EXPERIMENT-10

Student Name: Samuel **UID: 24MAI10018**

Branch: ME-AIML **Section: 24MAI-1**

Semester: 2nd **Date:**

Subject Name: ML Lab **Subject Code: 24CSH-667**

AIM - Case Study on application of AI (Early Disease Detection using AI).

Software Required –

- Python
- IDE

Theory –

What is Artificial Intelligence (AI)?

Artificial Intelligence (AI) is a branch of computer science focused on creating systems capable of performing tasks that normally require human intelligence. These tasks include learning, reasoning, problem-solving, perception, and natural language understanding.

What is Machine Learning (ML)?

Machine Learning is a subfield of AI that enables computers to learn patterns from data and make predictions or decisions without being explicitly programmed. ML involves algorithms that improve automatically through experience.

Types of Machine Learning:

- Supervised Learning: Learns from labelled data (e.g., regression, classification).
- Unsupervised Learning: Identifies patterns in unlabelled data (e.g., clustering, dimensionality reduction).
- Reinforcement Learning: Learns from interactions with an environment to maximize reward.

ML Workflow (Pipeline)

1. Data Collection: Gather historical or real-time data (e.g., sensors, logs).
2. Data Preprocessing: Clean and normalize data for analysis.
3. Feature Selection: Choose relevant variables for model training.

4. Model Selection: Choose algorithm (Random Forest, SVM, etc.).
5. Model Training: Feed data to algorithm to learn relationships.
6. Model Evaluation: Use metrics like accuracy, precision, recall.
7. Deployment: Integrate into real-time system or dashboard.
8. Monitoring: Track performance and retrain as needed.

Application: Early Disease Detection using AI

Problem:

In healthcare, delayed or inaccurate diagnosis of diseases such as breast cancer can lead to life-threatening consequences, prolonged treatment, and higher medical costs. Traditional diagnostic methods often rely heavily on manual interpretation and are prone to human error.

AI-Based Solution:

Machine Learning (ML) is increasingly being used to analyze medical data (like cell nucleus measurements, tissue density, and more) to predict the likelihood of diseases, such as whether a tumor is malignant (cancerous) or benign (non-cancerous).

How It Works:

- Medical Data Collection: Patient test results (e.g., biopsies, imaging, measurements) are collected.
- Feature Extraction: Key measurements (e.g., mean radius, texture, perimeter) are extracted from cells.
- Model Training: ML algorithms are trained on labelled patient records to learn patterns between features and outcomes.
- Disease Prediction: The trained model predicts whether new patient data indicates a malignant or benign tumor.
- Decision Support: Doctors use these predictions as decision-support tools for faster and more accurate diagnoses.

Real-World Applications:

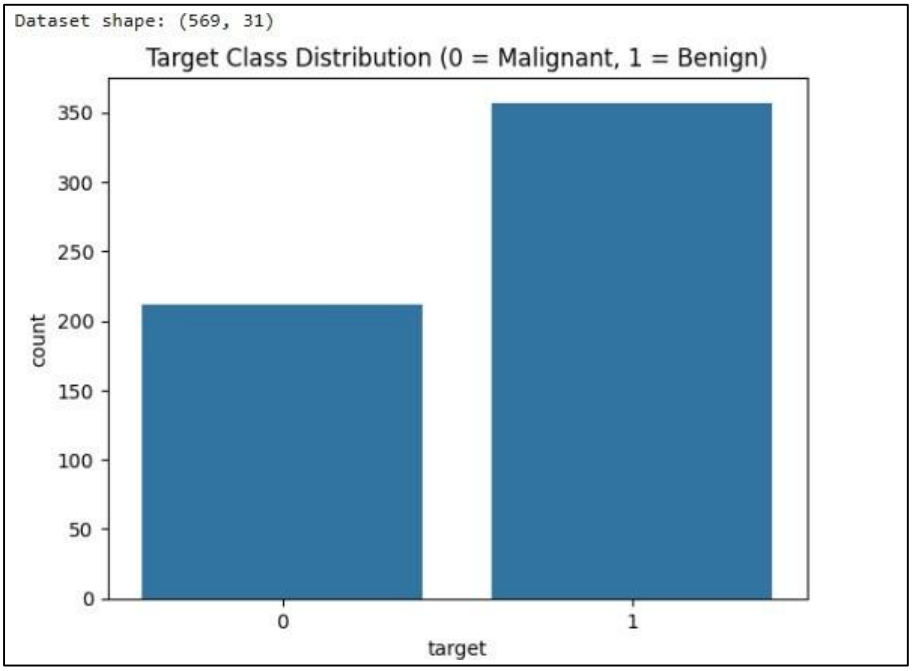
- Radiology: AI tools analyze mammograms, CT scans, and MRIs to detect abnormalities.
- Pathology: ML assists in detecting cancerous cells in tissue samples.
- Preventive Healthcare: Early prediction models help flag high-risk patients before symptoms worsen.
- Clinical Decision Support Systems (CDSS): AI assists doctors by highlighting potential diagnosis or treatment pathways.

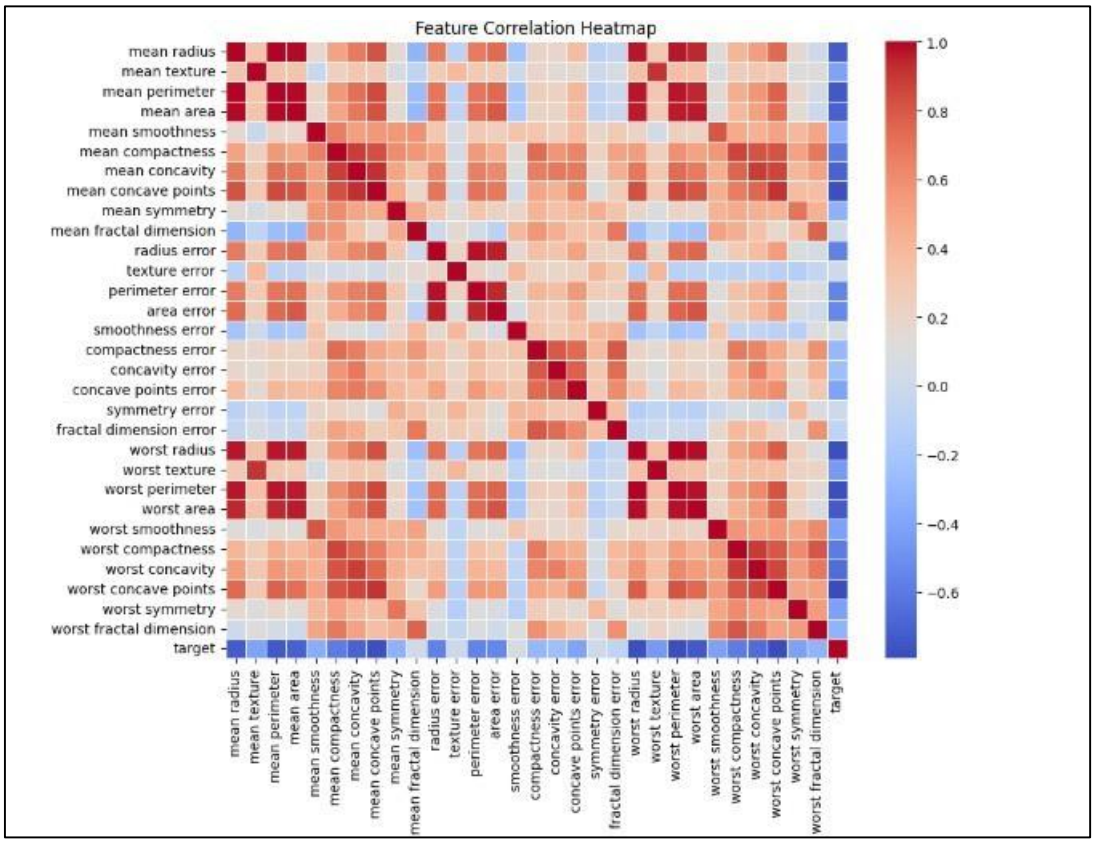
CODE –

```
# Install required libraries
!pip install scikit-learn matplotlib seaborn

# Import necessary modules
from sklearn.datasets import load_breast_cancer
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Load the built-in breast cancer dataset
data = load_breast_cancer()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

# Show dataset shape and preview
print("Dataset shape:", df.shape)
df.head()

# Visualizing target distribution
sns.countplot(x='target', data=df)
plt.title("Target Class Distribution (0 = Malignant, 1 = Benign)")
plt.show()

# Correlation heatmap
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), cmap='coolwarm', linewidths=0.5)
```

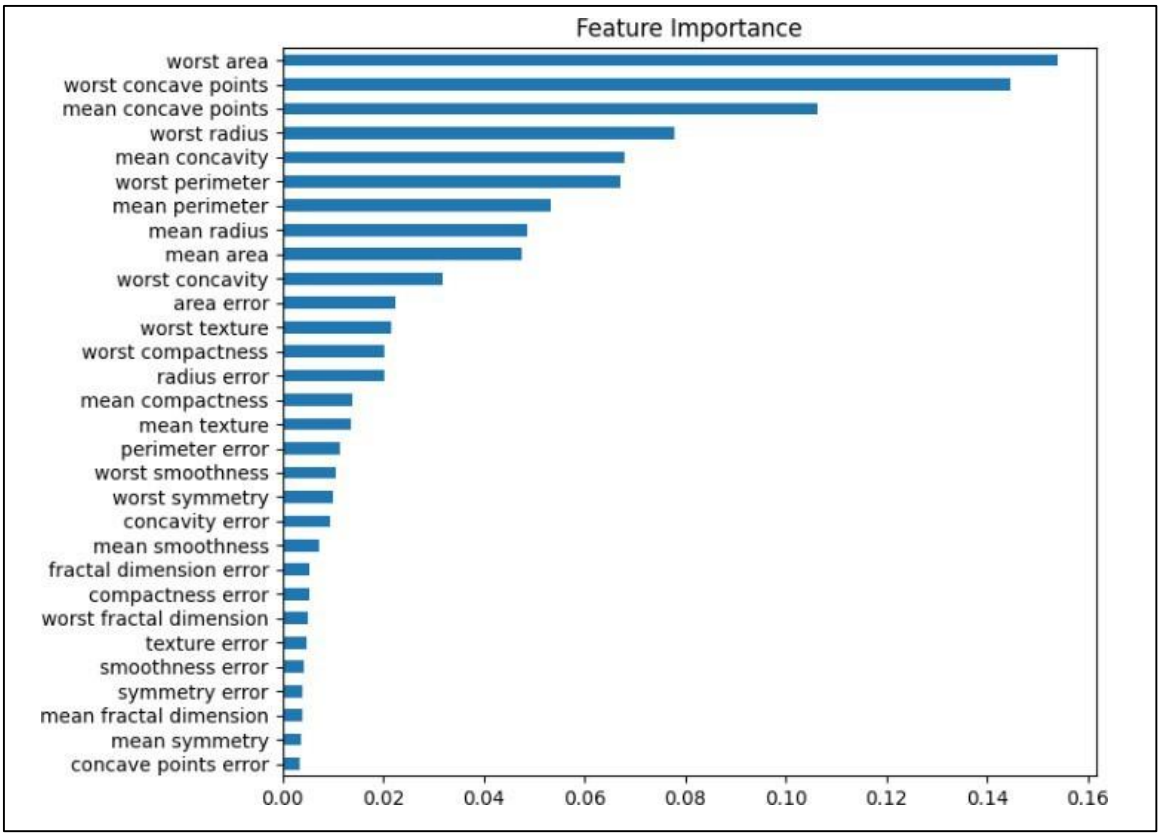
```
plt.title("Feature Correlation Heatmap")
plt.show()

# Splitting features and target
X = df.drop('target', axis=1)
y = df['target']

# Splitting dataset
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Training a Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
print("Classification Report:\n", classification_report(y_test,
y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Feature Importance
feature_importance = pd.Series(model.feature_importances_,
index=X.columns)
feature_importance.sort_values().plot(kind='barh', figsize=(8,6),
title="Feature Importance")
plt.tight_layout()
plt.show()
```

Output –


Dataset shape: (569, 31)
Target Class Distribution (0 = Malignant, 1 = Benign)
count
target
0
1
0
50
100
150
200
250
300
350


Feature Correlation Heatmap
mean radius
mean texture
mean perimeter
mean area
mean smoothness
mean compactness
mean concavity
mean concave points
mean symmetry
mean fractal dimension
radius error
texture error
perimeter error
area error
smoothness error
compactness error
concavity error
concave points error
symmetry error
fractal dimension error
worst radius
worst texture
worst perimeter
worst area
worst smoothness
worst compactness
worst concavity
worst concave points
worst symmetry
worst fractal dimension
target
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114

Confusion Matrix:
 [[40  3]
 [ 1 70]]
```



Learning Outcomes -

- Understand core concepts of AI and ML.
- Differentiate between types of machine learning.
- Apply Random Forest for classification tasks.
- Analyze sensor data for predictive maintenance.