

EARLY STAGE DETECTION OF OVARIAN CANCER

"Transforming Data Into Early Diagnoses"

Project Report

Submitted to - Dr. Sanchali Das

*Submitted By :
Aditi Singh - E23CSEU1484
Nischay Rawat - E23CSEU1485
Deep Bansal - E23CSEU1480*

INTRODUCTION:

The seventh most prevalent illness in women is ovarian cancer (OC), which carries a high lifetime risk. Low survival rates are caused by late-stage diagnosis and a lack of early indications, which highlights the critical need for efficient diagnostic instruments. Our goal is to provide comprehensive research of malignant epithelial ovarian tumours by integrating clinical data and biomarkers.

Crucial Biomarkers (RF Model)

- > CA125
- > HE4
- > NEU

These biomarkers represent the cutting edge of our research, each providing a distinct window into the complex molecular and clinical features of ovarian cancers. In addition to the well-known ovarian cancer marker CA125, HE4 and NEU complete a trio of biomarkers that together have the potential to improve the sensitivity and specificity of our diagnostic models.

METHODOLOGY:

We have taken 3 different datasets of 350 women and integrated it into one such that it has a wide variety of attributes to analyse the data and make appropriate predictions. The attributes of datasets are as follows:

```

Dataset 1: OC_Markers.csv
Age: Age of the individual.
Menopause: Menopausal status (1: Yes, 0: No).
CA19-9: Carbohydrate antigen 19-9 biomarker.
CA72-4: Carbohydrate antigen 72-4 biomarker.
AFP: Alpha-fetoprotein biomarker.
CA125: Carbohydrate antigen 125 biomarker.
HE4: Human epididymis protein 4 biomarker.
CEA: Carcinoembryonic antigen biomarker.
TYPE: Target variable indicating the type of ovarian tumor (Binary classification).

```

```

Dataset 2: OC_General_Chem.csv
Age: Age of the individual.
AG: Alpha-1 globulin.
ALB: Albumin.
ALP: Alkaline phosphatase.
ALT: Alanine transaminase.
AST: Aspartate transaminase.
BUN: Blood urea nitrogen.
Ca: Calcium.
CL: Chloride.
CO2CP: Carbon dioxide content.
CREA: Creatinine.
DBIL: Direct bilirubin.
GGT: Gamma-glutamyl transferase.
GLO: Globulin.
GLU: Glucose.
IBIL: Indirect bilirubin.
K: Potassium.
Mg: Magnesium.
Na: Sodium.
PHOS: Phosphate.
TBIL: Total bilirubin.
TP: Total protein.
UA: Uric acid.
TYPE: Target variable indicating the type of ovarian tumor (Binary classification).

```

```

Dataset 3: OC_Blood_Routine.csv
Age: Age of the individual.
MPV: Mean platelet volume.
BASO#: Basophils count.
BASO%: Basophils percentage.
EO#: Eosinophils count.
EO%: Eosinophils percentage.
MCH: Mean corpuscular hemoglobin.
RDW: Red cell distribution width.
PDW: Platelet distribution width.
HGB: Hemoglobin.
LYM#: Lymphocytes count.
LYM%: Lymphocytes percentage.
MONO#: Monocytes count.
MONO%: Monocytes percentage.
PLT: Platelet count.
NEU: Neutrophils count.
RBC: Red blood cell count.
PCT: Plateletcrit.
HCT: Hematocrit.
MCV: Mean corpuscular volume.
TYPE: Target variable indicating the type of ovarian tumor (Binary classification).

```

DATA PRE-PROCESSING:

Data preparation is a crucial first step in preparing raw datasets for analysis and model construction. It employs multiple techniques to correct missing values, enhance the quality of the data, and ensure that machine learning algorithms can use it. The following are some essential steps in data preprocessing:

- Data Integration
- Handling Missing Data
- Dealing with Duplicates
- Handling Outliers
- Data Scaling

```
[ ] combined_data = pd.concat([data1.set_index(['Age']),
                               oc_blood_routine.set_index(['Age']),
                               oc_general_chem.set_index(['Age'])],
                               axis=1).reset_index()

# Print the resulting combined dataset
print(combined_data)
```

1. This is how we have integrated 3 datasets, namely OC_Marker, OC_General_Chem and OC_Blood_Routine into 1 combined dataset and used this for the data analysis.
2. There are no NULL values in our final dataset.
3. Size of the combined_dataset:

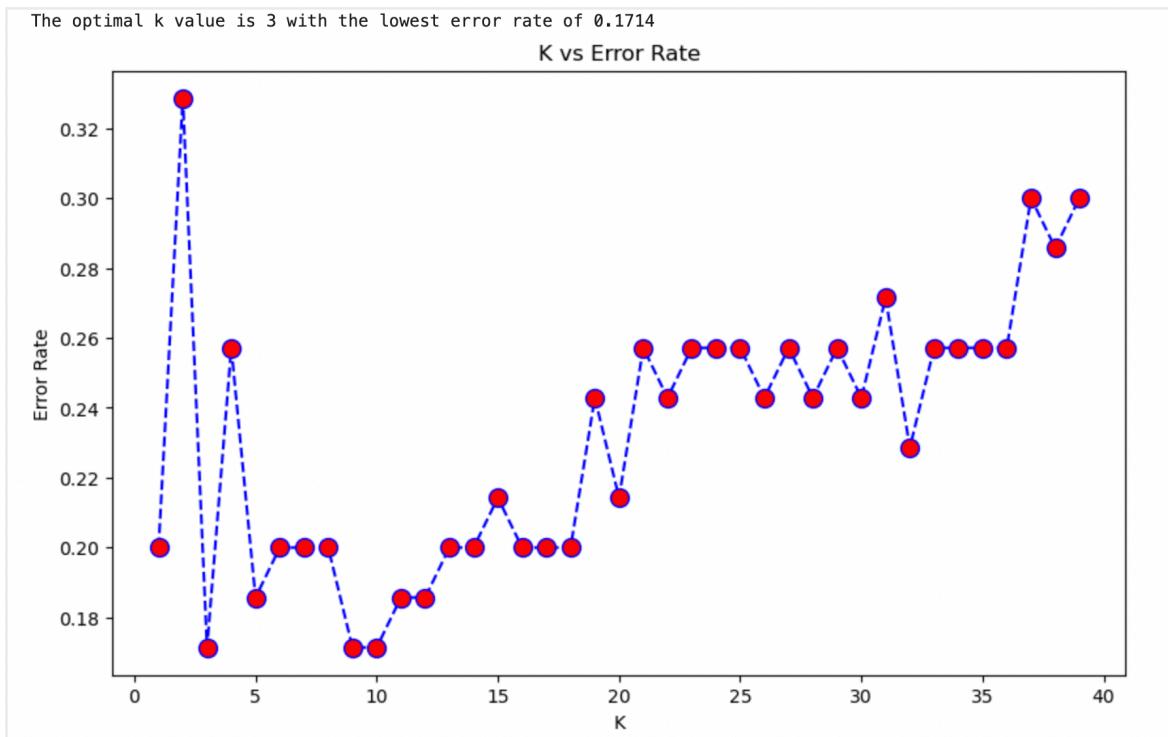
[349 rows x 51 columns]

MODEL DEVELOPMENT:

1) K-NEAREST NEIGHBORS(KNN) Classifier :

Implementation Steps

1. The implementation process began with training a basic KNN model using $n_neighbors=5$ and using a test-train split ($test_size=0.2$).
2. To optimize the model, an analysis was conducted by iterating through k values from 1 to 40, with a plot of error rate versus k used to identify the optimal choice.
3. Following this, hyperparameter tuning was performed using RandomizedSearchCV, exploring $n_neighbors$ from 1 to 10 to further refine the model and select the best k value for improved performance.



2) RANDOM- FOREST:

Implementation Steps

- **Best RF Model**

1. Initial Setup: A basic Random Forest Classifier was trained with default hyper-parameters and a train-test split of 80/20.
2. **Hyper-parameter Tuning:** GridSearchCV was used to optimize key hyper-parameters, such as n_estimators, max_depth, max_features, and criterion, through 5-fold cross-validation.
3. Model Evaluation: The best model was trained and evaluated on the test set using metrics like accuracy, precision, recall, F1 score, AUC, and log loss.

- **Overfitting:** The model showed a significant gap between training and testing accuracy, particularly when the training accuracy was much higher than the test accuracy.

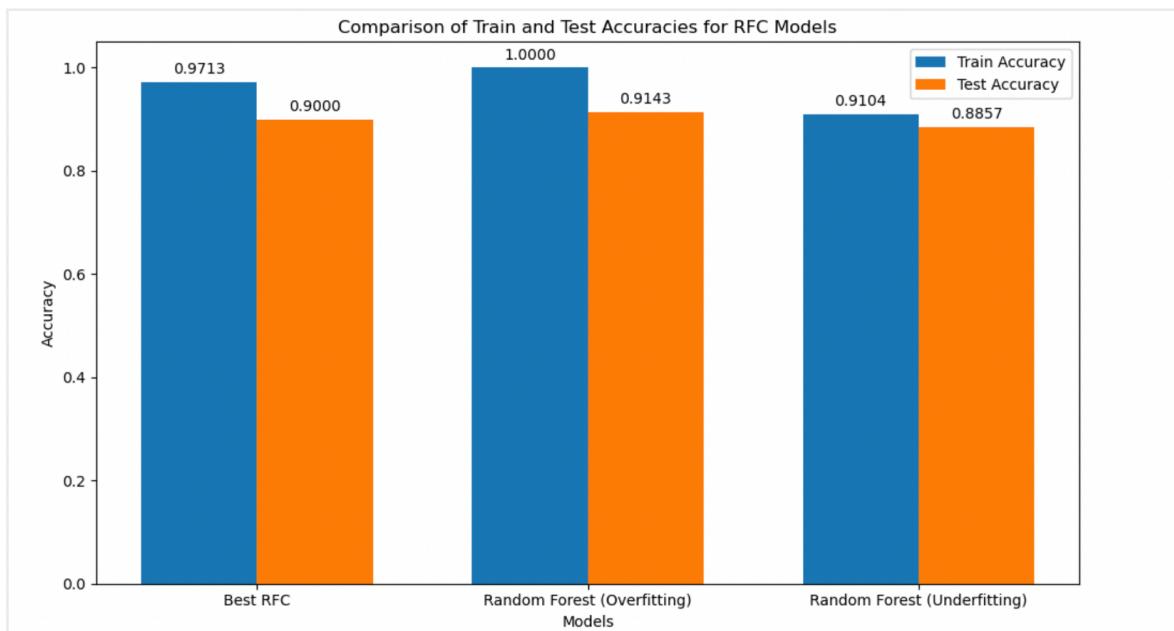
```
Random Forest (Overfitting):
Train Accuracy: 1.0000
Test Accuracy: 0.9143
[[135  0]
 [ 0 144]]
[[31  5]
 [ 1 33]]
```

- **Under-fitting:** In contrast, under-fitting was observed when the model's performance was poor both on the training and test data, indicating that the model was too simple to capture the underlying patterns of the data.

```

Random Forest (Underfitting):
Train Accuracy: 0.9104
Test Accuracy: 0.8857
[[115 20]
 [ 5 139]]
[[29 7]
 [ 1 33]]

```



3) DECISION-TREE:

Implementation Steps

1. Initial Setup: A basic Decision Tree Classifier was trained with default hyper-parameters and a train-test split of 80/20.
 2. **Hyper-parameter Tuning:** RandomizedSearchCV was used to explore a range of hyper-parameters, including criterion (gini /entropy), max_depth (with limits to prevent overfitting), min_samples_split (to control model complexity), min_samples_leaf (to avoid overfitting by reducing leaf nodes), class_weight (for imbalanced data), and max_features (to limit the number of features considered for each split). This approach employed 3-fold cross-validation to identify the optimal set of hyper-parameters.
 3. Model Evaluation: The best Decision Tree model was trained and evaluated on the test set using key performance metrics such as accuracy, precision, recall, F1 score, AUC, and log loss.
- **Overfitting:** The model showed a significant gap between training and testing accuracy, particularly when the training accuracy was much higher than the test accuracy.

```

Overfitting Decision Tree Model:
Train Accuracy: 1.0000
Test Accuracy: 0.7714
Training confusion matrix:
[[135  0]
 [ 0 144]]
Testing confusion matrix:
[[28  8]
 [ 8 26]]

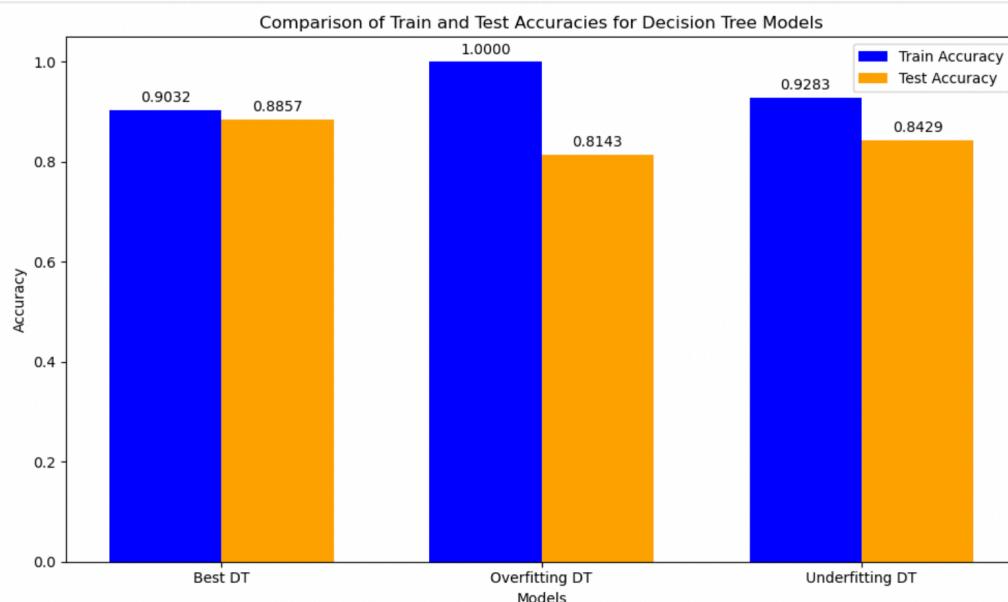
```

- **Under-fitting:** In contrast, under-fitting was observed when the model's performance was poor both on the training and test data, indicating that the model was too simple to capture the underlying patterns of the data.

```

Underfitting Decision Tree Model:
Train Accuracy: 0.8710
Test Accuracy: 0.8143
Training confusion matrix:
[[103  32]
 [ 4 140]]
Testing confusion matrix:
[[25 11]
 [ 2 32]]

```



RESULTS AND DISCUSSIONS

The Best Accuracies of the models obtained are as follows:

- K-Nearest Neighbours:
-> Basic KNN model

```

Basic KNN Model Evaluation:
[[27  5]
 [ 8 30]]
      precision    recall   f1-score   support
0         0.77     0.84     0.81      32
1         0.86     0.79     0.82      38
accuracy                          0.81      70
macro avg       0.81     0.82     0.81      70
weighted avg    0.82     0.81     0.81      70

```

-> Optimised KNN model (with tuned parameters and K=8)

The best k value is: 8

```

Evaluating KNN Model...
[[29  7]
 [ 3 31]]
Accuracy: 0.8571
Precision: 0.8623
Recall: 0.8571
F1 Score: 0.8569
AUC: 0.9105
Log Loss: 1.3167

```

- Random Forest:

```

Best RFC Model Evaluation:
Training confusion matrix:
[[127  8]
 [ 0 144]]
Testing confusion matrix:
[[30  6]
 [ 1 33]]
Train Accuracy: 0.9713
Test Accuracy: 0.9000
Precision: 0.9087
Recall: 0.9000
F1 Score: 0.8997
AUC: 0.9412
Log Loss: 0.3424

```

- Decision Tree:

```
Best Decision Tree Model Evaluation:  
Training confusion matrix:  
[[115  20]  
 [ 5 139]]  
Testing confusion matrix:  
[[28  8]  
 [ 2 32]]  
Train Accuracy: 0.9104  
Test Accuracy: 0.8571  
Precision: 0.8686  
Recall: 0.8571  
F1 Score: 0.8564  
AUC: 0.7970  
Log Loss: 3.8012
```

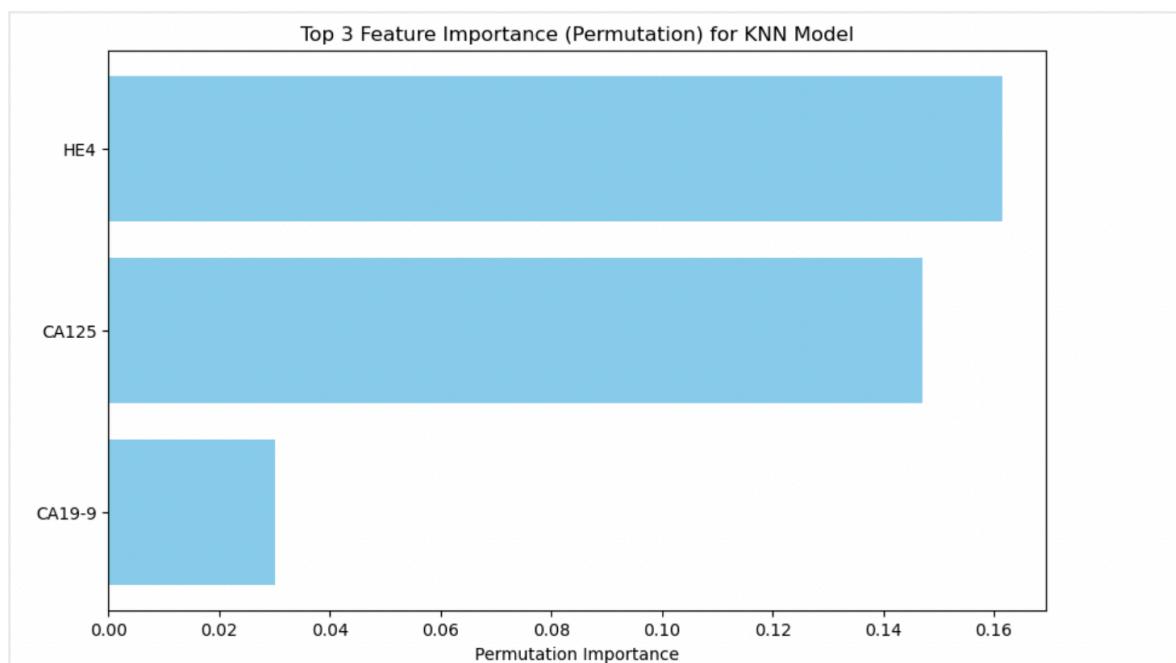
FEATURE IMPORTANCE ANALYSIS

Two key techniques were used to assess feature contributions:

1. Permutation Importance:

Identified the top features influencing model predictions, aiding in feature selection and ensuring interpretability.

- KNN Model :-

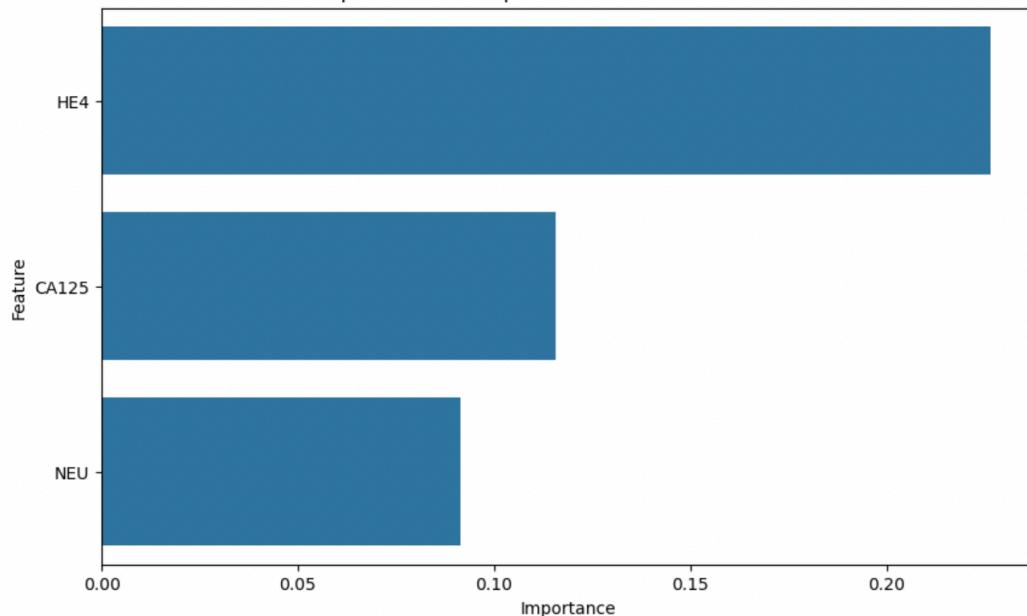


- RF Model :-

Top Three Important Features for RFC Model:

	Feature	Importance
6	HE4	0.226577
5	CA125	0.115553
22	NEU	0.091461

Top Three Most Important Features for RFC Model

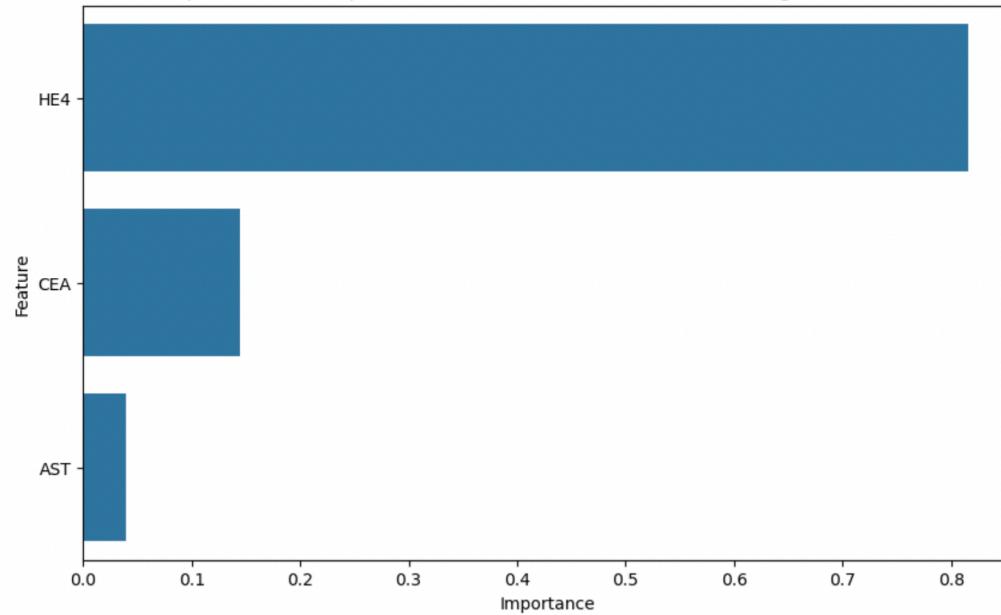


- DT Model :-

Top Three Important Features for DT Model:

	Feature	Importance
6	HE4	0.816087
7	CEA	0.144680
31	AST	0.039232

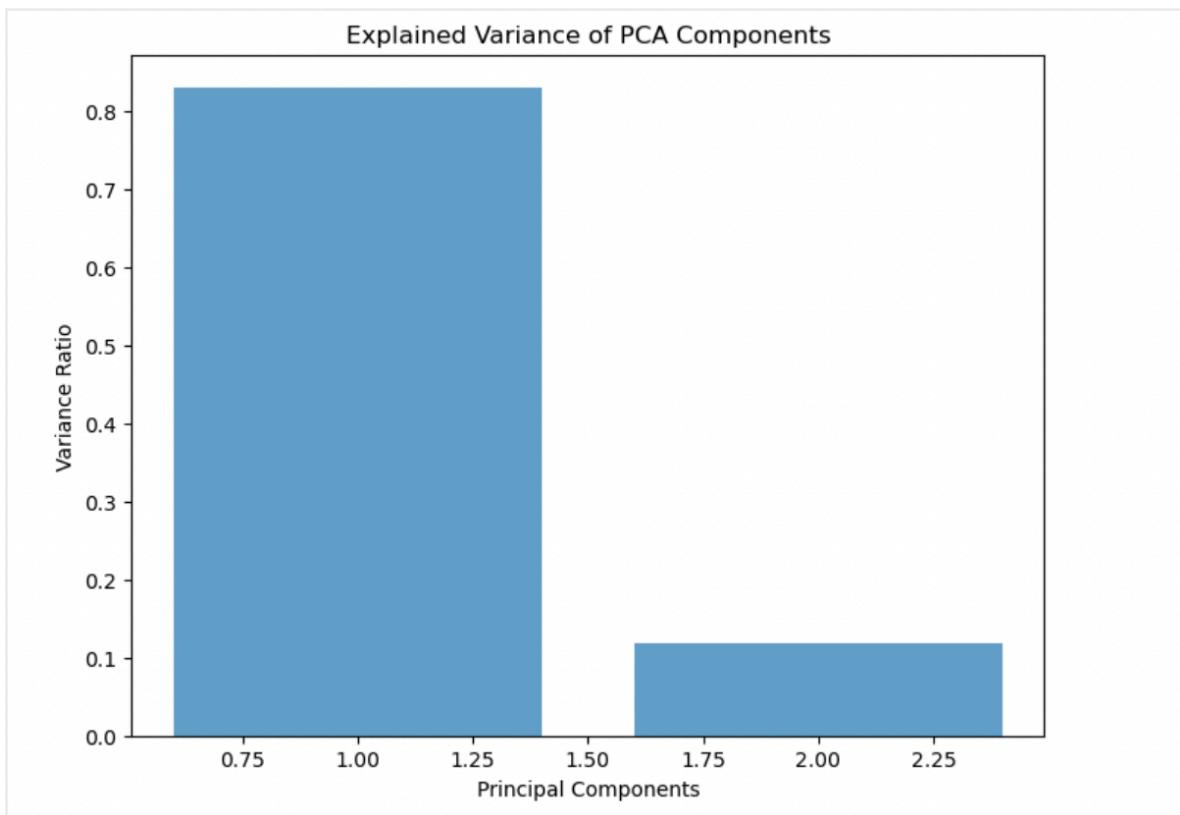
Top Three Most Important Features for Cancer Prediction using Decision Tree



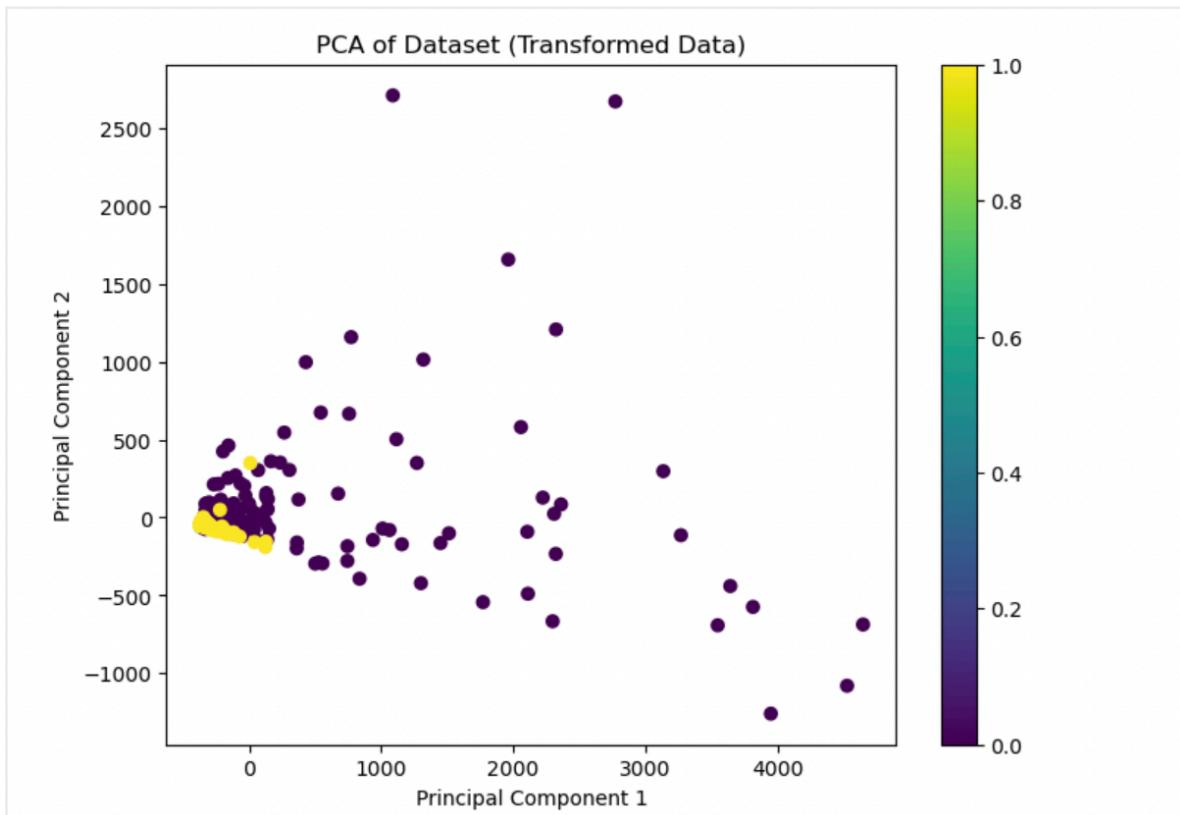
2. Principal Component Analysis (PCA):

Reduced dataset dimensionality, simplifying the data structure while retaining essential patterns for model training.

- KNN Model :-



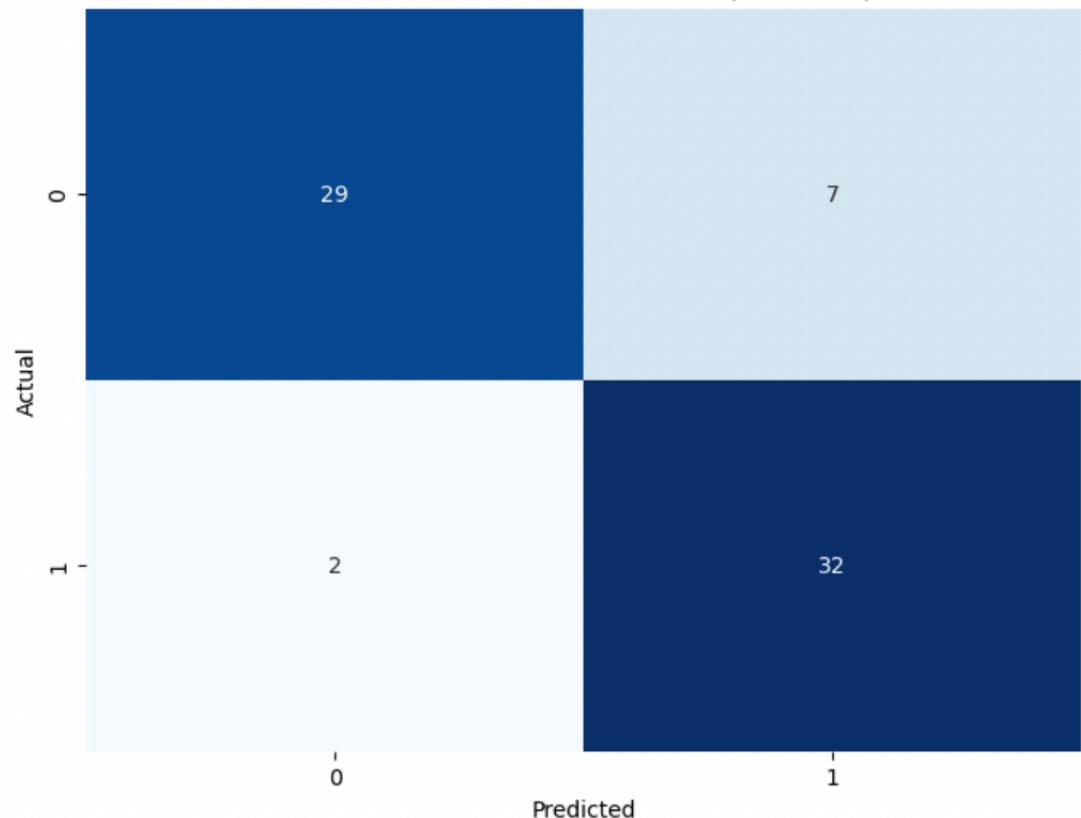
The explained variance ratio indicated that the first principal component (PC1) accounted for over 80% of the variance, highlighting its significant contribution to the dataset's structure.



The PCA scatterplot showed a clear clustering of data points, which suggests the dataset's separability in the reduced-dimensional space.

```
Train Accuracy: 0.8746
Test Accuracy: 0.8714
Training Confusion Matrix:
[[108  27]
 [ 8 136]]
Testing Confusion Matrix:
[[29  7]
 [ 2 32]]
```

Confusion Matrix for KNN Model (Test Data)



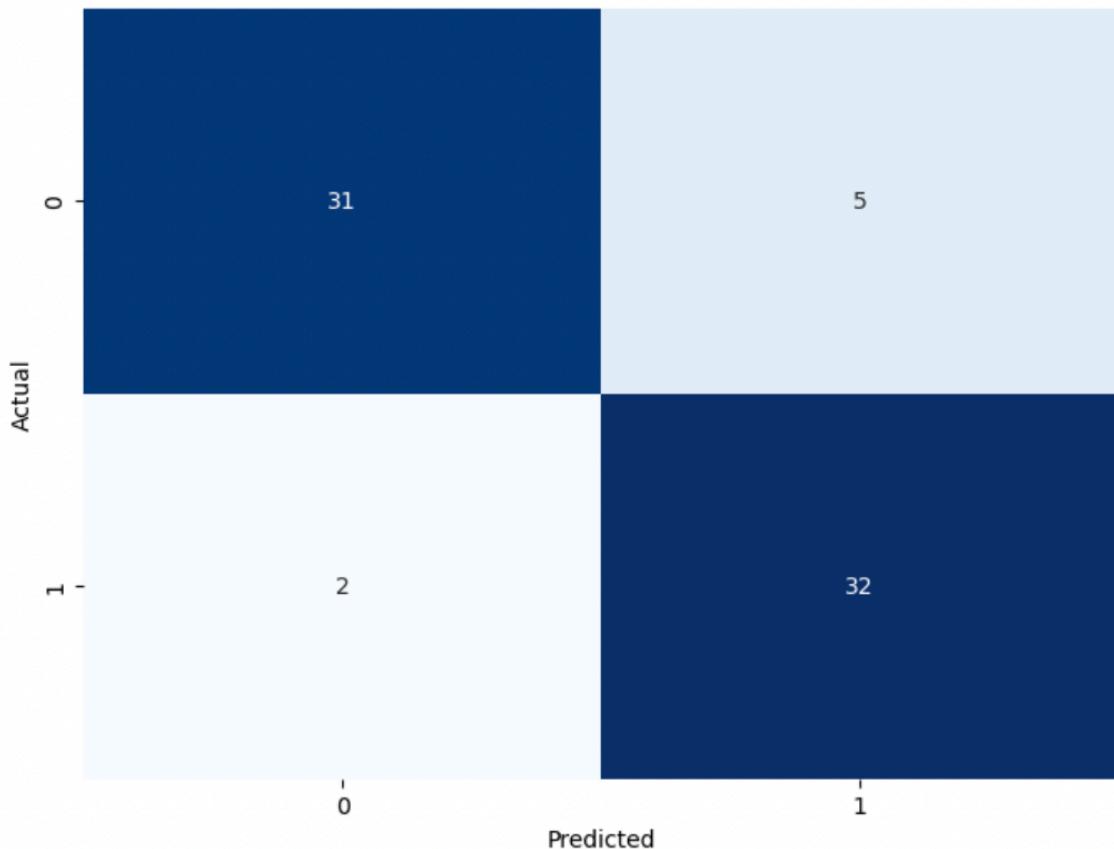
- RF Model :-

```

Train Accuracy: 1.0000
Test Accuracy: 0.9000
Training Confusion Matrix:
[[135  0]
 [ 0 144]]
Testing Confusion Matrix:
[[31  5]
 [ 2 32]]

```

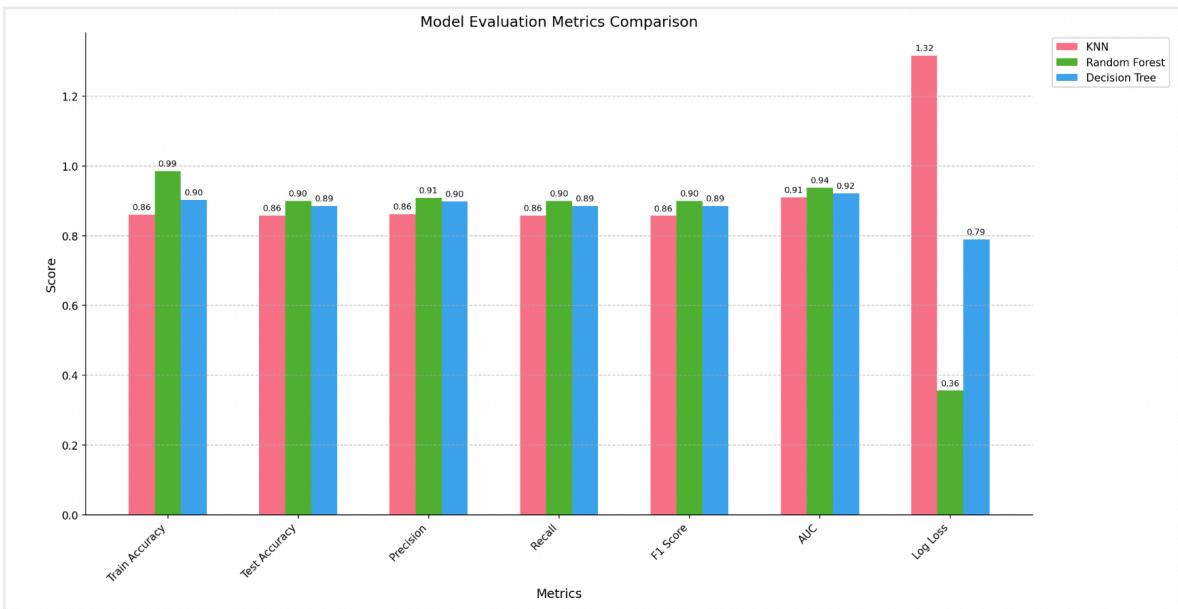
Confusion Matrix for Test Data



CONCLUSION :-

In conclusion, this study employs advanced data mining and machine learning techniques, including KNN, RF and DT, to uncover the relationship between biological/clinical data and cancer.

The **Random Forest model** outperforms the others with the highest accuracy (90%) and best metrics across precision, recall, F1 score, and AUC. The **Decision Tree** follows closely with accuracy of (88%), while **KNN** shows the lowest accuracy of (85.71%) and less effective precision and recall. Overall, **Random Forest** is the most effective model for EARLY STAGE DETECTION OF OVARIAN CANCER.



APPLICATIONS :-

- 1. Early Disease Detection:** Identifying diseases in their early stages for timely intervention and treatment.
- 2. Personalised Treatment Plans:** Tailoring medical care to individual patients based on their specific data.
- 3. Biomarker Discovery:** Identifying biological markers to aid in disease diagnosis and prognosis.
- 4. Clinical Decision Support Systems:** Enhancing healthcare decisions through data-driven insights.
- 5. Predictive Health Analytics:** Using data to predict potential health risks and prevent future medical issues.