UNIT-5

Linear Regression in Healthcare

Learning Objectives

- Understand the concept and mathematical formulation of linear regression
- Apply linear regression to healthcare datasets
- Interpret regression coefficients in a clinical context

Understand model evaluation metrics for regression

1. Introduction to Linear Regression

What is Linear Regression?

- A **supervised learning algorithm** for **predicting continuous values**
- Models the **relationship between dependent variable (Y)** and one or more **independent variables (X)**

Types:

- **Simple Linear Regression**: One predictor
- **Multiple Linear Regression**: More than one predictor

2. Mathematical Foundation & Assumptions

- **Equation:**

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$$

  - Y: Dependent variable (e.g., hospital stay length)
  - X_i: Independent variables (e.g., age, blood pressure)
  - β_i: Coefficients
  - ϵ: Error term

Assumptions:

1. **Linearity**: Relationship between inputs and output is linear
2. **Independence**: Observations are independent
3. **Homoscedasticity**: Constant variance of errors
4. **Normality**: Errors are normally distributed
5. **No multicollinearity**: Predictors should not be highly correlate

3. Application in Healthcare + Case Study

Real-World Use Cases:

- Predicting **hospital stay length**
- Estimating **healthcare costs**
- Monitoring **disease progression**
- Modeling **patient vitals**

Case Study: Predicting Diabetes Progression

Dataset: Diabetes Dataset from sklearn
Features: BMI, Age, Blood Pressure, Blood Sugar
Target: Disease progression metric (quantitative score)

Steps:

1. **Load Dataset**

```
python
CopyEdit
from sklearn.datasets import load_diabetes
X, y = load_diabetes(return_X_y=True)
```

2. **Split & Train Model**

```
python
CopyEdit
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
model = LinearRegression().fit(X_train, y_train)
```

3. **Predict & Interpret Coefficients**

```
python
CopyEdit
model.coef_, model.intercept_
```

Interpretation:

Coefficients tell us the **change in outcome per unit change in input** (e.g., 1 unit increase in BMI increases disease score by X)

Support Vector Machines (SVM) in Healthcare

Learning Objectives

- Understand the intuition and mathematics behind SVM
- Explore how SVM works for classification problems
- Apply SVM models to real healthcare data
- Evaluate SVM model performance

A healthcare case study using SVM

1. Introduction to SVM

What is SVM?

- A **supervised machine learning algorithm** used mainly for **classification**
- Finds the **optimal hyperplane** that separates different classes

Key Idea:

- SVM looks for the **maximum margin** (distance) between the classes
- Effective in **high-dimensional spaces**

Works well with both **linear and non-linear** decision boundaries

2. Mathematical Intuition & Kernel Trick

Hyperplane:

- A decision boundary that separates classes.
- In 2D, it's a line; in 3D, a plane.

Margin:

- Distance between the hyperplane and the closest support vectors (data points)
- **Maximizing margin** → Better generalization

Support Vectors:

- Data points **closest to the hyperplane**
- They **"support"** or define the decision boundary

Kernel Trick:

- Transforms data into **higher-dimensional space** to make it linearly separable.

Popular Kernels:

- **Linear**: Fast, good for text data

- **Polynomial**: Captures curved boundaries
- **RBF (Gaussian)**: Very powerful, default choice in non-linear problems

3. Application in Healthcare + Case Study

Use Cases in Healthcare:

- **Disease classification**: E.g., diabetes, cancer, heart disease
- **Medical image classification**: Detecting tumors in MRI, X-rays
- **Patient risk prediction**: Identifying high-risk patients from EMRs

Case Study: Breast Cancer Diagnosis

Dataset: Breast Cancer Wisconsin Dataset (from `sklearn.datasets`)

Goal: Classify tumors as *malignant* or *benign*

Steps:

```
python
CopyEdit
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix

# Load Data
data = load_breast_cancer()
X = data.data
y = data.target

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Train SVM Model
model = SVC(kernel='rbf', C=1.0, gamma='scale')
model.fit(X_train, y_train)

# Evaluate
y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Output Discussion:

- Precision, Recall, F1-score for each class
- Importance of sensitivity in healthcare (e.g., catching all malignant tumors)

Model Evaluation for SVM

Metrics:

Metric	Description
Accuracy	Overall correctness
Precision	TP / (TP + FP) – Important in low false positives
Recall (Sensitivity)	TP / (TP + FN) – Crucial in healthcare
F1-Score	Harmonic mean of precision & recall
ROC-AUC	Area under the ROC curve for binary classifiers

Summary

Concept	Key Point
SVM	Classifier using maximum-margin principle
Kernels	Allow non-linear classification
Healthcare Use	Classification of diseases, diagnostics
Evaluation	Accuracy, Precision, Recall, F1, ROC-AUC

Random Forests in Healthcare

Learning Objectives

- Understand the concept of decision trees and Random Forests (RF)
- Explore how Random Forests improve model performance
- Apply Random Forests to healthcare datasets
- Evaluate model accuracy and interpret results
- Analyze a healthcare case study using Random Forests

1. Introduction to Decision Trees & Random Forest

Decision Tree Recap:

- A tree-like model that splits data into branches to reach a decision.
- **Root Node** → internal decision nodes → **leaf nodes** (outcomes)

Random Forest:

- An **ensemble method** that combines **multiple decision trees**
- Final prediction is made by **majority voting** (classification) or **averaging** (regression)

"A forest of uncorrelated trees is more accurate than any of the individual trees."

2. Working of Random Forest

Key Concepts:

- **Bootstrap Sampling**: Each tree gets a random subset of data.
- **Feature Randomness**: At each split, only a random subset of features is considered.
- **Ensemble Averaging**: Reduces variance and avoids overfitting.

Benefits in Healthcare:

- Handles **missing data** and **imbalanced datasets** well
- Works with **high-dimensional** datasets

Provides **feature importance** scores

3. Applications in Healthcare + Case Study

Use Cases:

- **Disease prediction**: Heart disease, diabetes, cancer
- **Patient stratification**: Risk levels, treatment plans
- **Clinical decision support systems**
- **Readmission prediction**

Case Study: Diabetes Prediction

Dataset: PIMA Indians Diabetes Dataset

Goal: Predict whether a patient has diabetes (binary classification)

Sample Code (Using Scikit-learn):

```
python
CopyEdit
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load data (you can use pandas to load PIMA CSV if preferred)
import pandas as pd
df = pd.read_csv("diabetes.csv")
X = df.drop("Outcome", axis=1)
y = df["Outcome"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Random Forest
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Feature Importance Plot:

```
python
CopyEdit
import matplotlib.pyplot as plt
import seaborn as sns

feature_imp = pd.Series(model.feature_importances_,
index=X.columns).sort_values(ascending=False)
sns.barplot(x=feature_imp, y=feature_imp.index)
plt.title("Feature Importance")
plt.show()
```

Discussion:

- Which features contribute most to the outcome? (e.g., glucose level, BMI)
- Why is interpretability important in healthcare?

4. Model Evaluation

Metric	Description
Accuracy	Overall correct predictions
Precision	TP / (TP + FP) – Useful in reducing false positives
Recall (Sensitivity)	TP / (TP + FN) – Key in medical diagnosis
F1 Score	Balance of precision and recall
AUC-ROC	Area under the ROC curve for binary classification

Summary

Concept	
Random Forest	Ensemble of decision trees using majority voting
Strengths	Robust, reduces overfitting, handles noise
Healthcare Use	Disease classification, risk prediction, diagnostics
Evaluation	Accuracy, Precision, Recall, AUC-ROC

Convolutional Neural Networks (CNNs) in Healthcare

Learning Objectives

- Understand the architecture and function of CNNs
- Describe how CNNs process medical images
- Explore CNN applications in diagnostic imaging
- Analyze a healthcare case study using CNN
- Evaluate CNN model performance in healthcare settings

1. CNN Overview and Biological Inspiration

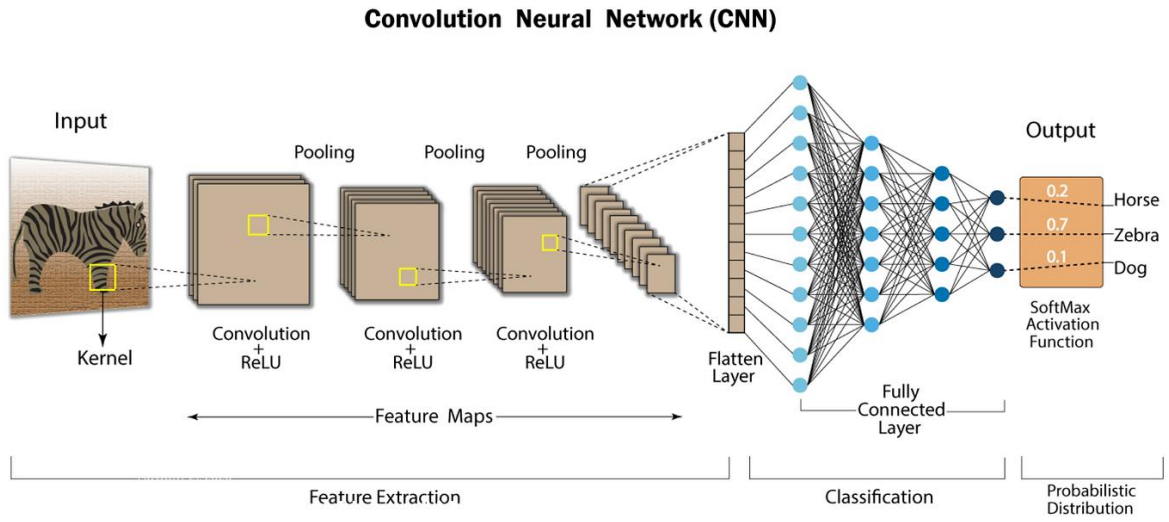
- **Biological analogy**: Inspired by the human visual cortex.
- **Traditional ML vs CNN**:
  - Traditional ML requires **manual feature extraction**
  - CNNs **automatically learn features** from data

CNNs are excellent for image-based tasks such as radiology, pathology, dermatology, and ophthalmology.

2. CNN Architecture and Layers

Main Layers in a CNN:

Layer	Function
Convolution Layer	Detects local features (edges, shapes)
ReLU Activation	Adds non-linearity
Pooling Layer	Reduces dimensionality (e.g., MaxPooling)
Fully Connected Layer	Performs classification
Softmax Layer	Outputs class probabilities



3. CNN Applications in Healthcare (20–35 mins)

Use Cases:

- **Radiology**: Detecting pneumonia, TB, tumors in X-rays, CT, MRI
- **Ophthalmology**: Retinal disease detection (e.g., Diabetic Retinopathy)
- **Dermatology**: Skin lesion classification (melanoma vs benign)
- **Pathology**: Cancer cell detection in biopsy slides

CNNs can outperform or assist radiologists in image-based diagnosis.

---

4. Case Study: Pneumonia Detection Using Chest X-rays (35–50 mins)

Dataset:

- **Chest X-Ray Images (Pneumonia)** – available on Kaggle or NIH

Problem:

- Classify X-ray images into:
  - **Normal**
  - **Pneumonia**

CNN Code (Keras/TensorFlow):

```
python
CopyEdit
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 1)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Data (Assuming preprocessed)
train_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory('train/',
target_size=(150, 150), color_mode='grayscale')
```

```
model.fit(train_generator, epochs=10)
```

Results:

- Accuracy: ~85–90%
- Confusion Matrix, ROC Curve
- Discuss **false positives vs false negatives** in medical settings

---

5. Evaluation & Challenges

Model Evaluation:

Metric	Why it matters
Accuracy	General performance
Precision	Reducing false positives (e.g., wrongly diagnosing pneumonia)
Recall	Reducing false negatives (missing actual disease)
F1 Score	Balance of precision and recall
AUC-ROC	Model's ability to discriminate between classes

Limitations:

- Need for **large labeled datasets**
- **Interpretability** is still limited ("black box" issue)
- **Generalizability** across hospitals and populations

Open Discussion:

- How can explainable AI (XAI) improve CNN adoption in healthcare?
- Should CNNs be used as assistants or autonomous systems?

Summary Slide

Concept	Summary
CNN	Deep learning model for image data
Layers	Convolution, Pooling, ReLU, FC
Healthcare Use	X-rays, MRIs, skin lesions, retinal scans
Key Metric	Recall and AUC critical in medical context

Recurrent Neural Networks (RNNs) in Healthcare

Learning Objectives

- Understand RNN and its architecture
- Apply RNNs for sequential data in healthcare
- Explore applications like patient monitoring and clinical event prediction
- Analyze a healthcare case study using RNN
- Identify limitations and potential improvements (e.g., LSTM, GRU)

---

1. Sequential Data and RNN Concepts

What is Sequential Data?

- Time-series data from wearables (heart rate, ECG, BP)
- EHR event logs (admissions, medication, vitals)
- Text data (doctor's notes)

Why Traditional Models Fail:

- Standard ML/DL models assume independence between data points.
- Sequential nature is **ignored**, which is crucial in healthcare.

What is RNN?

- RNNs are designed for **temporal (sequence) data**.
- Maintains **hidden state** to preserve history.
- Output at time *t* depends on input at time *t* and previous states.

---

2. RNN Architecture and Limitations

Vanilla RNN Structure:

```
makefile
CopyEdit
xt → [Input]
ht = tanh(W·xt + U·ht-1 + b) → [Hidden State]
yt = softmax(V·ht + c) → [Output]
```

- **Looping structure** allows memory of past inputs
- Weights shared across time steps

Challenges:

- **Vanishing/Exploding gradients** during backpropagation
- Difficult to learn **long-term dependencies**

Solutions:

- **LSTM (Long Short-Term Memory)**
- **GRU (Gated Recurrent Unit)**

Diagram:

- Show basic RNN and LSTM cell (optional animation or simplified diagram)

---

3. RNN Applications in Healthcare

Use Cases:

Application	Description
Clinical Event Prediction	Predict future diagnoses, medications, or ICU interventions
Vital Sign Forecasting	Predict heart rate, BP, glucose levels
Disease Progression	Model stages of diseases (e.g., diabetes, Alzheimer's)
Mortality Risk	Predict death risk over next 24–48 hours in ICU
Medical Text Mining	Analyze physician notes or discharge summaries

RNNs are powerful for **longitudinal patient data** where time matters.

---

4. Case Study: Vital Signs Forecasting in ICU

Dataset:

- **MIMIC-III** – ICU data including vitals, labs, meds

Task:

- Predict future **heart rate and respiratory rate** based on past 12 hours

RNN (LSTM-based) Code Sample:

```
python
CopyEdit
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
import numpy as np

# Input shape: (samples, timesteps, features)
```

```
X_train = np.random.rand(1000, 12, 5)  # 12 timesteps, 5 vitals
y_train = np.random.rand(1000, 1)      # Predict next HR

model = Sequential([
    LSTM(64, input_shape=(12, 5)),
    Dense(1)  # Output: predicted heart rate
])

model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=10)
```

Discussion:

- **Evaluate** using RMSE, MAE
- Plot: Actual vs Predicted HR
- Impact: Anticipate deterioration, alert clinicians earlier

---

5. Summary & Q&A

Key Takeaways:

Topic	Summary
RNN	Designed for sequence data
Limitation	Can struggle with long-term dependencies
Solution	Use LSTM or GRU
Application	Forecast vitals, predict events, understand progressions
Case Study	RNN for HR forecasting in ICU patients

Challenges, Evaluation, and Deployment of ML/DL in Healthcare

---

Learning Objectives

- Understand challenges in developing and applying ML/DL models in healthcare.
- Identify key evaluation metrics suitable for clinical tasks.
- Explore model interpretability and ethical concerns.
- Learn strategies for deploying healthcare models in real-world environments.

---

1. Challenges in Healthcare ML/DL

A. Data-Related Challenges

- **Heterogeneous data**: Structured (EHR), unstructured (notes), images, signals
- **Missing or incomplete data**: Due to human error, irregular sampling
- **Data privacy and compliance**: HIPAA, GDPR
- **Small sample sizes**: Especially for rare diseases

B. Domain-Specific Challenges

- **Need for clinical validation**
- **Labeling issues**: Expert annotations are expensive
- **High risk of error**: Lives may be affected

C. Model-Specific Challenges

- Overfitting on small datasets
- Lack of generalization across hospitals or populations
- Temporal distribution shift (concept drift)

---

2. Evaluation Metrics in Healthcare

A. Binary Classification Metrics

Metric	Formula	Healthcare Use
Accuracy	(TP+TN)/(Total)	Misleading with class imbalance
Precision	TP / (TP + FP)	Important in diagnosis
Recall (Sensitivity)	TP / (TP + FN)	Critical for detecting conditions
F1 Score	2·(P·R)/(P+R)	Balance between precision & recall
AUC-ROC	Area under ROC curve	Discrimination power

B. Multi-class or Regression Tasks

- **RMSE / MAE**: For vital signs prediction
- **Confusion Matrix**: For multiple disease classification
- **Calibration Curve**: Is the probability estimate accurate?

C. Special Considerations

- Time-aware metrics (early warning score accuracy)
- Clinical Utility Score (how useful a prediction is in practice)

---

3. Interpretability, Ethics & Bias

A. Interpretability

- Black-box models (e.g., deep neural nets) are **hard to trust**
- Clinicians demand **explainable AI**
- Use:
    - **SHAP / LIME** for feature attribution
    - **Saliency maps** for imaging models
    - Attention layers in RNNs (for text or sequence)

B. Bias in Models

- Trained on non-representative data
- May underperform for certain groups (e.g., race, gender)
- Reinforces health disparities

C. Ethical Considerations

- Informed consent for data use
- Model auditability & accountability
- Explainability for patient decisions

---

4. Deployment in Real-World Settings

A. From Model to Product

- **Model serving** using APIs (e.g., Flask, FastAPI)
- **Real-time inference** in hospital systems
- Need for **data pipelines**, continuous monitoring

B. Human-in-the-Loop

- Doctors review AI recommendations before action
- Improves safety and acceptance

C. Continuous Learning & Monitoring

- Models must be updated with new data
- **Drift detection** to ensure relevance
- Monitor performance degradation over time

D. Regulatory and Legal Aspects

- FDA approval in the U.S. for AI in medical devices
- Audit trails and explainability for decisions

---

5. Summary & Discussion

Key Takeaways:

Theme	Summary
Data	Complex, incomplete, and sensitive
Metrics	Must be clinically meaningful
Interpretability	Required for trust and adoption
Deployment	Needs robust infrastructure, monitoring, and compliance
Ethics	Avoid bias, ensure fairness and transparency