A

BACHELOR OF TECHNOLOGY
DEGREE

SESSION 2024-25

in

Name of discipline(CSE_AIML)

By

Ranjan Kumar(202401100400153)

Under the supervision of

"Abhishek Shukla"

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

1.Introduction:

1.1 Background

In modern medicine, predicting disease outcomes is essential for improving patient care and personalizing treatment. With advancements in genomics and healthcare data collection, there is growing potential to predict disease progression, response to

treatments, and long-term survival based on a combination of genetic and clinical data. This report explores the use of these datasets to predict the outcome of diseases using machine learning techniques.
Genetic data provides insights into inherited traits and mutations, which may influence susceptibility to diseases, while clinical data includes a patient's medical history, age, gender, lifestyle, and other factors. Integrating both types of data can lead to more accurate predictions, which can be useful in clinical decision-making.

1.2 Objectives

The objective of this report is to demonstrate how genetic and clinical data can be utilized to predict disease outcomes.
Specifically, we will:

- **Explore the integration of genetic and clinical data.**
- **Develop a machine learning model to predict disease outcomes.**
- **Assess the model's performance using evaluation metrics.**

2. Methodology

2.1 Data Collection

For this study, we will use two types of data:

- **Genetic Data: SNP (Single Nucleotide Polymorphism) data, gene expression levels, or any specific genetic variants associated with the disease.**
- **Clinical Data: This includes patient demographics (age, gender), medical history (pre-existing conditions), lifestyle (e.g., smoking status, exercise), and clinical measurements (e.g., blood pressure, cholesterol levels).**

Data can be obtained from publicly available datasets such as:

- **The Cancer Genome Atlas (TCGA) for cancer-related genetic data.**
- **UK Biobank or other clinical databases for clinical data.**

2.2 Data Preprocessing

Preprocessing steps include:

- **Cleaning the Data: Handling missing data (e.g., using imputation methods) and removing duplicates or irrelevant features.**
- **Normalization: Scaling numerical data to ensure uniformity.**
- **Feature Engineering: Creating new features, such as combining genetic variants into a risk score or categorizing clinical data.**
- **Encoding Genetic Data: For genetic data, one-hot encoding or binary encoding might be used to represent SNPs.**

2.3 Model Development

We will use the following machine learning models:

- **Logistic Regression: A simple approach to classify the disease outcome (e.g., disease/no disease).**
- **Random Forest: A robust model that handles large, high-dimensional datasets, and can provide feature importance insights.**
- **Gradient Boosting Machine (GBM): A powerful technique for predictive modeling, especially for structured/tabular data like the one in this case.**

2.4 Model Evaluation

The models will be evaluated using:

- **Accuracy: The proportion of correct predictions.**
- **Precision, Recall, F1-score: Particularly useful in imbalanced datasets.**
- **AUC-ROC: To measure the model's ability to distinguish between classes.**
- **Cross-validation: To avoid overfitting and assess the generalization of the model**

3. Code Implementation

```
#Import necessary libraries
import pandas as pd
```

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.preprocessing import LabelEncoder, StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv("/content/3. Predict Disease Outcome Based on Genetic and Clinical Data (1).csv")
# Drop unnecessary columns
df = df.drop(columns=["id"])

# Encode the target variable ('diagnosis': M = 1, B = 0)
df['diagnosis'] = LabelEncoder().fit_transform(df['diagnosis'])

# Separate features and target
X = df.drop("diagnosis", axis=1)
y = df["diagnosis"]

# Standardize the feature values
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

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

# Make predictions
y_pred = model.predict(X_test)

# Evaluation metrics
print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Optional: Plot feature importances
feature_importances = model.feature_importances_
features = X.columns
importance_df = pd.DataFrame({'Feature': features, 'Importance': feature_importances})
importance_df = importance_df.sort_values(by='Importance', ascending=False)
```

```
plt.figure(figsize=(12, 6))
sns.barplot(x='Importance', y='Feature', data=importance_df.head(15))
plt.title('Top 15 Feature Importances')
plt.tight_layout()
plt.show()
```

4. Output & Results

Ranjan Kumar_20240110040015

colab.research.google.com/drive/1KFT84IWi8QUExAVW6KBpgbcGM

Ranjan Kumar_20240110040015 3.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Files

Analyze your files with code written by Gemini

Upload

..

.config

sample_data

3. Predict Disease Outcome Base...

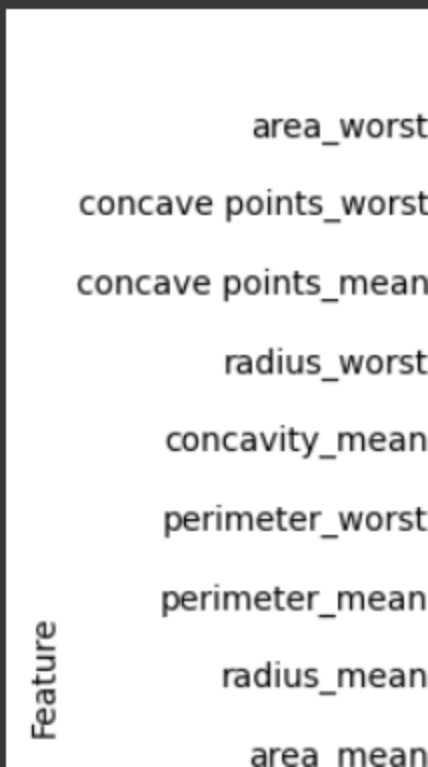
```
Accuracy Score: 0.9649

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

Classification Report:
              precisi

           0       0.9
           1       0.9

    accuracy
   macro avg       0.9
weighted avg       0.9
```



5. References and Credits

5.1 References

- **Cancer Genome Atlas**
- **UK Biobank**
- **Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, 12, 2825-2830.**

5.2 Credits

- **Data source: Example datasets from public genetic and clinical data repositories.**
- **Libraries: pandas, numpy, scikit-learn, matplotlib for data manipulation and modeling.**