Experiment No.: 01

Title: To Clean, Integrate and Transform Electronic Healthcare Records

What We Do in This Experiment

1. **Import Libraries**
   - Python libraries like `numpy`, `pandas`, `matplotlib`, `seaborn` are imported for data handling, analysis, and visualization.
2. **Read Datasets**
   - `patients.csv` → Contains patient details
   - `conditions.csv` → Contains patient conditions
3. **Rename Columns**
   - Standardize the key column to `PATIENTID` in both datasets for merging.
4. **Clean Data**
   - Drop unwanted columns (`DRIVERS`, `SUFFIX`)
   - Remove duplicate records
5. **Integrate Datasets**
   - Merge patients and conditions datasets on `PATIENTID` using **inner join** to get only matching records.
6. **Transform Data**
   - Convert `START` and `STOP` columns to **datetime format**
   - Calculate healthcare coverage length in days:
     `HEALTHCARE_COVERAGE_LENGTH = STOP - START`
   - Map categorical variables for clarity:
     `GENDER` → `M: Male`, `F: Female`
7. **Save Cleaned Dataset**
   - Export the cleaned and transformed dataset as `cleaned_records.csv` for further analysis.

Practical Application:

- The cleaned and transformed dataset is ready for healthcare analytics, predictive modeling, and research.

Likely Viva Questions and Answers

Basic Conceptual Questions

1. **Q:** What is healthcare data?
   A: Patient information including demographics, medical history, conditions, treatments, and medications.

2. **Q:** Why do we clean and transform healthcare data?
   A: To remove errors, duplicates, and inconsistencies, making the data usable for analysis.
3. **Q:** What are the primary sources of healthcare data?
   A: Hospitals (EMRs), insurance companies (claims), public health databases, and wearable devices.
4. **Q:** What challenges exist in sharing healthcare data securely?
   A: Privacy concerns, data breaches, and regulatory compliance. Solutions include encryption, secure protocols, and anonymization.
5. **Q:** What ethical considerations are involved in sharing healthcare data?
   A: Maintaining confidentiality, obtaining informed consent, and using data responsibly for research.

Experiment-Specific Questions

6. **Q:** Which Python libraries were used and why?
   A: `pandas` for data handling, `numpy` for numerical operations, `matplotlib` & `seaborn` for visualization.
7. **Q:** Why do we rename columns before merging?
   A: To ensure both datasets have a common key (`PATIENTID`) for merging.
8. **Q:** How is healthcare coverage length calculated?
   A: `HEALTHCARE_COVERAGE_LENGTH = STOP - START` in days using pandas datetime.
9. **Q:** How are categorical variables transformed?
   A: Mapping values (`M → Male`, `F → Female`) to make the data consistent.
10. **Q:** What does `drop_duplicates()` do?
    A: Removes repeated records to avoid redundancy.
11. **Q:** Difference between cleaning and transforming data?
    A: Cleaning removes errors/unwanted data; transforming changes format or derives new features.
12. **Q:** Difference between inner, outer, left, and right merge?
    A:
    - Inner → Only matching records
    - Outer → All records from both datasets
    - Left → All left + matching right
    - Right → All right + matching left
13. **Q:** How is this dataset useful for healthcare analytics?
    A: Enables analysis of coverage, conditions, and demographics for research and decision-making.

Advanced / Extra Questions

14. **Q:** How would you handle missing data in healthcare datasets?
    A: Fill with mean/median/mode or remove rows/columns with too many missing values.
15. **Q:** How would you anonymize sensitive patient information?
    A: Remove personally identifiable info or encode it using hashing/pseudonyms.

16. **Q:** Why convert date columns to datetime format?
    A: To perform calculations like coverage length, sorting, and filtering by date.
17. **Q:** How can you visualize healthcare coverage data?
    A: Histograms (coverage length), pie charts (gender distribution), bar/line graphs (condition trends).

Experiment No.: 02

Title: To Apply Various Data Analysis and Visualization Techniques on Electronic Healthcare Records (EHR)

What We Do in This Experiment

1. **Import Libraries**
   - `numpy`, `pandas` → Data handling and preprocessing
   - `matplotlib`, `seaborn` → Data visualization
   - `plotly.express`, `folium` → Interactive maps and geospatial visualization
   - `sklearn` → Preprocessing, clustering (k-Means), PCA
2. **Read Datasets**
   - `patients.csv` → Patient demographic and healthcare data
   - `conditions.csv` → Patient medical conditions
   - Standardize column names (`PATIENTID`) for integration
3. **Data Cleaning**
   - Drop irrelevant columns: `DRIVERS, SUFFIX, MAIDEN, PREFIX, PASSPORT`
   - Remove duplicate records
4. **Data Integration**
   - Merge patients and conditions datasets using **inner join** on `PATIENTID`
5. **Visualization**
   - **Countplots** → City-wise patient distribution, condition-wise distribution
   - **Scatter plots & Mapbox** → Geospatial visualization of patients with healthcare expenses
   - **Interactive Folium Maps** → Show clusters of patients geographically
6. **Data Transformation for Clustering**
   - One-hot encode categorical features (`DESCRIPTION`)
   - Group by city and sum occurrences
7. **k-Means Clustering**
   - Determine optimal number of clusters using **Elbow Method**
   - Apply clustering to group similar cities based on patient conditions
8. **PCA (Principal Component Analysis)**
   - Reduce dimensionality of data for visualization
   - Analyze variance explained by principal components
9. **Cluster Visualization**
   - Scatter plots using PCA components
   - Bar plots to visualize conditions per cluster
10. **Map Clusters on Folium Map**
    - Circle markers sized by patient count
    - Color-coded by cluster

Practical Application:

- Helps in **pattern detection**, **healthcare resource allocation**, and **decision-making** based on geographic and condition-based analysis of patients.

Likely Viva Questions and Answers

Basic Conceptual Questions

1. **Q:** What are structured vs unstructured healthcare data?
   A:
   - Structured → Tabular data like patient demographics, lab results
   - Unstructured → Clinical notes, medical images, free-text observations
2. **Q:** How can data analysis support clinical decision-making?
   A: Identify trends, high-risk patients, disease patterns, and resource needs.
3. **Q:** What types of machine learning are used in healthcare?
   A:
   - **Supervised** → Regression, classification (predict disease, treatment outcomes)
   - **Unsupervised** → Clustering (group similar patients)
   - **Reinforcement learning** → Personalized treatment optimization

---

Experiment-Specific Questions

4. **Q:** Why do we use one-hot encoding for categorical variables?
   A: Converts categories into numeric form for algorithms like k-Means.
5. **Q:** What is k-Means clustering, and why use it?
   A: Groups similar data points into clusters based on feature similarity. Used to identify patterns in healthcare data.
6. **Q:** How do you choose the number of clusters in k-Means?
   A: Using the **Elbow Method**, plot distortions vs number of clusters and select the "elbow" point.
7. **Q:** What is PCA, and why is it applied here?
   A: PCA reduces dimensionality while retaining variance, making high-dimensional data easier to visualize.
8. **Q:** How are Folium maps useful?
   A: Visualize patient distributions and clusters geographically for resource planning.
9. **Q:** How do you interpret cluster analysis results in healthcare?
   A: Each cluster represents cities/patients with similar conditions or patterns, aiding targeted interventions.
10. **Q:** Why merge `patients.csv` and `conditions.csv`?
    A: To have a single integrated dataset linking patients to their medical conditions for analysis.

Advanced / Extra Questions

11. **Q:** How would you handle missing or NaN values before clustering?
    A: Drop, fill with mean/mode, or use imputation techniques depending on the feature.

12. **Q:** Why visualize data with Plotly or Folium instead of only Matplotlib/Seaborn?
    A: Plotly and Folium allow **interactive visualization** and geospatial mapping, making insights easier to explore.
13. **Q:** What insights can clustering and PCA provide to healthcare administrators?
    A: Identify high-risk areas, prevalent conditions, allocate resources, and plan interventions effectively.
14. **Q:** How can this experiment be extended?
    A: Include predictive modeling (disease prediction), real-time dashboards, or correlation with socioeconomic factors.

Experiment No.: 03

Title: To Implement Biomedical Image Segmentation

What We Do in This Experiment

1. **Introduction**
   - Study image processing and segmentation techniques applied to biomedical images.
   - Understand medical image analysis for healthcare applications.
2. **Setup and Preprocessing**
   - Install necessary Python libraries (`bebi103`, `iqplot`, `scikit-image`, `bokeh`)
   - Load biomedical images from dataset (`.tif` files)
   - Store interpixel distance for accurate measurement in microns
3. **Viewing Images**
   - Visualize images using **Bokeh** and **bebi103**
   - Apply different colormaps (gray, magma, viridis, turbo) for clarity
   - Compare zoomed regions using linked x and y ranges
4. **Image Analysis**
   - Plot intensity distribution using `iqplot.spike()`
   - Determine threshold manually by eye for segmentation
5. **Image Segmentation**
   - Generate binary (black & white) images based on threshold
   - Highlight segmented regions (e.g., bacteria) by stacking grayscale into RGB images
6. **Filtering and Morphology**
   - Apply **median filter** using structuring elements (`square(3)`) to reduce noise
   - Re-visualize filtered images and intensity distribution
7. **Otsu's Thresholding**
   - Compute optimal threshold using **Otsu's method**
   - Compare with manual threshold
8. **Compute Quantitative Metrics**
   - Calculate **bacterial area in pixels**
   - Convert to **micron²** using interpixel distance
9. **Final Outputs**
   - Segmented and filtered images
   - Bacterial area measurement
   - Comparison of manual vs automated thresholding

Practical Application:

- Biomedical image segmentation is critical for **diagnosis**, **quantifying biological structures**, and **analyzing medical images** in healthcare.

Likely Viva Questions and Answers

Basic Conceptual Questions

1. **Q:** What is the importance of image processing in healthcare?
   A: Helps visualize, analyze, and quantify medical images for diagnosis, treatment planning, and research.
2. **Q:** Define image segmentation. Why is it important in healthcare?
   A: Segmentation separates objects or regions of interest in an image. It is crucial for measuring tissues, identifying anomalies, or isolating specific cells.
3. **Q:** Name common segmentation techniques.
   A: Thresholding, Otsu's method, edge detection, region growing, clustering, deep learning-based segmentation.

Experiment-Specific Questions

4. **Q:** Why do we use median filtering?
   A: To remove noise from images while preserving edges, improving segmentation accuracy.
5. **Q:** What is Otsu's method?
   A: An automated thresholding technique that minimizes intra-class variance to segment images.
6. **Q:** How is the bacterial area calculated?
   A: Count pixels above threshold → multiply by square of interpixel distance → gives area in µm².
7. **Q:** Why do we stack grayscale images into RGB?
   A: To highlight segmented regions in color for better visualization.
8. **Q:** Why use linked x and y ranges in visualization?
   A: To ensure zooming/panning shows the same region across multiple plots for comparison.
9. **Q:** What libraries were used and why?
   A:
   - `skimage` → Image processing and filtering
   - `bebi103` → Rendering and interactive visualization
   - `iqplot` → Intensity distribution plotting
   - `bokeh` → Interactive plots
10. **Q:** How can image segmentation help healthcare professionals?
    A: Identify cell structures, measure tissue or bacterial area, detect anomalies, and support automated diagnostics.

Advanced / Extra Questions

11. **Q:** How would you handle overlapping objects in segmentation?
    A: Use watershed algorithm or advanced deep learning segmentation methods.

12. **Q:** How do you evaluate segmentation accuracy?
    A: Compare with ground truth using metrics like Dice coefficient, Jaccard index, or pixel accuracy.
13. **Q:** Why is interpixel distance important?
    A: Converts pixel measurements to real-world units (microns) for meaningful analysis.

Experiment No.: 04

Title: To Perform Biomedical Image Analysis Using CNN

What We Do in This Experiment

1. **Objective:**
   The goal of this experiment is to use **Convolutional Neural Networks (CNNs)** to classify **X-ray images** into **Normal** and **Pneumonia** cases, demonstrating the use of deep learning in healthcare image analysis.
2. **Dataset Handling:**
   - The dataset (`pneumonia-xray-images`) is downloaded from **Kaggle** using the Kaggle API.
   - Images are extracted and divided into **training**, **testing**, and **validation** sets.
3. **Data Preprocessing:**
   - Images are **rescaled (1/255)** for normalization.
   - **Image augmentation** (rotation, zoom, shift, shear, brightness changes) is used to improve model generalization.
   - Images are resized to **500x500** and converted to **grayscale**.
4. **CNN Model Architecture:**
   - Model is created using **Sequential API** in TensorFlow Keras.
   - Layers used:
     - **Convolutional layers (Conv2D)** → feature extraction
     - **MaxPooling2D** → reduces spatial size
     - **Flatten** → converts feature maps to 1D
     - **Dense layers** → final classification
   - **Activation functions:** ReLU for hidden layers, Sigmoid for output layer.
   - **Loss Function:** Binary Cross-Entropy (since it's a binary classification).
   - **Optimizer:** Adam
   - **Metrics:** Accuracy
5. **Model Training and Optimization:**
   - Trained for **3 epochs** using training and validation data.
   - Used **EarlyStopping** and **ReduceLROnPlateau** callbacks to avoid overfitting.
   - **Class weights** were applied to handle imbalance between "Normal" and "Pneumonia" images.
6. **Model Evaluation:**
   - Model is evaluated on the **test dataset** to compute accuracy.
   - Predictions are generated and visualized using:
     - **Confusion Matrix**
     - **Classification Report (Precision, Recall, F1-score)**
     - **Visualization of sample predictions** with actual labels.
7. **Output:**
   - Confusion matrix showing correct and incorrect classifications.
   - Classification report summarizing model performance.
   - Visual plots showing the model's confidence for each image.

8. **Practical Application:**
   This CNN-based image analysis helps doctors automatically detect **Pneumonia** from chest X-rays, supporting faster and more accurate diagnosis in healthcare.

Viva Questions and Answers

Basic Questions

1. **Q:** What is a CNN and why is it used for image analysis?
   A: CNN (Convolutional Neural Network) is a deep learning algorithm that captures spatial features from images through convolutional and pooling operations. It's ideal for tasks like classification, segmentation, and object detection.
2. **Q:** How does CNN differ from traditional image processing?
   A: Traditional methods rely on manual feature extraction, while CNNs automatically learn features directly from pixel data using multiple layers.
3. **Q:** What are the main layers in a CNN?
   A:
   - **Convolutional Layer:** Extracts local features
   - **Pooling Layer:** Reduces dimensionality
   - **Flatten Layer:** Converts data into 1D
   - **Fully Connected Layer (Dense):** Performs final classification
4. **Q:** Why do we use ReLU activation?
   A: ReLU (Rectified Linear Unit) introduces non-linearity and helps the model learn complex patterns efficiently.
5. **Q:** What is the purpose of the Sigmoid activation function?
   A: It maps the output to a range between 0 and 1, suitable for binary classification (Normal vs Pneumonia).

Experiment-Specific Questions

6. **Q:** Why did we normalize the image data (rescale 1/255)?
   A: To standardize pixel intensity values between 0 and 1, improving model training stability and convergence.
7. **Q:** What is ImageDataGenerator and why is it used?
   A: It is used for **data augmentation** — generating new variations of images (rotated, flipped, zoomed) to prevent overfitting and improve model generalization.
8. **Q:** What is the role of EarlyStopping and ReduceLROnPlateau?
   A:
   - **EarlyStopping:** Stops training when validation loss stops improving.
   - **ReduceLROnPlateau:** Reduces learning rate when progress stalls, helping fine-tune the model.
9. **Q:** What is the loss function used here and why?
   A: Binary Cross-Entropy, because the task is binary classification (Normal vs Pneumonia).
10. **Q:** What metrics are used to evaluate the model?
    A: Accuracy, Confusion Matrix, Precision, Recall, and F1-score.

11. **Q:** What is a confusion matrix?
    A: It's a table that shows the number of correct and incorrect predictions for each class — helping analyze performance.
12. **Q:** Why are class weights used?
    A: To handle imbalance between classes by giving higher weight to minority class samples.

Advanced / Extra Questions

13. **Q:** What is overfitting and how can we reduce it?
    A: Overfitting is when a model performs well on training data but poorly on new data. It can be reduced using data augmentation, dropout, regularization, and early stopping.
14. **Q:** What are convolutional filters?
    A: Small matrices that slide over the image to extract patterns like edges, corners, or textures.
15. **Q:** Why use grayscale images instead of RGB for X-rays?
    A: X-rays contain intensity information only; color channels are unnecessary and increase computational load.
16. **Q:** What is the optimizer used and its function?
    A: Adam — it adjusts learning rate adaptively for faster convergence.
17. **Q:** What does model.evaluate() return?
    A: It returns loss and accuracy values of the trained model on test data.

Experiment No.: 05

Title:

To Apply Text Analytics to Extract Medical Insights from Clinical Text Data

---

What We Do in This Experiment

Objective:

The goal of this experiment is to apply **text analytics** techniques on **clinical text data** to extract medical insights and classify medical records based on their specialties using Natural Language Processing (NLP) and Machine Learning algorithms.

Dataset Handling:

- The dataset **`mtsamples.csv`** is used, which contains **clinical transcriptions** and their associated **medical specialties**.
- The data is loaded using **Pandas** and basic inspection (`head()`, `columns()`) is performed.
- Missing or null values in the `transcription` column are removed to ensure data quality.

Data Exploration:

- Data is grouped by **medical_specialty** to observe category distribution.
- Categories with fewer than 50 records are filtered out to keep only major specialties.
- The category distribution is visualized using **Seaborn count plots** for better understanding.

Text Preprocessing:

1. **Tokenization:**
   - Each transcription is split into words and sentences using NLTK's `word_tokenize()` and `sent_tokenize()` functions.
2. **Lemmatization:**
   - Words are converted to their base forms using `WordNetLemmatizer()` to reduce redundancy.
3. **Text Cleaning:**

    - Special characters, digits, and punctuation are removed using regex and Python string translation.
    - Text is converted to lowercase for uniformity.

Feature Extraction:

- Used **TF-IDF (Term Frequency–Inverse Document Frequency)** Vectorizer from scikit-learn to convert text into numerical feature vectors.
- Extracted top 1000 features (unigrams, bigrams, and trigrams) representing important medical terms.

Dimensionality Reduction & Visualization:

1. **t-SNE (t-distributed Stochastic Neighbor Embedding):**
    - Used to visualize high-dimensional TF-IDF data in a 2D space for better understanding of clusters among medical specialties.
2. **PCA (Principal Component Analysis):**
    - Applied to reduce feature dimensions while retaining 95% of the data variance before model training.

Model Training and Evaluation:

1. **Train-Test Split:**
    - Data divided into **training (75%)** and **testing (25%)** sets with stratification on labels.
2. **Model Used:**
    - **Logistic Regression** classifier with **elastic net regularization** (L1 + L2) for multi-class classification.
3. **Evaluation Metrics:**
    - Confusion Matrix visualized using Seaborn heatmap.
    - **Classification Report** generated showing Precision, Recall, and F1-score for each specialty.

Output:

- Visualization of medical specialties distribution.
- Confusion matrix showing correct vs. incorrect specialty predictions.
- Classification report summarizing model performance across multiple medical domains.

Practical Application:

This experiment demonstrates how **text analytics and NLP** can be used in healthcare to automatically **classify medical records** or **extract insights** from clinical notes, helping doctors and researchers in efficient data retrieval, disease pattern detection, and decision-making.

Viva Questions and Answers

Basic Questions

Q1. What is text analytics?
A: Text analytics is the process of extracting meaningful insights and patterns from textual data using techniques like tokenization, lemmatization, and vectorization.

Q2. Why is preprocessing important in text analytics?
A: Preprocessing removes noise, standardizes text, and converts it into a format suitable for machine learning models, improving accuracy and efficiency.

Q3. What is TF-IDF and why is it used?
A: TF-IDF (Term Frequency–Inverse Document Frequency) measures how important a word is in a document relative to a corpus, helping models focus on significant words rather than common ones.

Q4. What is lemmatization and how does it differ from stemming?
A: Lemmatization reduces words to their base form using a dictionary (e.g., “running” → “run”), while stemming just removes word endings (e.g., “running” → “runn”) without considering meaning.

Q5. What is tokenization?
A: Tokenization splits text into smaller units such as words or sentences, which are used for further analysis.

Experiment-Specific Questions

Q6. Why do we use PCA in this experiment?
A: PCA helps reduce feature dimensions while keeping most information intact, improving computational efficiency and visualization.

Q7. What is the purpose of t-SNE visualization?
A: t-SNE projects high-dimensional data into 2D/3D space, allowing us to visualize and understand the clustering of medical specialties.

Q8. Why is Logistic Regression used here?
A: Logistic Regression performs well for multi-class text classification problems and is interpretable, making it suitable for healthcare analytics.

Q9. What are stop words and why are they removed?
A: Stop words (like “is”, “the”, “and”) are common words that don’t add meaning to analysis, so removing them helps focus on significant terms.

Q10. What does the confusion matrix show?
A: It shows the comparison between predicted and actual classes, highlighting the correct and incorrect classifications.

Advanced / Extra Questions

Q11. How can imbalanced datasets affect model performance?
A: Imbalance causes the model to favor majority classes; techniques like SMOTE or class weighting can help balance them.

Q12. Why is it important to remove punctuation and digits from text?
A: They generally don’t contribute to semantic meaning in clinical texts and can introduce noise in vectorization.

Q13. What is the difference between Bag of Words and TF-IDF?
A: Bag of Words counts word occurrences, while TF-IDF also considers how rare or important a word is across documents.

Q14. What is Elastic Net Regularization in Logistic Regression?
A: It combines L1 (Lasso) and L2 (Ridge) regularization to prevent overfitting and handle correlated features effectively.

Q15. Give one real-world application of clinical text analytics.
A: Extracting disease symptoms, drug interactions, or treatment outcomes from Electronic Health Records (EHRs) for healthcare decision support.

Experiment No.: 06

Title:

To Diagnose Disease Risk from Patient Data

What We Do in This Experiment

Objective:

The objective of this experiment is to use **data science and machine learning techniques** to predict **disease risk** from patient health data, based on features like age, weight, blood pressure, diabetes, and smoking habits.

Dataset Handling:

- A **synthetic dataset** of 10,000 patient records is generated using Python's `random` module.
- Each record includes attributes such as:
  - Age
  - Height
  - Weight
  - Systolic and Diastolic Blood Pressure
  - Diabetes (0 = No, 1 = Yes)
  - Smoker (0 = No, 1 = Yes)
  - Heart Disease (0 = No, 1 = Yes)
  - Diagnosis (Healthy / Risk)
- The `diagnosis` label is generated based on whether the patient has diabetes, smokes, and has heart disease simultaneously — these increase the disease risk.

Data Preprocessing:

1. **Encoding Categorical Variables:**
   - Binary variables (like diabetes, smoker, and heart disease) are converted into dummy variables using `pd.get_dummies()`.
2. **Feature Splitting:**
   - The dataset is divided into **independent features (X)** and **target labels (y)**.
3. **Train-Test Split:**
   - Data is split into **80% training** and **20% testing** using `train_test_split()` to evaluate model performance.
4. **Feature Scaling:**
   - Features are standardized using **StandardScaler** to normalize data for machine learning models.

Model Building and Training:

- The **Random Forest Classifier** from scikit-learn is used to build the prediction model.
- Random Forest is an **ensemble learning algorithm** that combines multiple decision trees to improve prediction accuracy and control overfitting.
- The model is trained on the scaled training dataset (`X_train_scaled`, `y_train`).

Model Evaluation:

- Predictions are made on the test set (`X_test_scaled`).
- Model accuracy is calculated using `accuracy_score()`.
- In this experiment, the model achieved **100% accuracy (Accuracy: 1.00)** on the test data, showing that the generated dataset is perfectly separable by the features used.

Disease Risk Prediction for New Patient:

- A new patient's data (e.g., Age = 45, Height = 165 cm, Weight = 70 kg, BP = 120/80, Diabetic = Yes, Smoker = No, Heart Disease = No) is provided as input.
- The data is scaled using the same `StandardScaler` and passed to the model for prediction.
- The model outputs **"Healthy"** or **"Risk"** depending on the patient's attributes.

Output:

- Columns printed:
```
['age', 'height', 'weight', 'systolic_bp', 'diastolic_bp',
'diabetes', 'smoker', 'heart_disease', 'diagnosis']
```
- Model Accuracy: **1.00**
- Predicted Disease Risk for new patient: **"Healthy" or "Risk"** (depending on input).

Practical Application:

This experiment shows how **machine learning can predict disease risks** based on patient parameters, which can help in **early diagnosis, preventive care**, and **decision support systems** in healthcare.

Viva Questions and Answers

Basic Questions

Q1. What is the main goal of this experiment?
A: To predict whether a patient is at disease risk or healthy using machine learning techniques based on health-related features.

Q2. What type of dataset is used here?
A: A **synthetic dataset** created programmatically using random values to simulate patient health data.

Q3. What is feature scaling and why is it important?
A: Feature scaling standardizes values across all features, preventing variables with larger ranges from dominating the learning process.

Q4. What is the role of `train_test_split()`?
A: It divides the dataset into training and testing subsets to evaluate model performance on unseen data.

Q5. What is the target variable in this experiment?
A: The **diagnosis** column, which indicates whether a patient is "healthy" or at "risk."

Experiment-Specific Questions

Q6. Why did we use Random Forest Classifier?
A: Random Forest is robust, handles non-linear relationships, reduces overfitting, and works well with categorical and continuous features.

Q7. What does an accuracy of 1.00 indicate?
A: It means the model perfectly classified all test samples — likely due to the simplicity and clear pattern in the synthetic data.

Q8. What are dummy variables and why are they created?
A: Dummy variables convert categorical features into numeric binary columns so the model can process them.

Q9. What factors are considered in predicting disease risk here?
A: Factors like diabetes, smoking habits, heart disease, and blood pressure are used to determine the risk.

Q10. How does the model predict new patient data?
A: The input patient data is scaled and passed into the trained model, which outputs the predicted class ("Healthy" or "Risk").

Advanced / Extra Questions

Q11. What is Random Forest and how does it work?
A: Random Forest is an **ensemble algorithm** that builds multiple decision trees and combines their outputs to improve accuracy and stability.

Q12. What is the difference between overfitting and underfitting?
A:

- **Overfitting:** Model performs well on training data but poorly on new data.
- **Underfitting:** Model fails to learn patterns from the data.

Q13. Why is scaling necessary before training?
A: Scaling ensures all features contribute equally to distance-based models and improves training convergence for algorithms like Random Forest.

Q14. What is feature selection and why is it important?
A: Feature selection identifies the most important features for prediction, improving accuracy and reducing complexity.

Q15. Mention real-life use cases of disease risk prediction models.
A: Used in predicting risks of diabetes, heart disease, cancer, and stroke based on patient health and lifestyle data.

Experiment No.: 07

Title: To Implement Social Media Analytics for Outbreak Prediction

---

What We Do in This Experiment

In this experiment, we use **social media data (tweets)** to perform **sentiment analysis** related to a health outbreak (like COVID-19). The goal is to understand **public emotions and discussions** that could help predict or track an outbreak trend.

We follow these key steps:

1. **Import necessary libraries** – pandas, re, seaborn, matplotlib, wordcloud, and TextBlob.
2. **Load dataset** – Read a CSV file containing tweets related to COVID-19.
3. **Data Cleaning** – Remove usernames, links, punctuations, and special characters using regular expressions.
4. **Sentiment Analysis** –
   - Use **TextBlob** to calculate the **polarity** of each tweet.
   - Classify tweets as **Positive**, **Negative**, or **Neutral**.
5. **Visualization** –
   - Use **count plots** to visualize the sentiment distribution.
   - Generate **word clouds** to display frequently used words in each sentiment category.
6. **Interpretation** – Analyze which sentiment dominates and how public opinion trends can indicate outbreak awareness or panic levels.

Output Summary

- Total tweets analyzed.
- Count of **Positive**, **Negative**, and **Neutral** tweets.
- A **bar chart** (countplot) showing sentiment distribution.
- **Word clouds** for:
  - All tweets combined
  - Positive tweets
  - Negative tweets

Most Possible Viva Questions and Answers

1. What is the purpose of this experiment?
To analyze social media data (tweets) using sentiment analysis and identify public opinions that can help in outbreak prediction.

2. What is sentiment analysis?
Sentiment analysis is the process of determining whether a text expresses a **positive, negative, or neutral** emotion.

3. Which library is used for sentiment analysis here?
We use **TextBlob**, a Python library that provides sentiment polarity and subjectivity scores.

4. What is the role of text preprocessing in this experiment?
Preprocessing removes noise like URLs, mentions (@user), hashtags, and symbols to ensure clean text for accurate sentiment analysis.

5. How does TextBlob determine sentiment?
It calculates the **polarity score** of text:

- 0 → Positive
- =0 → Neutral
- <0 → Negative

6. What is the significance of the word cloud?
A word cloud visually shows the **most frequent words** in tweets, helping to identify trending terms and topics in discussions.

7. What does the count plot represent?
It shows how many tweets belong to each sentiment category (Positive, Negative, Neutral).

8. How can social media analytics help in outbreak prediction?
By monitoring sentiment and keyword trends, authorities can detect early signs of fear, awareness, or misinformation related to disease outbreaks.

9. What is the function of regular expressions (re module) in this code?
It is used to clean tweets by removing unwanted characters, special symbols, and hyperlinks.

10. What is the difference between text mining and NLP?

- **Text Mining** extracts useful patterns or information from text data.
- **NLP (Natural Language Processing)** helps the computer understand, interpret, and process human language.

---

11. What is Named Entity Recognition (NER)?
NER identifies and classifies entities like disease names, places, or organizations from text data—useful for outbreak tracking.

---

12. What are some applications of social media analytics in healthcare?

- Outbreak prediction and monitoring
- Identifying misinformation
- Understanding public sentiment toward health policies or vaccines

---

13. Which visualization libraries are used?

- **Seaborn** → for sentiment count plot
- **Matplotlib** → for pie charts and displaying word clouds

---

14. Why do we convert tweets to lowercase?
To maintain uniformity and avoid treating the same word in different cases (like "Covid" and "covid") as separate terms.

---

15. How can you improve this experiment?
By using advanced NLP models like **BERT** or **VADER** for more accurate sentiment prediction and by including **geotags** for outbreak localization.

Experiment No.: 08

Title: To perform visual analytics for healthcare data

What We Do in This Experiment

In this experiment, we perform **visual analytics and predictive modeling** using a healthcare dataset related to **stroke prediction**.

Here's what is done step-by-step:

1. **Dataset Loading & Preprocessing**
   - The dataset `healthcare-dataset-stroke-data.csv` is imported using pandas.
   - Missing BMI values are handled using a **Decision Tree Regressor** model to predict and fill them.
   - Categorical columns like `gender`, `Residence_type`, and `work_type` are **encoded** numerically for model compatibility.
2. **Data Visualization**
   - Various visualization techniques are applied using **Matplotlib** and **Seaborn**:
     - **KDE plots** to show the distribution of numeric variables like age, BMI, and glucose levels.
     - **Comparison plots** to differentiate stroke and non-stroke patients.
     - **Line plot** showing **increasing stroke risk with age**.
     - **Waffle chart** showing proportion of stroke cases in the dataset.
     - **Bar charts** and **density plots** to analyze relationships with gender, smoking status, hypertension, work type, and heart disease.
3. **Handling Class Imbalance**
   - The dataset is imbalanced (only ~5% stroke cases).
   - Used **SMOTE (Synthetic Minority Oversampling Technique)** to balance the dataset.
4. **Model Building and Evaluation**
   - Three models are trained:
     - **Random Forest Classifier**
     - **Support Vector Machine (SVM)**
     - **Logistic Regression**
   - Models are trained using **pipelines** with scaling.
   - Cross-validation (`cross_val_score`) is used to compute F1 scores.
   - Evaluation metrics used: **Accuracy**, **Precision**, **Recall**, and **F1-Score**.
   - **Random Forest** performed best with mean F1 score ≈ **0.93**.
5. **Hyperparameter Tuning**
   - Used **GridSearchCV** to tune parameters for Random Forest and Logistic Regression for better performance.

Outcome / Conclusion

- Learned how to **visualize and interpret healthcare data** effectively.
- Identified key factors influencing stroke risk such as **age, BMI, glucose levels, heart disease**, and **hypertension**.
- Understood how to handle missing data, balance datasets, and evaluate ML models using F1-score and confusion matrices.
- Gained proficiency in building **data analytics pipelines** using Scikit-learn, Seaborn, and Matplotlib.

Most Possible Viva Questions & Answers

Basic Questions

1. **What is the aim of this experiment?**
   → To perform visual analytics on healthcare data and apply different visualization and machine learning techniques for stroke prediction.
2. **What dataset did you use?**
   → The "Healthcare Stroke Prediction" dataset which includes data like age, gender, BMI, glucose levels, smoking status, heart disease, and stroke occurrence.
3. **Which Python libraries were used?**
   → Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Imbalanced-learn (SMOTE), and PyWaffle.
4. **What is data visualization and why is it important in healthcare?**
   → Data visualization helps represent complex data graphically to identify patterns, trends, and correlations — crucial for medical decision-making.

Data Preprocessing Questions

5. **How were missing values handled?**
   → Missing BMI values were predicted and filled using a **Decision Tree Regressor** trained on age and gender.
6. **What is encoding and why is it used?**
   → Encoding converts categorical data (like gender or work type) into numerical form so that models can process them.
7. **What is SMOTE and why did you use it?**
   → SMOTE (Synthetic Minority Oversampling Technique) creates synthetic samples for minority classes to balance the dataset and improve model accuracy.

Model and Evaluation Questions

8. **Which models did you use for stroke prediction?**
   → Random Forest Classifier, SVM, and Logistic Regression.
9. **Which model performed best and why?**
   → Random Forest performed best with an F1-score of around **0.93** because it reduces overfitting and works well with complex datasets.

10. **What evaluation metrics were used?**
    → Accuracy, Precision, Recall, F1-score, and Confusion Matrix.
11. **Why is F1-score used instead of accuracy?**
    → Because the dataset is imbalanced — F1-score balances precision and recall to give a fair evaluation.
12. **What is cross-validation?**
    → Cross-validation tests model performance by splitting the dataset into multiple folds to ensure generalization.

Visualization & Interpretation Questions

13. **What type of visualization was used for numeric variables?**
    → KDE (Kernel Density Estimation) plots to show data distribution.
14. **What does the Waffle chart represent?**
    → It shows the proportion of people affected by stroke — roughly 1 in 20 in the dataset.
15. **Which features were found to be most important in stroke prediction?**
    → Age, BMI, average glucose level, hypertension, and heart disease.
16. **What insight did the age vs. stroke plot give?**
    → Stroke risk increases significantly with age.
17. **What is correlation in data visualization?**
    → It measures how one variable changes with respect to another — used to find relationships like glucose level vs stroke risk.

Advanced/Technical Questions

18. **Why use StandardScaler in the pipeline?**
    → To normalize data and bring all features to the same scale for better model convergence.
19. **What is hyperparameter tuning?**
    → The process of finding the best model parameters using techniques like GridSearchCV.
20. **Difference between Logistic Regression and Random Forest?**
    → Logistic Regression is a linear model; Random Forest is an ensemble of decision trees capable of handling non-linear data.
21. **What are the challenges in healthcare data visualization?**
    → Privacy concerns, data imbalance, missing values, and interpreting multidimensional data correctly.