनलाइज्ड लर्निंग
खुदरा बिक्री	ग्राहक सुझाव, चेहरा पहचान
परिवहन	ऑटो ड्राइविंग, ट्रैफिक विश्लेषण

Google Teachable Machine

Google Teachable Machine एक आसान ऑनलाइन टूल है जिससे आप बिना कोड के मॉडल बना सकते हैं।

✔ इमेज क्लासिफिकेशन
✔ आवाज पहचान
✔ पोज़ पहचान

वेबसाइट: https://teachablemachine.withgoogle.com

प्रैक्टिकल प्रोजेक्ट सुझाव

1. OpenCV से फेस मास्क डिटेक्टर बनाएं
2. HuggingFace से भाव विश्लेषण टूल बनाएँ
3. Dialogflow से अपना खुद का चैटबॉट बनाएं

AI टूल्स की तुलना सारांश

कार्य	टूल
इमेज प्रोसेसिंग	OpenCV
टेक्स्ट जनरेशन	Hugging Face / GPT
चैटबॉट निर्माण	ChatGPT / Dialogflow
बिना कोड AI मॉडल	Teachable Machine

Chapter-8

एआई कोर्स: प्रोजेक्ट और प्रेजेंटेशन

परिचय

प्रोजेक्ट किसी भी कोर्स का सबसे महत्वपूर्ण हिस्सा होता है। इससे विद्यार्थियों को अपने ज्ञान को व्यवहार में लाने का मौका मिलता है। इस खंड में हम एक छोटा एआई प्रोजेक्ट बनाएंगे और उसके प्रस्तुतिकरण की तैयारी करेंगे।

प्रोजेक्ट क्यों ज़रूरी हैं?

- सीखी गई बातों को अमल में लाने का अवसर
- रियल वर्ल्ड सॉल्यूशन्स बनाने की क्षमता
- कोडिंग से प्रेजेंटेशन तक की पूरी प्रक्रिया को समझना
- पोर्टफोलियो बनाने के लिए उपयोगी

मिनी प्रोजेक्ट 1 – इमेज क्लासिफायर

उद्देश्य:
एक ऐसा मॉडल बनाना जो इमेज को कैटेगरी में बांट सके, जैसे "बिल्ली या कुत्ता", या "0 से 9 तक की हस्तलिखित संख्या"।

उपयोग किए गए टूल्स:

- Python
- TensorFlow / Keras
- Jupyter Notebook
- Dataset (जैसे MNIST या CIFAR-10)

इमेज क्लासिफायर बनाने के स्टेप्स

1. लाइब्रेरी इम्पोर्ट करें
2. डेटासेट लोड करें
3. डेटा को प्रोसेस करें
4. मॉडल बनाएं (Convolutional Neural Network)
5. मॉडल को ट्रेन करें
6. टेस्ट और इवैल्युएट करें
7. परिणामों का विश्लेषण करें

इमेज क्लासिफायर का सैंपल कोड

```
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train/255.0, x_test/255.0

model = Sequential([
    Flatten(input_shape=(28,28)),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5)
```

मिनी प्रोजेक्ट 2 – चैटबॉट

उद्देश्य:
एक सरल चैटबॉट बनाना जो सामान्य सवालों के उत्तर दे सके।

उपयोग किए गए टूल्स:

- Python
- NLTK या ChatterBot
- वैकल्पिक: Streamlit या Flask (UI के लिए)

चैटबॉट बनाने की प्रक्रिया

1. सवालों की सूची या इंटेंट बनाना
2. संवाद की स्क्रिप्ट बनाना
3. मॉडल को ट्रेन करना
4. टेस्टिंग और UI बनाना
5. उत्तरों का सुधार

चैटबॉट का सैंपल कोड

```
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer

bot = ChatBot('AI Bot')
trainer = ChatterBotCorpusTrainer(bot)
trainer.train("chatterbot.corpus.hindi")

while True:
```

```
question = input("आप: ")
print("बॉट:", bot.get_response(question))
```

प्रोजेक्ट का चुनाव कैसे करें?

प्रोजेक्ट	उपयुक्त विद्यार्थियों के लिए
इमेज क्लासिफायर	जो विजुअल डेटा में रुचि रखते हैं
चैटबॉट	जिन्हें NLP और संवाद पसंद है

आवश्यक टूल्स की सूची

- Python
- Jupyter Notebook / Google Colab
- TensorFlow / Keras
- ChatterBot / NLTK
- Streamlit (प्रस्तुतिकरण के लिए)

प्रोजेक्ट रिपोर्ट कैसे बनाएं?

1. प्रोजेक्ट का नाम और उद्देश्य
2. उपयोग किए गए टूल्स और डेटा
3. कोडिंग स्टेप्स (स्क्रीनशॉट के साथ)
4. परीक्षण और परिणाम
5. कठिनाइयाँ और समाधान
6. निष्कर्ष

प्रस्तुति कौशल

- स्लाइड्स को सरल रखें
- हर स्लाइड को स्पष्ट रूप से समझाएं
- चित्र, चार्ट और कोड स्निपेट्स का उपयोग करें
- आत्मविश्वास के साथ बोलें

प्रेजेंटेशन का प्रारूप

1. शीर्षक स्लाइड
2. उद्देश्य और समस्या विवरण

3. डेटासेट / टूल्स
4. वर्कफ़्लो
5. कोड स्निपेट
6. परिणाम
7. निष्कर्ष

प्रस्तुति के समय ध्यान देने योग्य बातें

- स्लाइड पढ़ें नहीं, समझाएं
- श्रोताओं के साथ संपर्क बनाए रखें
- समय का ध्यान रखें
- प्रश्न पूछने के लिए प्रेरित करें

मूल्यांकन मानदंड

मानदंड	अंक
नवाचार	20
कोड कार्यान्वयन	30
सटीकता	20
रिपोर्ट	15
प्रस्तुति	15

फीडबैक और सुधार

प्रस्तुति के बाद फीडबैक लें:

- सबसे रोचक भाग क्या था?
- क्या और सुधार किया जा सकता है?
- क्या परिणाम समझ में आए?

Chapter-9

डीप लर्निंग की मूल बातें (Deep Learning Essentials)

डीप लर्निंग का परिचय

डीप लर्निंग मशीन लर्निंग की एक उप-शाखा है, जिसमें **कृत्रिम तंत्रिका नेटवर्क (Artificial Neural Networks)** का प्रयोग होता है। इसमें अनेक स्तरों (Layers) में डेटा को प्रोसेस किया जाता है, जिससे जटिल समस्याओं का समाधान किया जा सकता है।

डीप लर्निंग क्यों?

- बड़ी मात्रा में डेटा का विश्लेषण करने की क्षमता
- इमेज, टेक्स्ट, वॉइस जैसे अनस्ट्रक्चर्ड डेटा पर बेहतर प्रदर्शन
- चैटबॉट, सेल्फ-ड्राइविंग कार, चेहरा पहचान प्रणाली आदि में प्रयोग

न्यूरल नेटवर्क की मूल संरचना

न्यूरल नेटवर्क मुख्यतः तीन भागों में बाँटा जाता है:

1. **इनपुट लेयर (Input Layer)**
2. **हिडन लेयर्स (Hidden Layers)**
3. **आउटपुट लेयर (Output Layer)**

हर नोड या न्यूरॉन एक छोटे से गणितीय कार्य की तरह काम करता है।

CNNs का परिचय

कॉन्वोल्यूशनल न्यूरल नेटवर्क (CNN) इमेज प्रोसेसिंग के लिए एक प्रसिद्ध आर्किटेक्चर है। यह इमेज से फीचर एक्सट्रैक्ट करता है और बिना मैनुअल प्री-प्रोसेसिंग के इमेज को पहचानने में सक्षम है।

CNN की परतें (Layers)

1. **कॉन्वोल्यूशन लेयर**: फिल्टर्स का प्रयोग कर इमेज से विशेषताएं निकाली जाती हैं।
2. **ReLU एक्टिवेशन**: नॉन-लाइनियरिटी लाने के लिए।
3. **पूलिंग लेयर**: इमेज साइज को घटाता है।
4. **फ्लैटन लेयर**: डेटा को एक वेक्टर में बदलता है।
5. **फुली कनेक्टेड लेयर**: अंतिम वर्गीकरण करता है।

Convolution क्या है?

Convolution एक गणितीय प्रक्रिया है जिसमें एक **फिल्टर (kernel)** को इमेज के ऊपर घुमाया जाता है और इमेज के छोटे-छोटे पैटर्न निकाले जाते हैं जैसे कि किनारे, आकृति, रंग।

CNN का उदाहरण

बिल्ली या कुत्ता पहचानने वाला मॉडल:
इंटरनेट से ली गई इमेज को CNN द्वारा प्रोसेस करके मॉडल यह अनुमान लगाता है कि वह इमेज किस कैटेगरी की है।

CNN का बेसिक कोड (Keras)

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
Flatten, Dense

model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(64, 64,
3)),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])
```

RNNs का परिचय

रिकरंट न्यूरल नेटवर्क (RNN) सीक्वेंस आधारित डेटा जैसे कि टेक्स्ट, स्पीच, टाइम-सीरीज़ डेटा के लिए प्रयुक्त होता है। RNN में पिछली इनपुट की जानकारी को "मेमोरी" के रूप में उपयोग किया जाता है।

RNN कैसे कार्य करता है?

- हर समय एक नया इनपुट लिया जाता है
- पिछले स्टेट (memory) को ध्यान में रखते हुए आउटपुट उत्पन्न होता है
- यह मॉडल टाइम-सीरीज़ और भाषा समझने में कुशल होता है

RNN का उपयोग

- भाषा अनुवाद

- टेक्स्ट जनरेशन
- स्पीच टू टेक्स्ट
- मौसम या स्टॉक प्रेडिक्शन

RNN की समस्याएं

- **Vanishing Gradient** समस्या के कारण लंबे इनपुट की जानकारी धीरे-धीरे गायब हो जाती है
- मॉडल ट्रेनिंग धीमी हो जाती है
- लॉन्ग टर्म डिपेंडेंसी को पकड़ना मुश्किल

LSTM का परिचय

लॉन्ग शॉर्ट टर्म मेमोरी (LSTM) एक विशेष प्रकार का RNN है, जो लंबी अवधि की जानकारी को संभालने में सक्षम है। इसमें तीन मुख्य गेट्स होते हैं:

- **Input Gate**
- **Forget Gate**
- **Output Gate**

LSTM का उपयोग

उदाहरण: वाक्य पूर्ण करना
इनपुट: "भारत का राष्ट्रीय पक्षी है..."
मॉडल का आउटपुट: "मोर"

LSTM का बेसिक कोड

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

model = Sequential()
model.add(LSTM(64, input_shape=(10, 1)))
model.add(Dense(1, activation='linear'))
```

CNN vs RNN vs LSTM

फ़ीचर	CNN	RNN	LSTM
डेटा प्रकार	इमेज	सीक्वेंस	लॉन्ग सीक्वेंस
मेमोरी	नहीं	हां	बेहतर मेमोरी
गति	तेज़	धीमी	मध्यम
प्रयोग	इमेज क्लासिफिकेशन	टेक्स्ट	टाइम सीरीज़, स्पीच

टूल्स और लाइब्रेरी

- **TensorFlow / Keras**
- **PyTorch**
- **Google Colab** (ऑनलाइन अभ्यास के लिए)
- **Matplotlib, NumPy, Pandas** (डेटा प्रोसेसिंग के लिए)

प्रैक्टिस प्रोजेक्ट सुझाव

1. बिल्ली-कुत्ते पहचानने वाला CNN मॉडल
2. LSTM आधारित वाक्य पूर्णता मॉडल
3. स्टॉक मूल्य भविष्यवाणी LSTM मॉडल द्वारा

मूल्यांकन मापदंड

- **सटीकता (Accuracy)**
- **लॉस फ़ंक्शन (Loss)**
- **कन्फ्यूज़न मैट्रिक्स**
- **RMSE / MAE** (Regression के लिए)

निष्कर्ष

CNN और LSTM आज के आधुनिक AI सिस्टम की रीढ़ हैं।
इन्हें समझना और प्रयोग करना छात्रों को AI क्षेत्र में एक मज़बूत आधार प्रदान करता है।

Chapter-10

प्राकृतिक भाषा प्रसंस्करण (Natural Language Processing - NLP): एक परिचय

प्राकृतिक भाषा प्रसंस्करण (NLP) क्या है?

प्राकृतिक भाषा प्रसंस्करण (Natural Language Processing या NLP) एक उप-क्षेत्र है कृत्रिम बुद्धिमत्ता (AI) का, जो कंप्यूटर और मानवीय भाषाओं के बीच संपर्क को सक्षम करता है। इसका उद्देश्य है — कंप्यूटर को इस योग्य बनाना कि वे मनुष्य की भाषा को समझ सकें, विश्लेषण कर सकें और प्रतिक्रिया दे सकें।

NLP क्यों जरूरी है?

- चैटबॉट्स और वर्चुअल असिस्टेंट (जैसे Alexa, Siri)
- ईमेल में स्पैम फ़िल्टरिंग
- ऑनलाइन समीक्षाओं का भाव विश्लेषण (Sentiment Analysis)
- भाषाओं का अनुवाद (Google Translate)
- वॉयस कमांड पर कार्य करना

टेक्स्ट प्रीप्रोसेसिंग क्या है?

जब हम टेक्स्ट डेटा को मशीन लर्निंग के लिए उपयोग करते हैं, तो सबसे पहले हमें उसे साफ और स्वरूपित करना होता है। इस प्रक्रिया को **Text Preprocessing** कहा जाता है।

प्रमुख चरण:

- लोअरकेस करना
- विराम चिह्न हटाना
- टोकनाइज़ेशन
- स्टॉपवर्ड्स हटाना
- स्टेमिंग और लेमाटाइज़ेशन

लोअरकेस और विराम चिह्न हटाना

उदाहरण:
"Natural Language Processing बहुत ज़बरदस्त है!"
→ lowercase: "natural language processing बहुत ज़बरदस्त है!"
→ विराम हटाने के बाद: "natural language processing बहुत ज़बरदस्त है"

टोकनाइज़ेशन (Tokenization)

यह प्रक्रिया वाक्य को शब्दों या टोकनों में तोड़ती है।

उदाहरण:
"AI ज़िन्दाबाद" → ["AI", "ज़िन्दाबाद"]

स्टॉपवर्ड्स (Stopwords) हटाना

ये ऐसे सामान्य शब्द होते हैं जो ज्यादा अर्थ नहीं रखते जैसे "है", "और", "यह", "वह"।

उदाहरण वाक्य: "यह फिल्म बहुत अच्छी है"
→ स्टॉपवर्ड्स हटाने के बाद: "फिल्म अच्छी"

स्टेमिंग और लेमाटाइज़ेशन

- **स्टेमिंग**: शब्दों की जड़ों तक ले जाने की प्रक्रिया (जैसे "चलना", "चलता" → "चल")
- **लेमाटाइज़ेशन**: सही व्याकरणिक जड़ शब्द निकालना

Python में NLTK और spaCy से किया जा सकता है।

बैग ऑफ वर्ड्स (Bag of Words - BoW)

BoW टेक्स्ट को संख्यात्मक रूप में बदलने की एक विधि है ताकि इसे ML मॉडल में इस्तेमाल किया जा सके।

चरण:

1. सभी शब्दों का एक शब्दकोश बनाएं
2. प्रत्येक वाक्य को शून्य और शब्द गिनती से युक्त एक वेक्टर में बदलें

BoW उदाहरण

वाक्य 1: "मुझे एआई पसंद है"
वाक्य 2: "एआई मुझे भी पसंद है"

शब्दकोश: ["मुझे", "एआई", "पसंद", "है", "भी"]
वेक्टर 1: [1,1,1,1,0]
वेक्टर 2: [1,1,1,1,1]

BoW की सीमाएँ

- शब्दों के क्रम को नहीं समझता
- भाव (Sentiment) को पकड़ने में असफल
- नए शब्दों से निपटना मुश्किल

फिर भी साधारण टेक्स्ट वर्गीकरण में कारगर होता है।

सेंटिमेंट एनालिसिस क्या है?

यह NLP की वह तकनीक है जिसमें किसी टेक्स्ट में भावनात्मक स्वर (positive, negative, neutral) को पहचाना जाता है।

उदाहरण:
"यह प्रोडक्ट बेहतरीन है" → Positive
"मुझे यह पसंद नहीं आया" → Negative

TextBlob से भाव विश्लेषण

Python में TextBlob का उपयोग करके सरल भाव विश्लेषण किया जा सकता है:

```
from textblob import TextBlob
text = "यह फिल्म शानदार थी"
blob = TextBlob(text)
print(blob.sentiment)
```

सेंटिमेंट क्लासिफायर कैसे बनाएं

1. डेटा संग्रह करें (समीक्षा + लेबल)
2. टेक्स्ट को साफ करें (प्रीप्रोसेसिंग)
3. फीचर बनाएं (BoW या TF-IDF)
4. मॉडल ट्रेन करें (Naive Bayes, SVM आदि से)

टेक्स्ट वर्गीकरण क्या है?

टेक्स्ट क्लासिफिकेशन एक प्रक्रिया है जिसमें किसी टेक्स्ट को एक वर्ग में डाला जाता है। जैसे:

- स्पैम या नॉट स्पैम
- समाचार श्रेणियाँ (खेल, राजनीति, मनोरंजन)
- भाव (positive, negative)

Naive Bayes से टेक्स्ट क्लासिफिकेशन

Python कोड (सरल उदाहरण):

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB

texts = ['मुझे यह पसंद है', 'यह बहुत बुरा है']
labels = ['positive', 'negative']

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(texts)

model = MultinomialNB()
model.fit(X, labels)

model.predict(vectorizer.transform(['यह शानदार है']))
```

मॉडल मूल्यांकन

- **Accuracy (शुद्धता)**
- **Precision (निपुणता)**
- **Recall (पुनःप्राप्ति)**
- **F1 स्कोर** = Precision और Recall का समुचित संतुलन

NLP के लोकप्रिय टूल्स

टूल	उपयोग
NLTK	बेसिक NLP, टोकनाइज़ेशन, स्टेमिंग
spaCy	तेज़ और कुशल NLP
TextBlob	भाव विश्लेषण, आसान इंटरफ़ेस
Scikit-learn	क्लासिफिकेशन मॉडल
HuggingFace	एडवांस NLP मॉडल जैसे BERT

अभ्यास परियोजना (Project Idea)

ट्विटर सेंटिमेंट एनालाइज़र

- ट्विटर API से ट्वीट इकट्ठा करें
- ट्वीट्स की सफाई करें
- TextBlob से विश्लेषण करें
- ग्राफ़ में परिणाम दिखाएं

NLP के वास्तविक जीवन उपयोग

- **स्वास्थ्य**: मेडिकल रिकॉर्ड्स का विश्लेषण
- **वित्त**: धोखाधड़ी पहचान
- **ग्राहक सेवा**: चैटबॉट्स
- **शिक्षा**: उत्तर स्वतः जाँच प्रणाली

निष्कर्ष और अगला कदम

- NLP कंप्यूटर को मानव भाषा समझने की शक्ति देता है
- टेक्स्ट प्रीप्रोसेसिंग एक महत्वपूर्ण पहला चरण है
- BoW, सेंटिमेंट एनालिसिस, और क्लासिफिकेशन शुरुआती के लिए उपयोगी हैं

Chapter-11

AI मॉडल डिप्लॉयमेंट

विषय:

- मॉडल सेव करना (Pickle / Joblib)
- Flask और Streamlit से डिप्लॉय करना

AI मॉडल डिप्लॉयमेंट क्या है?

जब भी हम कोई मशीन लर्निंग मॉडल तैयार करते हैं, तो उसे इस्तेमाल करने योग्य बनाना बहुत जरूरी होता है। यही प्रक्रिया **डिप्लॉयमेंट** कहलाती है। डिप्लॉयमेंट का मतलब है – अपने मॉडल को किसी एप या वेबसाइट के रूप में उपलब्ध कराना ताकि अन्य लोग उसका उपयोग कर सकें।

डिप्लॉयमेंट क्यों जरूरी है?

- मॉडल को वास्तविक जीवन में इस्तेमाल करने योग्य बनाता है
- व्यवसायिक एप्लिकेशन में एकीकरण की अनुमति देता है
- लाइव प्रेडिक्शन संभव होता है
- प्रोजेक्ट को एक संपूर्ण प्रोडक्ट में बदलता है

मॉडल सेव करना – परिचय

एक बार जब मॉडल ट्रेन हो जाए, तो उसे सेव (save) करना जरूरी होता है ताकि हर बार दोबारा ट्रेन न करना पड़े।
इसके लिए हम आमतौर पर दो लाइब्रेरी का उपयोग करते हैं:

- **Pickle**
- **Joblib**

Pickle से मॉडल सेव करना

```
import pickle

# मॉडल सेव करना
with open('model.pkl', 'wb') as file:
    pickle.dump(model, file)

# मॉडल लोड करना
with open('model.pkl', 'rb') as file:
```

```
loaded_model = pickle.load(file)
```

Pickle सभी छोटे से मध्यम साइज के मॉडल के लिए उपयुक्त है।

Joblib से मॉडल सेव करना

```
from joblib import dump, load

# मॉडल सेव करना
dump(model, 'model.joblib')

# मॉडल लोड करना
model = load('model.joblib')
```

Joblib खासतौर पर बड़े और numpy आधारित डेटा के लिए तेज और कुशल होती है।

Flask और Streamlit का परिचय

डिप्लॉयमेंट के लिए Python में दो लोकप्रिय विकल्प हैं:

Framework	उपयोग	आसानाई	उपयोगकर्ता इंटरफेस
Flask	APIs के लिए	अधिक नियंत्रण	HTML/CSS की जरूरत
Streamlit	डैशबोर्ड के लिए	बहुत आसान	ऑटोमैटिक UI

Flask प्रोजेक्ट का ढांचा

```
project/
  ├── model.pkl
  ├── app.py
  ├── templates/
      └── index.html
```

Flask एप का बेसिक कोड

```
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
```

```
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    val = float(request.form['feature'])
    output = model.predict([[val]])
    return render_template('index.html', prediction=output)

if __name__ == '__main__':
    app.run(debug=True)
```

HTML फॉर्म (templates/index.html)

```
<form action="/predict" method="post">
  <input type="text" name="feature" placeholder="मूल्य दर्ज करें">
  <input type="submit" value="प्रेडिक्ट करें">
</form>

{% if prediction %}
  <h3>Prediction: {{ prediction[0] }}</h3>
{% endif %}
```

Flask एप चलाना

```
python app.py
```

इसके बाद ब्राउज़र में खोलें: `http://127.0.0.1:5000`

Streamlit का परिचय

Streamlit एक बेहद आसान Python फ्रेमवर्क है जिससे आप सिर्फ कोड लिखकर सुंदर वेब एप बना सकते हैं।

उदाहरण: डेटा डैशबोर्ड, ML डेमो, प्रोटोटाइप आदि।

Streamlit इंस्टॉल व रन करें

```
pip install streamlit
streamlit run app.py
```

Streamlit एप का बेसिक कोड

```
import streamlit as st
import pickle

model = pickle.load(open('model.pkl', 'rb'))

st.title('मशीन लर्निंग मॉडल')
val = st.number_input('इनपुट दर्ज करें')

if st.button('प्रेडिक्ट'):
    result = model.predict([[val]])
    st.success(f'Prediction: {result[0]}')
```

Streamlit Widgets

Widget	उपयोग
st.text_input	टेक्स्ट इनपुट
st.button	बटन क्लिक
st.success	परिणाम दिखाना
st.number_input	नंबर इनपुट

मॉडल को शेयर करना (Streamlit Cloud)

1. GitHub पर कोड अपलोड करें
2. https://share.streamlit.io पर जाएं
3. GitHub रिपोज़िटरी लिंक करें
4. App ऑटोमेटिकली लाइव हो जाएगा

आम समस्याएं

- मॉडल लोडिंग में समस्या
- requirements.txt फाइल नहीं है
- Input का फॉर्मेट गलत
- Python वर्ज़न mismatch

---

requirements.txt का उदाहरण

```
streamlit
scikit-learn
pandas
```

```
numpy
joblib

pip freeze > requirements.txt
```

मॉडल टेस्टिंग के तरीके

- छोटे डेटा पर टेस्ट करें
- Edge case पर ट्राय करें
- गलत इनपुट देकर देखें मॉडल कैसे behave करता है

प्रैक्टिस के लिए विचार

- प्राइस प्रेडिक्शन मॉडल बनाएं
- स्पैम डिटेक्टर बनाएं
- इमेज क्लासिफिकेशन मॉडल को Streamlit से deploy करें

निष्कर्ष

- मॉडल को सेव करना एक आवश्यक कदम है
- Flask APIs के लिए अच्छा है
- Streamlit डैशबोर्ड और प्रेजेंटेशन के लिए श्रेष्ठ है
- पहले लोकल पर अभ्यास करें, फिर क्लाउड पर तैनात करें

GOVT OF INDIA CERTIFIED SARVA EDUCATION ISO 9001:2015 CERTIFIED Start New Computer Centre Sarvaindia.com Join TODAY

Chapter-12

GOVT OF INDIA CERTIFIED SARVA EDUCATION ISO 9001:2015 CERTIFIED Start New Computer Centre Sarvaindia.com Join TODAY

फाइनल प्रोजेक्ट – एक संपूर्ण मार्गदर्शिका

भूमिका

फाइनल प्रोजेक्ट इस एआई कोर्स का अंतिम और सबसे महत्वपूर्ण हिस्सा है। यह आपको अपने ज्ञान को एक वास्तविक समस्या पर लागू करने का अवसर देता है।

उद्देश्य:

- अपने सीखे हुए कौशल को व्यावहारिक रूप में लागू करना
- एक समस्या को एआई के माध्यम से हल करना
- प्रेजेंटेशन के द्वारा प्रोजेक्ट का प्रदर्शन करना

प्रोजेक्ट श्रेणियाँ

आप निम्न में से किसी एक श्रेणी में प्रोजेक्ट चुन सकते हैं:

1. **टेक्स्ट आधारित प्रोजेक्ट (NLP)**
2. **इमेज आधारित प्रोजेक्ट (Computer Vision)**
3. **डेटा आधारित प्रोजेक्ट (Machine Learning)**

टेक्स्ट आधारित प्रोजेक्ट आइडियाज़

- **स्पैम ईमेल क्लासिफायर**
- **सेंटिमेंट एनालिसिस (जैसे मूवी रिव्यू)**
- **बेसिक चैटबॉट**

उपयोगी टूल्स: `NLTK`, `Scikit-learn`, `HuggingFace`, `spaCy`

इमेज आधारित प्रोजेक्ट आइडियाज़

- **हस्तलिखित अंकों की पहचान (MNIST)**
- **फेस मास्क डिटेक्शन**
- **कुत्ता बनाम बिल्ली छवि पहचान**

उपयोगी टूल्स: `OpenCV`, `TensorFlow`, `Keras`

डेटा आधारित प्रोजेक्ट आइडियाज़

- **हाउस प्राइस प्रिडिक्शन**
- **लोन अप्रूवल मॉडल**
- **ग्राहक छोड़ने की भविष्यवाणी (Customer Churn)**

उपयोगी टूल्स: `Pandas`, `Scikit-learn`, `Matplotlib`

प्रोजेक्ट के मुख्य घटक

1. समस्या विवरण (Problem Statement)
2. डेटा संग्रहण
3. डेटा प्रोसेसिंग और सफाई
4. मॉडल का चयन
5. ट्रेनिंग और टेस्टिंग
6. परिणामों का मूल्यांकन
7. रिपोर्ट और प्रेजेंटेशन

समस्या को परिभाषित करना

प्रारूप:

- **समस्या:** आप क्या हल करना चाहते हैं?
- **इनपुट:** कौन-से डेटा का प्रयोग करेंगे?
- **आउटपुट:** मॉडल क्या परिणाम देगा?

डेटा संग्रहण

स्रोत:

- **Kaggle Datasets**
- **UCI Machine Learning Repository**
- **CSV फाइल्स या API डेटा**

डेटा प्रोसेसिंग

- Null वैल्यू हटाना
- स्केलिंग / नॉर्मलाइजेशन

- टेक्स्ट क्लीनिंग (NLP में)
- इमेज रीसाइजिंग (CV में)

मॉडल ट्रेनिंग

उदाहरण:

- **Random Forest, SVM** डेटा के लिए
- **CNN** इमेज के लिए
- **Logistic Regression** टेक्स्ट के लिए

ट्रेनिंग व टेस्टिंग विभाजन (80:20)

मूल्यांकन मानदंड

- Accuracy
- Confusion Matrix
- Precision, Recall, F1-score
- Visualization के माध्यम से प्रस्तुति

उदाहरण प्रोजेक्ट – सेंटिमेंट एनालिसिस

- **डेटा:** मूवी रिव्यू (IMDB)
- **मॉडल:** Logistic Regression
- **आउटपुट:** सकारात्मक / नकारात्मक भावना

उदाहरण प्रोजेक्ट – इमेज क्लासिफिकेशन

- **डेटा:** डॉग बनाम कैट इमेजेज
- **मॉडल:** CNN
- **आउटपुट:** कुत्ता या बिल्ली

उदाहरण प्रोजेक्ट – हाउस प्राइस प्रिडिक्शन

- **डेटा:** हाउसिंग.csv
- **मॉडल:** Linear Regression

- **आउटपुट:** मूल्य का अनुमान

आवश्यक टूल्स और लाइब्रेरी

टूल/लाइब्रेरी	कार्य
Pandas	डेटा प्रोसेसिंग
NumPy	न्यूमेरिक गणना
Scikit-learn	ML मॉडल
Keras / TensorFlow	डीप लर्निंग
OpenCV	इमेज प्रोसेसिंग
Streamlit/Flask	डिप्लॉयमेंट

प्रेजेंटेशन के दिशा-निर्देश

- प्रॉब्लम स्टेटमेंट
- डेटा स्रोत और प्रोसेसिंग
- मॉडल विवरण
- परिणामों का विश्लेषण (ग्राफ/टेबल सहित)
- लाइव डेमो या स्क्रीनशॉट

समय सीमा: 5–10 मिनट

रिपोर्ट सबमिशन प्रारूप

रिपोर्ट में शामिल करें:

- शीर्षक और विवरण
- उपयोग किए गए टूल्स
- मॉडल आर्किटेक्चर
- कोड स्निपेट्स
- परिणामों की व्याख्या
- निष्कर्ष और सुझाव

मूल्यांकन मानदंड (100 अंक)

घटक	अंक
समस्या परिभाषा	10
डेटा प्रोसेसिंग	10

मॉडल चयन	15
सटीकता/परिणाम	20
प्रेजेंटेशन	15
नवीनता	10
रिपोर्ट गुणवत्ता	10
टीम सहयोग	10
कुल	**100**

निष्कर्ष

फाइनल प्रोजेक्ट आपके लिए एक सुनहरा अवसर है अपनी प्रतिभा दिखाने का। ध्यानपूर्वक विषय का चयन करें, मेहनत करें, और एक प्रभावी प्रेजेंटेशन के साथ अपने कोर्स की सफलता को दर्शाएं।

Chapter-13

उन्नत आर्टिफिशियल इंटेलिजेंस (AI) की अवधारणाएं

(रीइन्फोर्समेंट लर्निंग, एडवांस डीप लर्निंग, ऑब्जेक्ट डिटेक्शन)

रिइन्फोर्समेंट लर्निंग का परिचय

रिइन्फोर्समेंट लर्निंग (Reinforcement Learning – RL) एक ऐसा मशीन लर्निंग तरीका है जिसमें एक एजेंट (Agent) किसी पर्यावरण (Environment) के साथ इंटरैक्शन करके सीखता है कि कौन-सा कार्य (Action) किस स्थिति (State) में करने से सबसे ज़्यादा इनाम (Reward) मिलेगा।

मुख्य घटक:

- **एजेंट (Agent):** निर्णय लेने वाला
- **एनवायरनमेंट (Environment):** जहाँ एजेंट कार्य करता है
- **स्टेट (State):** किसी समय पर एजेंट की स्थिति
- **एक्शन (Action):** एजेंट द्वारा लिया गया निर्णय
- **रिवॉर्ड (Reward):** एनवायरनमेंट से मिलने वाला पॉज़िटिव या नेगेटिव फ़ीडबैक

उदाहरण:

मान लीजिए एक रोबोट चलना सीख रहा है। हर बार चलने पर +1 अंक और गिरने पर -1 अंक मिलता है। धीरे-धीरे रोबोट वो तरीका सीख जाता है जिससे वह गिरने से बचकर चल सके।

रिइन्फोर्समेंट लर्निंग के प्रकार और एल्गोरिदम

Q-Learning क्या है?

Q-learning एक प्रकार का Model-Free Reinforcement Learning एल्गोरिदम है जिसमें एजेंट विभिन्न स्टेट्स और एक्शन्स के लिए Q-Values को अपडेट करता है:

फॉर्मूला:

$$Q(s, a) = Q(s, a) + \alpha [r + \gamma \max Q(s', a') - Q(s, a)]$$

जहाँ,

- s = वर्तमान स्टेट
- a = वर्तमान एक्शन
- r = रिवॉर्ड
- s' = अगली स्टेट

- α = लर्निंग रेट
- γ = डिस्काउंट फैक्टर

एडवांस डीप लर्निंग की झलक

डीप लर्निंग की उन्नत तकनीकों में बड़े और अधिक सटीक मॉडल शामिल हैं जो जटिल समस्याओं को हल कर सकते हैं।

मुख्य विषय:

- ट्रांसफर लर्निंग (Transfer Learning)
- रेसनेट (ResNet – Residual Networks)
- इंसेप्शन नेटवर्क (Inception Network)

ट्रांसफर लर्निंग

परिभाषा:
ट्रांसफर लर्निंग का मतलब है किसी पहले से प्रशिक्षित मॉडल को नए, लेकिन समान प्रकार के कार्यों के लिए दोबारा उपयोग करना।

फ़ायदे:

- समय और संसाधनों की बचत
- छोटे डाटा सेट पर भी अच्छा परफॉर्मेंस
- मौजूदा मॉडल्स जैसे VGG, Inception, ResNet का इस्तेमाल

उदाहरण:
ImageNet पर प्रशिक्षित ResNet को मेडिकल इमेज क्लासिफिकेशन के लिए रीट्रेन करना।

ResNet – Residual Networks

समस्या:
गहरे न्यूरल नेटवर्क में Gradient Vanishing की समस्या आती है।

समाधान:
ResNet में **Residual Blocks** होते हैं जो "Skip Connections" बनाते हैं। इससे मॉडल पहले सीखी गई जानकारी को भूलता नहीं है।

विशेषता:

- ResNet18, ResNet50 जैसे मॉडल
- `F(x) + x` फॉर्मेट की Residual संरचना

Inception Network

लक्ष्य:
एक ही लेयर पर कई साइज के फ़िल्टर का उपयोग कर अधिक जानकारी निकालना।

मॉड्यूल में शामिल:

- 1x1, 3x3, 5x5 Convolutions
- Max Pooling

वर्शन:

- Inception v1 (GoogLeNet), v3, v4

उपयोग:

- बड़े डाटा सेट पर इमेज क्लासिफिकेशन
- हाई परफॉर्मेंस टास्क

ऑब्जेक्ट डिटेक्शन का परिचय

ऑब्जेक्ट डिटेक्शन का मतलब होता है छवि में वस्तु की पहचान करना **और** उसका स्थान (Bounding Box) बताना।

दो मुख्य काम:

1. क्या है? → क्लासिफिकेशन
2. कहाँ है? → लोकलाइज़ेशन

YOLO (You Only Look Once)

YOLO एक प्रसिद्ध रीयल-टाइम ऑब्जेक्ट डिटेक्शन एल्गोरिदम है।

YOLO की विशेषताएं:

- एक ही नेटवर्क पूरा इमेज प्रोसेस करता है
- तेज़ और कुशल (Real-Time Friendly)

- ग्रिड आधारित आउटपुट

YOLO क्या देता है?

- Bounding Boxes
- Confidence Score
- Class Probabilities

वर्ज़न:
YOLOv3, v4, v5, v8 (PyTorch आधारित)

Haar Cascade Classifier

Haar Cascades एक पुरानी लेकिन तेज़ ऑब्जेक्ट डिटेक्शन तकनीक है, जिसे विशेष रूप से **चेहरे की पहचान** के लिए इस्तेमाल किया जाता है।

कैसे काम करता है?

- Haar Features (जैसे किनारा, रेखाएं) निकालता है
- कई कमजोर Classifiers मिलकर एक मजबूत Classifier बनाते हैं
- Cascade में कई चरण होते हैं – तेज़ पहचान

उपयोग:

- फेस डिटेक्शन
- OCR, बेसिक डिटेक्शन टास्क

तुलना – YOLO vs Haar Cascade

विशेषता	YOLO	Haar Cascade
स्पीड	बहुत तेज़	तेज़
सटीकता	उच्च	मध्यम
एप्लीकेशन	ऑब्जेक्ट्स, वाहन	चेहरा, सरल वस्तुएं
डीप लर्निंग	हाँ	नहीं

उपकरण (Tools & Libraries)

- **TensorFlow/Keras** – डीप लर्निंग मॉडल के लिए
- **OpenCV** – इमेज प्रोसेसिंग और Haar
- **PyTorch** – YOLO मॉडल
- **LabelImg** – इमेज लेबलिंग टूल

- **Flask/Streamlit** – डिप्लॉयमेंट
- **Google Colab** – अभ्यास के लिए मुफ़्त GPU प्लेटफ़ॉर्म

प्रैक्टिकल प्रोजेक्ट सुझाव

1. **चेहरे की पहचान ऐप** (Haar Cascade)
2. **रियल-टाइम वस्तु पहचान ऐप** (YOLOv5)
3. **ट्रांसफर लर्निंग पर आधारित इमेज क्लासिफायर**
4. **संकेत भाषा पहचान प्रणाली**

मूल्यांकन और अभ्यास प्रश्न

1. Q-Learning और Supervised Learning में क्या अंतर है?
2. ResNet में Skip Connection का क्या लाभ है?
3. YOLO और R-CNN में कौन तेज़ है और क्यों?
4. Transfer Learning का लाभ कब उठाना चाहिए?
5. Haar Cascade किन कार्यों के लिए उपयुक्त है?

निष्कर्ष

रिइन्फोर्समेंट लर्निंग, ट्रांसफर लर्निंग, और ऑब्जेक्ट डिटेक्शन आधुनिक एआई के सबसे प्रभावशाली और व्यावहारिक क्षेत्रों में से हैं। इनकी समझ और अभ्यास से छात्र वास्तविक दुनिया की समस्याओं को हल कर सकते हैं।

Chapter-14

एडवांस्ड NLP और चैटबॉट विकास

विषय:

- ट्रांसफॉर्मर, BERT और GPT मॉडल की जानकारी
- रीयल-वर्ल्ड डेटा सेट्स (Kaggle, UCI)
- चैटबॉट डेवलपमेंट (Rasa/Dialogflow)

अनुक्रमणिका

1. NLP का परिचय
2. ट्रांसफॉर्मर मॉडल क्या है?
3. BERT मॉडल का अवलोकन
4. GPT मॉडल और उसके संस्करण
5. BERT और GPT की तुलना
6. वास्तविक दुनिया के NLP डेटा सेट्स
7. डेटा प्रीप्रोसेसिंग (पूर्व-संसाधन)
8. टोकनाइज़ेशन, पैडिंग और अटेंशन
9. फाइन-ट्यूनिंग और ट्रांसफॉर्मर का उपयोग
10. चैटबॉट क्या है?
11. नियम आधारित बनाम AI आधारित चैटबॉट
12. Rasa से चैटबॉट निर्माण
13. Dialogflow से चैटबॉट निर्माण
14. चैटबॉट के केस स्टडी
15. प्रदर्शन और मूल्यांकन
16. चैटबॉट को वेब पर होस्ट करना
17. NLP में करियर विकल्प
18. उपयोगी उपकरण और लाइब्रेरी
19. अभ्यास प्रश्न
20. समापन और आगे की राह

---

1. NLP का परिचय

प्राकृतिक भाषा प्रसंस्करण (Natural Language Processing) कंप्यूटर को इंसानी भाषा समझने और उसका विश्लेषण करने में सक्षम बनाता है। इसकी मदद से मशीनें टेक्स्ट, वॉयस और चैट को समझ सकती हैं।

2. ट्रांसफॉर्मर मॉडल क्या है?

ट्रांसफॉर्मर मॉडल एक Deep Learning आर्किटेक्चर है जो 2017 में "Attention is All You Need" पेपर द्वारा प्रस्तुत किया गया था। इसकी विशेषता है **Self-Attention**, जो वाक्य में सभी शब्दों के बीच संबंध को एक साथ समझने में मदद करता है।

---

3. BERT का अवलोकन

BERT (Bidirectional Encoder Representations from Transformers) गूगल द्वारा विकसित किया गया है जो शब्दों को दोनो दिशाओं से (बाएं और दाएं) पढ़ सकता है।

विशेषताएँ:

- Pre-trained मॉडल
- Question Answering, Sentiment Analysis में उपयोगी

उदाहरण:
वाक्य: "मैं बैंक के पास बैठा हूँ।"
BERT यह समझता है कि "बैंक" नदी का किनारा है या वित्तीय संस्था।

---

4. GPT मॉडल

GPT (Generative Pre-trained Transformer) मॉडल OpenAI द्वारा विकसित एक text generation मॉडल है।

संस्करण:

- GPT-1, GPT-2, GPT-3, GPT-4
- GPT-3 और GPT-4 में बहुत शक्तिशाली भाषा समझने की क्षमता है

उदाहरण: GPT से कविता, लेख, ईमेल, कोड आदि बनवाया जा सकता है।

---

5. BERT vs GPT तुलना

विशेषता	BERT	GPT
दिशा	Bidirectional	Unidirectional (Left-to-Right)
उद्देश्य	टेक्स्ट को समझना	टेक्स्ट जनरेट करना
एप्लीकेशन	Classifier, QA	लेखन, चैटबॉट, कोडिंग

---

6. वास्तविक डेटा सेट्स

स्रोत:

- **Kaggle**: ग्राहक समीक्षा, ट्विटर एनालिसिस
- **UCI Repository**: SMS स्पैम, ईमेल डाटा
- **HuggingFace Datasets**: IMDb, AG News आदि

इनका उपयोग मॉडल को ट्रेन और परीक्षण करने के लिए किया जाता है।

---

7. डेटा प्रीप्रोसेसिंग

NLP में उच्च गुणवत्ता वाले डेटा की आवश्यकता होती है:

- Lowercasing
- Stop Words हटाना
- टोकनाइज़ेशन
- लेमेटाइजेशन/स्टेमिंग
- वर्ड एंबेडिंग: Word2Vec, GloVe, BERT Embeddings

---

8. टोकनाइज़ेशन और अटेंशन

- **टोकनाइज़ेशन**: शब्दों को संख्याओं में बदलना
- **Padding**: एक ही लंबाई का इनपुट बनाना
- **Attention Mechanism**: शब्दों के बीच प्राथमिकता तय करना

---

9. फाइन-ट्यूनिंग

आप BERT या GPT को किसी विशेष कार्य के लिए फाइन-ट्यून कर सकते हैं जैसे कि:

- ट्विटर पर हेट स्पीच डिटेक्शन
- कस्टमर सपोर्ट चैटबॉट

लाइब्रेरी: HuggingFace Transformers, PyTorch, TensorFlow

---

10. चैटबॉट क्या है?

चैटबॉट एक प्रोग्राम होता है जो मानव भाषा में बातचीत करता है।
प्रकार:

- नियम आधारित (Rule-Based)
- AI आधारित (Intent + Context)

---

11. नियम आधारित बनाम AI आधारित चैटबॉट

प्रकार	नियम आधारित	AI आधारित
समझ क्षमता	सीमित	उच्च
टेक्नोलॉजी	Regex, लोजिक	NLP, ML, DL
उदाहरण	मेनू बॉट	हेल्पडेस्क असिस्टेंट

---

12. Rasa से चैटबॉट निर्माण

Rasa एक ओपन-सोर्स फ्रेमवर्क है:

- Rasa NLU: इंटेंट पहचान
- Rasa Core: बातचीत की योजना
- Actions: यूज़र के इनपुट पर कार्य करना

इंस्टालेशन:

```
pip install rasa
```

---

13. Dialogflow से चैटबॉट निर्माण

Dialogflow गूगल द्वारा विकसित एक क्लाउड-आधारित टूल है।

विशेषताएं:

- इंटरफेस: GUI आधारित
- Intents, Entities, Fulfillment
- वॉयस इनपुट भी सपोर्ट करता है

---

14. चैटबॉट केस स्टडी

उदाहरण 1: मानसिक स्वास्थ्य सहायक

- इंटेंट पहचान
- भावनात्मक विश्लेषण
- हेल्पलाइन नंबर सुझाव

उदाहरण 2: ग्राहक सेवा चैटबॉट

- ऑर्डर ट्रैकिंग
- रिफंड नीति
- लाइव एजेंट कनेक्शन

---

15. प्रदर्शन और मूल्यांकन

- इंटेंट की सटीकता
- यूज़र संतुष्टि स्कोर
- प्रतिक्रिया समय
- चैट लॉग विश्लेषण

---

16. चैटबॉट होस्टिंग

- **Streamlit** या **Gradio** पर इंटरफ़ेस
- **Flask API** से बैकएंड
- Telegram या WhatsApp बॉट के रूप में लाइव करें

---

17. NLP में करियर विकल्प

- NLP इंजीनियर
- चैटबॉट डेवलपर
- डेटा वैज्ञानिक (टेक्स्ट आधारित)
- कंटेंट मॉडरेशन विशेषज्ञ

---

18. उपयोगी उपकरण और लाइब्रेरी

- HuggingFace Transformers
- Rasa NLU/Core
- SpaCy
- NLTK
- Dialogflow

19. अभ्यास प्रश्न

1. ट्रांसफॉर्मर मॉडल का मुख्य घटक क्या है?
2. BERT और GPT में क्या अंतर है?
3. HuggingFace क्या है?
4. चैटबॉट बनाने के दो प्लेटफॉर्म कौन-कौन से हैं?

---

20. समापन और आगे की राह

एडवांस NLP ने भाषा की समझ और टेक्स्ट जनरेशन को नई ऊँचाइयों तक पहुँचाया है।
आगे आप निम्नलिखित में गहराई से जा सकते हैं:

- GPT Fine-tuning
- Dialogue Management Systems
- Advanced Chatbot Analytics

Chapter-15

AI Project Lifecycle & MLOps Basics

विषयवस्तु:

1. AI प्रोजेक्ट लाइफसायकल
2. MLOps का परिचय
3. मॉडल को क्लाउड पर डिप्लॉय करना (Heroku, AWS, GCP)
4. AI में नैतिकता, पक्षपात (Bias) और निष्पक्षता (Fairness)
5. Capstone प्रोजेक्ट और Viva की तैयारी

1: AI प्रोजेक्ट का जीवनचक्र (AI Project Lifecycle)

AI प्रोजेक्ट 6 मुख्य चरणों में पूरे होते हैं:

1. **Problem Definition**
   - समस्या को स्पष्ट रूप से परिभाषित करना
   - उद्देश्य तय करना
2. **Data Collection**
   - डेटा स्रोत: API, CSV, डेटाबेस, वेब स्क्रैपिंग
   - डेटा की मात्रा और गुणवत्ता
3. **Data Preparation & Cleaning**
   - Missing values हटाना
   - Feature selection & normalization
4. **Model Selection & Training**
   - एल्गोरिदम: Linear Regression, Decision Tree, etc.
   - Evaluation: Accuracy, Precision, Recall
5. **Model Evaluation**
   - Cross-validation
   - Confusion Matrix, ROC-AUC
6. **Deployment & Monitoring**
   - Flask, Streamlit, FastAPI
   - Post-deployment performance checks

2: MLOps का परिचय

MLOps (Machine Learning Operations) DevOps + ML का मेल है। यह मॉडल को उत्पादन स्तर तक पहुँचाने में सहायक होता है।

मुख्य घटक:

- **Continuous Integration (CI)**
- **Continuous Deployment (CD)**
- **Version Control (Git, DVC)**
- **Model Tracking (MLflow, Weights & Biases)**
- **Automated Testing**

MLOps सुनिश्चित करता है कि ML मॉडल बार-बार बिना त्रुटि के डिप्लॉय और अपडेट हो सके।

3: मॉडल को क्लाउड पर डिप्लॉय करना

A. Heroku पर डिप्लॉयमेंट

- Flask ऐप बनाएँ
- `requirements.txt` और `Procfile` बनाएँ
- Heroku CLI से GitHub रिपोजिटरी को जोड़ें
- `heroku push master` कमांड से डिप्लॉय करें

B. AWS (Amazon Web Services)

- EC2 इंस्टेंस बनाएं
- Flask/Streamlit ऐप को इंस्टॉल करें
- Webserver (nginx/gunicorn) से होस्ट करें
- S3 और Lambda जैसे टूल्स से इंटीग्रेशन संभव

C. GCP (Google Cloud Platform)

- Google App Engine या Vertex AI
- Storage और Compute विकल्प
- स्केलेबिलिटी और सिक्योरिटी फीचर्स

4: AI में नैतिकता, पक्षपात और निष्पक्षता

A. नैतिक चिंताएँ:

- Surveillance और Privacy उल्लंघन
- Fake content (Deepfakes)
- Automated decision-making के दुष्परिणाम

B. Bias (पक्षपात)

AI मॉडल अक्सर training डेटा से सीखते हैं। यदि डेटा biased है, तो निर्णय भी biased होंगे।

उदाहरण:

- Gender bias in resume filtering
- Racism in facial recognition

C. Fairness तकनीक:

- Data balancing techniques
- Explainable AI (XAI)
- Fairness metrics: Equal Opportunity, Disparate Impact

5: Capstone Project

Capstone प्रोजेक्ट आपके ज्ञान और कौशल का प्रदर्शन है। यह अंत में Viva के साथ मूल्यांकित किया जाता है।

उदाहरण प्रोजेक्ट विषय:

1. **स्पैम डिटेक्टर** – NLP आधारित टेक्स्ट क्लासिफिकेशन
2. **हाउस प्राइस प्रेडिक्शन** – Regression मॉडल
3. **चैटबॉट डेवलपमेंट** – Intent Recognition और Dialog Flow
4. **इमेज क्लासिफिकेशन** – CNN आधारित मॉडल

Project Documentation:

- Problem Statement
- Data Source & Cleaning
- Model Description
- Accuracy & Metrics
- Screenshots और कोड snippets
- Deployment लिंक

6: Viva की तैयारी

Viva में पूछे जाने वाले संभावित प्रश्न:

1. आपने कौन सा एल्गोरिदम और क्यों चुना?
2. Model accuracy कैसे improve किया?
3. आप Bias को कैसे कम करेंगे?
4. आपने कौन से टूल/लाइब्रेरी का उपयोग किया?
5. आपने मॉडल को कहाँ और कैसे होस्ट किया?

7: भविष्य की दिशा और करियर मार्ग

MLOps और Ethical AI का ज्ञान भविष्य में निम्नलिखित जॉब्स में मदद करेगा:

- AI Product Engineer
- MLOps Specialist
- Cloud ML Developer
- Responsible AI Researcher

अभ्यास प्रश्न

1. AI Project Lifecycle के 6 चरण कौन से हैं?
2. MLOps में "CI/CD" का क्या महत्व है?
3. Heroku और AWS में क्या अंतर है डिप्लॉयमेंट को लेकर?
4. Bias को कम करने की तीन तकनीकें क्या हैं?
5. Capstone प्रोजेक्ट में क्या-क्या शामिल किया जाता है?

निष्कर्ष

AI केवल मॉडल बनाने तक सीमित नहीं है — सही डेटा, नैतिक निर्णय, और स्थिर डिप्लॉयमेंट सभी एक सफल AI सिस्टम का हिस्सा हैं। MLOps और नैतिकता को समझना आज के AI इंजीनियर के लिए अनिवार्य है।

Chapter-16

एडवांस्ड स्पेशलाइजेशन ट्रैक्स इन एआई

(AI की गहराई में उतरें: Computer Vision, NLP, Robotics, और Business AI)

अनुक्रमणिका

1. परिचय
2. कंप्यूटर विज़न
   - OpenCV
   - CNNs
   - ऑब्जेक्ट डिटेक्शन
   - GANs
3. NLP (नेचुरल लैंग्वेज प्रोसेसिंग)
   - Transformers
   - BERT
   - GPT
   - LangChain
4. रोबोटिक्स + AI
   - सिमुलेशन
   - पाथफाइंडिंग
   - रिइंफोर्समेंट लर्निंग
5. बिज़नेस में AI
   - डेटा साइंस + AI
   - निर्णय प्रणाली
6. निष्कर्ष

---

1 परिचय

आज के समय में आर्टिफिशियल इंटेलिजेंस केवल एक सामान्य विषय नहीं रहा। अब यह कई क्षेत्रों में विशेष रूप से उपयोग किया जा रहा है। इस पुस्तक में हम 4 महत्वपूर्ण विशेषज्ञता क्षेत्रों पर ध्यान देंगे:

- **कंप्यूटर विज़न**: छवि और वीडियो से समझ निकालना
- **NLP**: भाषा को समझना और संसाधित करना
- **रोबोटिक्स + AI**: स्मार्ट मशीनें जो निर्णय लेती हैं
- **बिज़नेस AI**: डेटा-संचालित निर्णय प्रणाली

---

2 कंप्यूटर विज़न

OpenCV

OpenCV एक ओपन-सोर्स लाइब्रेरी है जो इमेज प्रोसेसिंग में प्रयोग होती है।

- मुख्य उपयोग: फेस डिटेक्शन, ऑब्जेक्ट ट्रैकिंग, इमेज फिल्टर
- उदाहरण:

```
import cv2
img = cv2.imread('image.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow("Gray", gray)
```

CNNs (कन्वोल्यूशन न्यूरल नेटवर्क्स)

CNNs इमेज क्लासिफिकेशन के लिए बनाए गए विशेष नेटवर्क होते हैं।

- उपयोग: पहचान, मेडिकल इमेज एनालिसिस, ऑटोमेटेड व्हीकल्स
- लेयर: Convolution → Pooling → Dense

ऑब्जेक्ट डिटेक्शन

- एक इमेज में कई ऑब्जेक्ट को पहचानना
- तकनीक:
  - YOLO
  - Haar Cascades
  - SSD

GANs (जनरेटिव एडवर्सेरियल नेटवर्क्स)

- दो नेटवर्क: Generator और Discriminator
- उपयोग: फोटो जनरेशन, डीपफेक्स, कला निर्माण

---

3 NLP (नेचुरल लैंग्वेज प्रोसेसिंग)

भाषा को समझना और उस पर काम करना NLP का काम है।

Transformers

- NLP को क्रांतिकारी रूप देने वाली संरचना
- ध्यान केंद्रित (Attention) प्रणाली पर आधारित
- समानांतर रूप से शब्दों को संसाधित करता है

BERT

- Google द्वारा विकसित

- बायडायरेक्शनल मॉडल जो शब्दों के संदर्भ को समझता है
- उपयोग: भावना विश्लेषण, प्रश्न-उत्तर प्रणाली

GPT

- OpenAI द्वारा विकसित
- GPT-3 और GPT-4 अत्यधिक प्रभावशाली मॉडल हैं
- उपयोग: टेक्स्ट जनरेशन, अनुवाद, कोड लेखन

LangChain

- एक फ्रेमवर्क जो LLMs को टूल्स से जोड़ता है
- उदाहरण: GPT + डेटाबेस + API = इंटेलिजेंट चैटबॉट्स
- उपयोग: एजेंट्स, चैटबॉट्स, डाटा सर्च

---

4 रोबोटिक्स + AI

AI जब रोबोटिक्स से जुड़ता है, तो स्मार्ट मशीनें बनती हैं जो निर्णय लेती हैं और गतिशील होती हैं।

सिमुलेशन

- Gazebo, Unity, Webots जैसे सॉफ्टवेयर में रोबोट को वर्चुअल रूप से टेस्ट किया जाता है
- हकीकत में प्रयोग से पहले मॉडल तैयार करना

पाथफाइंडिंग

- लक्ष्य तक पहुंचने के लिए रास्ता ढूंढने के एल्गोरिदम
- प्रमुख एल्गोरिदम:
  - A*
  - Dijkstra
  - RRT

रिइंफोर्समेंट लर्निंग

- AI मॉडल वातावरण से सीखता है
- प्रत्येक क्रिया पर इनाम या सज़ा मिलती है
- उपयोग: गेम्स, सेल्फ ड्राइविंग, इंडस्ट्रियल रोबोट्स

---

5 AI इन बिज़नेस

AI अब व्यापार निर्णयों को तेज और सटीक बना रहा है।

डेटा साइंस + AI

- डाटा से पैटर्न ढूंढना और पूर्वानुमान लगाना
- टूल्स: Pandas, Scikit-learn, Power BI
- उपयोग: ग्राहक विश्लेषण, बिक्री पूर्वानुमान, मार्केटिंग रणनीति

निर्णय प्रणाली (Decision Support Systems)

- AI सिस्टम जो CEO/मैनेजर को निर्णय लेने में मदद करते हैं
- उदाहरण:
  - AI-आधारित डैशबोर्ड
  - रीकमेंडेशन सिस्टम्स (Netflix, Amazon)

---

6 निष्कर्ष

AI में स्पेशलाइजेशन करने से करियर के अवसर कई गुना बढ़ जाते हैं।

क्षेत्र	मुख्य टूल्स	संभावित करियर
कंप्यूटर विज़न	OpenCV, YOLO, GANs	CV इंजीनियर, मेडिकल इमेज एनालिस्ट
NLP	GPT, BERT, LangChain	NLP डेवलपर, कंटेंट जनरेटर
रोबोटिक्स	ROS, DQN	ऑटोनॉमस इंजीनियर, मशीन कंट्रोल एक्सपर्ट
AI + बिज़नेस	BI टूल्स, ML	डाटा एनालिस्ट, AI कंसल्टेंट

Chapter-17

एडवांस AI इंफ्रास्ट्रक्चर: MLOps, डेटा इंजीनियरिंग और वेब इंटीग्रेशन

विषय-वस्तु:

- MLOps: CI/CD पाइपलाइन, Docker, Kubernetes
- डेटा इंजीनियरिंग: Big Data, Spark, ETL
- AI + Web Integration (Flask + React)

अनुक्रमणिका

1. भूमिका
2. MLOps क्या है?
3. CI/CD पाइपलाइन का परिचय
4. Docker क्या है और क्यों ज़रूरी है
5. Kubernetes द्वारा मॉडल प्रबंधन
6. डेटा इंजीनियरिंग की मूल बातें
7. Big Data और Distributed Systems
8. Apache Spark का उपयोग
9. ETL पाइपलाइन की समझ
10. AI और वेब का एकीकरण
11. Flask के साथ बैकएंड बनाना
12. React के साथ फ्रंटएंड इंटरफेस
13. Flask + React एकीकरण
14. क्लाउड पर डिप्लॉयमेंट
15. एक केस स्टडी
16. चुनौतियाँ और समाधान
17. टूल्स और तकनीकों की सूची
18. करियर के अवसर
19. अंतिम क्विज़
20. निष्कर्ष और आगे की दिशा

---

1. भूमिका

AI मॉडल को प्रोडक्शन में उपयोगी और स्केलेबल बनाने के लिए केवल प्रशिक्षण (training) ही पर्याप्त नहीं है।
ज़रूरत होती है — **सही पाइपलाइनों**, **ऑटोमेशन**, और **स्केलेबल इंफ्रास्ट्रक्चर** की।
यह ईबुक तीन मुख्य स्तंभों को कवर करता है:

- MLOps
- डेटा इंजीनियरिंग
- वेब इंटीग्रेशन

---

2. MLOps क्या है?

MLOps (Machine Learning Operations) मशीन लर्निंग को प्रोफेशनल प्रोडक्ट में बदलने की प्रक्रिया है। इसमें शामिल है:

- डेटा और मॉडल का version control
- मॉडल का परीक्षण और डिप्लॉयमेंट
- monitoring और अपडेट करना
- ऑटोमेशन और स्केलेबिलिटी

---

3. CI/CD पाइपलाइन

CI/CD (Continuous Integration/Deployment) एक ऑटोमेटेड प्रक्रिया है, जो कोड और मॉडल को test करके production में भेजती है।

CI: कोड या मॉडल बदलते ही टेस्टिंग शुरू
CD: सफल होने पर ऑटो डिप्लॉयमेंट

उपयोगी टूल्स:

- GitHub Actions
- Jenkins
- DVC (Data Version Control)
- MLflow

---

4. Docker: मॉडल को पैक करने का तरीका

Docker एक virtualization टूल है जो आपके प्रोजेक्ट को "container" में पैक करता है।

लाभ:

- सभी dependencies एक साथ
- कहीं भी रन हो सकता है
- टीम में आसानी से शेयर

उदाहरण Dockerfile:

```
FROM python:3.10
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

---

5. Kubernetes: ऑटोमैटेड मॉडल डिप्लॉयमेंट

Kubernetes एक ऑर्केस्ट्रेशन टूल है जो आपके कई Docker कंटेनर्स को स्केलेबल और ऑटोमैटेड तरीके से मैनेज करता है।

फायदे:

- Load balancing
- ऑटोमेटेड अपडेट्स
- High availability

उदाहरण: AI chatbot जो हजारों users को simultaneously response देता है।

---

6. डेटा इंजीनियरिंग: आधारभूत जानकारी

AI के लिए डेटा की मात्रा और गुणवत्ता महत्वपूर्ण है।
डेटा इंजीनियरिंग वह प्रक्रिया है जिसमें:

- डेटा को ingest करना
- Transform करना
- Storage और delivery के लिए pipelines बनाना शामिल है।

---

7. Big Data और Distributed Systems

Big Data तब आता है जब डेटा बहुत बड़ा होता है जिसे सामान्य कंप्यूटर संसाधित नहीं कर सकता।

उपयोगी टूल्स:

- Hadoop
- Apache Hive
- Google BigQuery

Distributed processing की ज़रूरत तब होती है जब लाखों रिकॉर्ड्स एक साथ प्रोसेस करने हों।

---

8. Apache Spark का परिचय

Apache Spark एक तेज़ और distributed processing इंजन है।

क्यों उपयोग करें?

- बड़ी मात्रा में डेटा के लिए
- Real-time स्ट्रीमिंग
- ML pipelines (Spark MLlib) के लिए

कोड उदाहरण (PySpark):

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("AIApp").getOrCreate()
data = spark.read.csv("data.csv")
```

---

9. ETL पाइपलाइन

ETL (Extract, Transform, Load)
AI मॉडल्स को अच्छी गुणवत्ता का डेटा देने के लिए यह जरूरी प्रक्रिया है।

उदाहरण:

- Extract: API या Database से डेटा निकालना
- Transform: Null values हटाना, scaling करना
- Load: AI मॉडल में इनपुट देना

टूल्स: Apache Airflow, Talend, Spark, Luigi

---

10. AI + वेब इंटीग्रेशन

AI मॉडल को प्रयोग में लाने के लिए उसे वेबसाइट या ऐप से जोड़ना पड़ता है।

उदाहरण:

- Text classification API
- Face detection webapp
- Customer chatbot

---

11. Flask API बनाना

Flask एक Python वेब फ्रेमवर्क है जिससे आप अपने AI मॉडल को API के रूप में एक्सपोज़ कर सकते हैं।

```
from flask import Flask, request
app = Flask(__name__)

@app.route("/predict", methods=["POST"])
def predict():
    data = request.json
    return {"output": "Positive"}

app.run()
```

---

12. React इंटरफेस बनाना

React एक फ्रंटएंड लाइब्रेरी है जिससे आप इंटरैक्टिव UI बना सकते हैं।

उदाहरण:

- Input फॉर्म
- आउटपुट दिखाना (जैसे "Spam" या "Not Spam")
- API से जुड़ना

---

13. Flask + React एकीकरण

कैसे काम करता है:

- Flask में ML मॉडल लोड
- React फ्रंटएंड से request भेजना
- JSON response दिखाना

लाभ:

- मॉड्यूलर
- आसान मेंटेनेंस
- स्केलेबिलिटी

---

14. क्लाउड डिप्लॉयमेंट

अपने AI+Web ऐप को क्लाउड पर तैनात करें:

- **Heroku**: आसान और फ्री
- **AWS EC2**: कंट्रोल अधिक
- **Google Cloud Run**: स्केलेबल

---

15. केस स्टडी

Project: Movie Review Sentiment App

- Flask में sentiment मॉडल
- React में textarea input
- Heroku पर होस्टेड

---

16. चुनौतियाँ और समाधान

चुनौतियाँ:

- डेटा ड्रिफ्ट
- मॉडल performance गिरना
- स्केलेबिलिटी

समाधान:

- डेटा और मॉडल version control
- मॉनिटरिंग tools
- Auto-retraining

---

17. टूल्स की सूची

श्रेणी	टूल्स
MLOps	GitHub Actions, Docker, Kubernetes
डेटा इंजीनियरिंग	Spark, Airflow, Hive
वेब	Flask, React, Axios
क्लाउड	Heroku, AWS, GCP

---

18. करियर के अवसर

- MLOps इंजीनियर
- डेटा इंजीनियर

- AI वेब डेवलपर
- Full Stack AI डेवलपर

सैलरी रेंज: ₹5 लाख - ₹25 लाख/वर्ष तक

19.अंतिम क्विज़

Q1. Docker का मुख्य लाभ क्या है?
A) Accuracy बढ़ाना
B) Portability □
C) डेटा सफाई
D) None

Q2. ETL में "T" का मतलब क्या है?
A) Transfer
B) Train
C) Transform □
D) Target

---

20. निष्कर्ष और आगे की दिशा

आपने सीखा:

- कैसे AI प्रोजेक्ट्स को डिप्लॉय और स्केल किया जाता है
- डेटा प्रोसेसिंग और ETL का महत्व
- वेब से AI को कैसे integrate करें

Chapter-18

एआई अनुसंधान और इंडस्ट्री एप्लिकेशन

विषयवस्तु:

- शोध पत्र पढ़ना और लिखना
- कैग्गल प्रतियोगिताएँ और मॉडल बेंचमार्किंग
- वास्तविक उद्योग समस्याओं का समाधान

अनुक्रमणिका:

1. परिचय
2. एआई शोध का महत्व
3. शोध पत्र की संरचना
4. शोध पत्र कैसे पढ़ें
5. प्रासंगिक शोध कैसे खोजें
6. शोध पत्र कैसे लिखें
7. शोध में उपयोगी उपकरण
8. सामान्य गलतियाँ
9. कैग्गल का परिचय
10. कैग्गल की प्रतियोगिताओं के प्रकार
11. कैग्गल पर शुरुआत कैसे करें
12. मॉडल बेंचमार्किंग क्या है
13. फ़ीचर इंजीनियरिंग
14. एनसेंबल तकनीक
15. कैग्गल केस स्टडी
16. उद्योग समस्या क्या होती है
17. हेल्थकेयर/रिटेल में एआई केस स्टडी
18. स्टेकहोल्डर के साथ काम करना
19. मॉडल से प्रोडक्शन तक
20. सारांश और आगे की राह

---

1. परिचय

यह ई-बुक अकादमिक और इंडस्ट्री दोनों पक्षों से एआई को समझने में मदद करती है। शोध, कैग्गल, और उद्योग समस्याओं को हल करने का तरीका इसमें विस्तार से समझाया गया है।

---

2. एआई शोध का महत्व

एआई शोध नई तकनीकों, जिम्मेदार एआई, और बायस को समझने का तरीका देता है। उदाहरण: BERT, GPT, YOLO जैसे मॉडल शोध का ही परिणाम हैं।

---

3. शोध पत्र की संरचना

एक आदर्श शोध पत्र में होते हैं:

- सारांश (Abstract)
- प्रस्तावना (Introduction)
- संबंधित कार्य (Literature Review)
- कार्यविधि (Methodology)
- प्रयोग (Experiments)
- परिणाम और विश्लेषण (Results & Discussion)
- निष्कर्ष (Conclusion)
- सन्दर्भ (References)

---

4. शोध पत्र कैसे पढ़ें

- पहले सारांश और निष्कर्ष पढ़ें
- फिर प्रस्तावना और परिणाम
- अंत में कार्यविधि और प्रयोग को समझें
- "Papers with Code" वेबसाइट पर कोड देखें

---

5. प्रासंगिक शोध कैसे खोजें

स्रोत:

- Google Scholar
- arXiv.org
- IEEE Xplore
- Springer, Elsevier
- ResearchGate

---

6. शोध पत्र कैसे लिखें

- एक स्पष्ट विषय चुनें

- समस्या, विधि और परिणाम पर फोकस करें
- ग्राफ, तालिका और कोड संलग्न करें
- Overleaf पर LaTeX का उपयोग करें
- ज़ोतेरो या मेंडेली से रेफरेंस जोड़ें

---

7. शोध में उपयोगी उपकरण

- **Overleaf / LaTeX** – लेखन
- **BibTeX** – संदर्भ प्रबंधन
- **Grammarly** – भाषा सुधार
- **GitHub** – कोड साझा करना

---

8. सामान्य गलतियाँ

- बिना तुलना के मॉडल देना
- गलत डेटा प्रयोग
- असंपूर्ण या बिना परीक्षण के निष्कर्ष
- कोड साझा न करना

---

9. कैग्गल का परिचय

Kaggle एक प्लेटफॉर्म है:

- डेटा प्रतियोगिताओं के लिए
- डेटा साझा करने के लिए
- सीखने और प्रैक्टिस के लिए
- नेटवर्क बनाने के लिए

---

10. कैग्गल की प्रतियोगिताओं के प्रकार

- Getting Started: शुरुआती के लिए
- Featured: स्पॉन्सर की गई
- Research: अकादमिक आधारित
- Hiring: नौकरियों के लिए

---

11. कैग्गल पर शुरुआत कैसे करें

- kaggle.com पर खाता बनाएं
- "Titanic" जैसी बेसिक प्रतियोगिता से शुरुआत करें
- टॉप कोड (Notebook) पढ़ें और समझें
- CSV फ़ाइल बनाकर पहली सबमिशन करें

---

12. मॉडल बेंचमार्किंग क्या है

मॉडल की तुलना एक समान मेट्रिक पर की जाती है।
उदाहरण:

- Classification: Accuracy, F1-score
- Regression: RMSE, MAE

बेसलाइन मॉडल: Logistic Regression, Decision Tree

---

13. फ़ीचर इंजीनियरिंग

- स्केलिंग, नॉर्मलाइजेशन
- वन-हॉट एन्कोडिंग
- Missing Values का समाधान
- डोमेन ज्ञान से नए फ़ीचर बनाना

---

14. एनसेंबल तकनीक

कई मॉडल के आउटपुट को मिलाकर सटीकता बढ़ाना।

उदाहरण:

- Bagging: Random Forest
- Boosting: XGBoost
- Voting, Stacking

---

15. कैग्गल केस स्टडी

House Prices Prediction

टेक्निक्स:

- Missing value imputation
- Feature scaling
- XGBoost + LightGBM ensemble

---

16. उद्योग समस्या क्या होती है

- अधूरा और असंगठित डेटा
- स्पष्ट KPI (Key Performance Indicators)
- वास्तविक समय में उपयोग
- ग्राहक या कंपनी की आवश्यकता

---

17. हेल्थकेयर/रिटेल में एआई केस स्टडी

रिटेल में ग्राहक चर्न भविष्यवाणी

चरण:

- डेटा संग्रह
- मॉडलिंग
- परिणाम प्रस्तुति
- रणनीतिक निर्णय

---

18. स्टेकहोल्डर के साथ काम करना

- सरल भाषा में रिपोर्ट
- व्यावसायिक प्रभाव की व्याख्या
- समस्याओं पर चर्चा
- परिणामों का डैशबोर्ड बनाना

---

19. मॉडल से प्रोडक्शन तक

- समस्या की परिभाषा
- डाटा प्रोसेसिंग
- मॉडल ट्रेन्ड करना
- Flask / Streamlit से डेमो

- Docker या क्लाउड से डिप्लॉयमेंट

---

20. सारांश और आगे की राह

आपने सीखा:

- शोध कैसे पढ़ें और लिखें
- कैग्गल पर भाग कैसे लें
- इंडस्ट्री समस्याओं को कैसे हल करें

GOVT OF INDIA CERTIFIED SARVA EDUCATION ISO 9001:2015 CERTIFIED Start New Computer Centre Sarvaindia.com Join TODAY

Chapter-19

GOVT OF INDIA CERTIFIED SARVA EDUCATION ISO 9001:2015 CERTIFIED Start New Computer Centre Sarvaindia.com Join TODAY

अंतिम प्रमुख परियोजना एवं करियर मार्गदर्शन

उद्योग स्तर की प्रोजेक्ट तैयारी, दस्तावेज़ीकरण, प्रस्तुति एवं करियर निर्माण

अनुक्रमणिका (Contents)

1. परिचय
2. अंतिम प्रमुख परियोजना क्या है?
3. सही AI प्रोजेक्ट कैसे चुनें
4. उद्योग-स्तर के प्रोजेक्ट सुझाव
5. प्रोजेक्ट योजना और टाइमलाइन
6. आवश्यक टूल्स और तकनीकी स्टैक
7. डेटा संग्रह और तैयारी
8. मॉडल बनाना और मूल्यांकन
9. दस्तावेज़ीकरण (Documentation)
10. तकनीकी रिपोर्ट तैयार करना
11. प्रोजेक्ट प्रस्तुति (Presentation)
12. डेमो और वेब ऐप बनाना
13. वर्शन कंट्रोल (GitHub)
14. प्रमाणपत्र (Certification)
15. अतिथि व्याख्यान (Guest Lectures)
16. इंटर्नशिप के अवसर
17. AI करियर के विकल्प
18. रिज़्यूमे और पोर्टफोलियो
19. इंटरव्यू तैयारी
20. अंतिम सुझाव और भविष्य की दिशा

---

1. परिचय

यह ई-बुक AI कोर्स के अंतिम चरण को समर्पित है, जहाँ आप एक वास्तविक परियोजना को तैयार करते हैं, उसका दस्तावेज़ बनाते हैं, प्रेजेंट करते हैं और करियर की दिशा तय करते हैं।

---

2. अंतिम प्रमुख परियोजना क्या है?

यह एक **Capstone Project** होती है जो आपने कोर्स में सीखी हर चीज़ को जोड़कर एक संपूर्ण प्रोजेक्ट में बदल देती है।

---

3. सही AI प्रोजेक्ट कैसे चुनें

- अपनी रुचि के क्षेत्र चुनें (NLP, Vision, Chatbot आदि)
- ऐसा विषय चुनें जो समाज या इंडस्ट्री की समस्या सुलझा सके
- डेटा आसानी से उपलब्ध हो
- प्रोजेक्ट 4–6 सप्ताह में पूरा किया जा सके

---

4. उद्योग-स्तर के प्रोजेक्ट सुझाव

- स्पैम ईमेल डिटेक्शन
- मेडिकल इमेज क्लासिफिकेशन
- रियल एस्टेट प्राइस प्रेडिक्शन
- चैटबॉट फॉर स्कूल/कॉलेज
- ग्राहक भावना विश्लेषण (Sentiment Analysis)

---

5. प्रोजेक्ट योजना और टाइमलाइन

सप्ताह 1: समस्या का विश्लेषण और डेटा संग्रह
सप्ताह 2: डेटा प्रोसेसिंग और एनालिसिस
सप्ताह 3: मॉडल ट्रेनिंग और टेस्टिंग
सप्ताह 4: यूआई/डेमो और रिपोर्ट
सप्ताह 5-6: प्रेजेंटेशन और फ़ाइनल सबमिशन

---

6. आवश्यक टूल्स और तकनीकी स्टैक

- **Python**, Numpy, Pandas
- **Scikit-learn, TensorFlow, Keras**
- **Matplotlib, Seaborn**
- **Streamlit या Flask**
- **GitHub** (वर्शन कंट्रोल)

---

7. डेटा संग्रह और तैयारी

- Kaggle, UCI या APIs से डेटा प्राप्त करें
- साफ-सफाई, नॉर्मलाइज़ेशन और एनकोडिंग करें

- विज़ुअलाइज़ेशन बनाएं

---

8. मॉडल बनाना और मूल्यांकन

- सही एल्गोरिदम का चयन करें
- हाइपरपैरामीटर ट्यून करें
- एक्यूरेसी, F1 स्कोर आदि से मूल्यांकन करें
- मॉडल को Pickle/Joblib से सेव करें

---

9. दस्तावेज़ीकरण (Documentation)

- समस्या विवरण
- डेटा स्रोत
- प्रयुक्त टूल्स
- मॉडलिंग प्रक्रिया
- परिणाम
- निष्कर्ष और भविष्य कार्य

---

10. तकनीकी रिपोर्ट तैयार करना

- शीर्षक और सारांश
- कार्यप्रणाली
- ग्राफ, चार्ट और विजुअल
- कोड स्निपेट
- रेफरेंस और GitHub लिंक

---

11. प्रोजेक्ट प्रस्तुति (Presentation)

- स्लाइड संक्षिप्त और स्पष्ट होनी चाहिए
- ग्राफिक्स और डेमो वीडियो जोड़ें
- सवाल-जवाब के लिए तैयार रहें
- 10–15 मिनट की समयसीमा में रहें

---

12. डेमो और वेब ऐप बनाना

- Streamlit या Flask का उपयोग करें
- UI को उपयोगकर्ता के अनुकूल बनाएं
- ऑप्शनल: डाउनलोड बटन, ग्राफ, आदि

---

13. वर्शन कंट्रोल (GitHub)

- GitHub पर कोड नियमित रूप से अपलोड करें
- README फ़ाइल में प्रोजेक्ट सारांश जोड़ें
- यह आपके पोर्टफोलियो में काम आता है

---

14. प्रमाणपत्र (Certification)

- रिपोर्ट, डेमो और प्रेजेंटेशन जमा करें
- जूरी द्वारा मूल्यांकन किया जाएगा
- सफल छात्रों को प्रमाण पत्र प्रदान किया जाएगा

---

15. अतिथि व्याख्यान (Guest Lectures)

- उद्योग विशेषज्ञों से सीखने का मौका
- वास्तविक AI प्रोजेक्ट कैसे होते हैं
- नेटवर्किंग और मार्गदर्शन

---

16. इंटर्नशिप के अवसर

- Internshala, Naukri, LinkedIn पर आवेदन करें
- स्थानीय स्टार्टअप्स से संपर्क करें
- इंटर्नशिप से अनुभव और प्रोजेक्ट बढ़ेगा

---

17. AI करियर के विकल्प

- मशीन लर्निंग इंजीनियर
- डेटा वैज्ञानिक
- एनएलपी स्पेशलिस्ट
- कंप्यूटर विजन इंजीनियर
- रिसर्च एसोसिएट

---

18. रिज़्यूमे और पोर्टफोलियो

- संक्षिप्त परिचय
- तकनीकी स्किल्स
- GitHub, Kaggle, लिंक्डइन लिंक
- प्रोजेक्ट हाइलाइट्स

---

19. इंटरव्यू तैयारी

- डेटा स्ट्रक्चर और एल्गोरिदम
- प्रोजेक्ट समझाने की कला
- MCQ + कोडिंग राउंड
- AI केस स्टडी विश्लेषण

---

20. अंतिम सुझाव और भविष्य की दिशा

बधाई! आपने AI सीखने की एक यात्रा पूरी की।
अब आप:

- इंडस्ट्री के लिए AI प्रोजेक्ट बना सकते हैं
- नौकरी या स्टार्टअप में जा सकते हैं
- रिसर्च में जा सकते हैं

निरंतर सीखते रहें, प्रयोग करते रहें। AI आपका भविष्य है!

Visit: **www.sarvaindia.com**