CERVICAL CANCER DETECTION USING MACHINE LEARNING

ANALYSIS AND MODEL IMPLEMENTATION

IMPORTANCE OF EARLY DETECTION

• Early detection of cervical cancer greatly increases the chances of successful treatment and survival.

• Regular screening tests, such as Pap smears and HPV tests, can detect precancerous changes or early-stage cancer before symptoms develop.

• When cervical cancer is detected early, it is highly treatable with less aggressive interventions, such as surgery, radiation therapy, or chemotherapy.

• Early detection also reduces the need for extensive and invasive treatments, resulting in better quality of life for patients.

AGENDA

Dataset Overview

Data Preprocessing

Exploratory Data Analysis (EDA)

Model Building

Results

DATASET OVERVIEW

Dataset Overview

Source: The dataset was obtained from https://www.kaggle.com/datasets/ranzeet013/cervical-cancer-dataset/data
Number of Samples: 835
Features: The dataset contains [number of features] features including [list key features].

DATA PREPROCESSING

Handling Missing Values:

Importance: Missing values can affect model performance and interpretation of results.

```
# Replace '?' with NaN
cancer_df = cancer_df.replace('?', np.nan)

# Drop rows or columns with missing values
cancer_df.dropna(inplace=True)
```

DATA PREPROCESSING

Data Cleaning Steps

Replacing '?' with NaN:

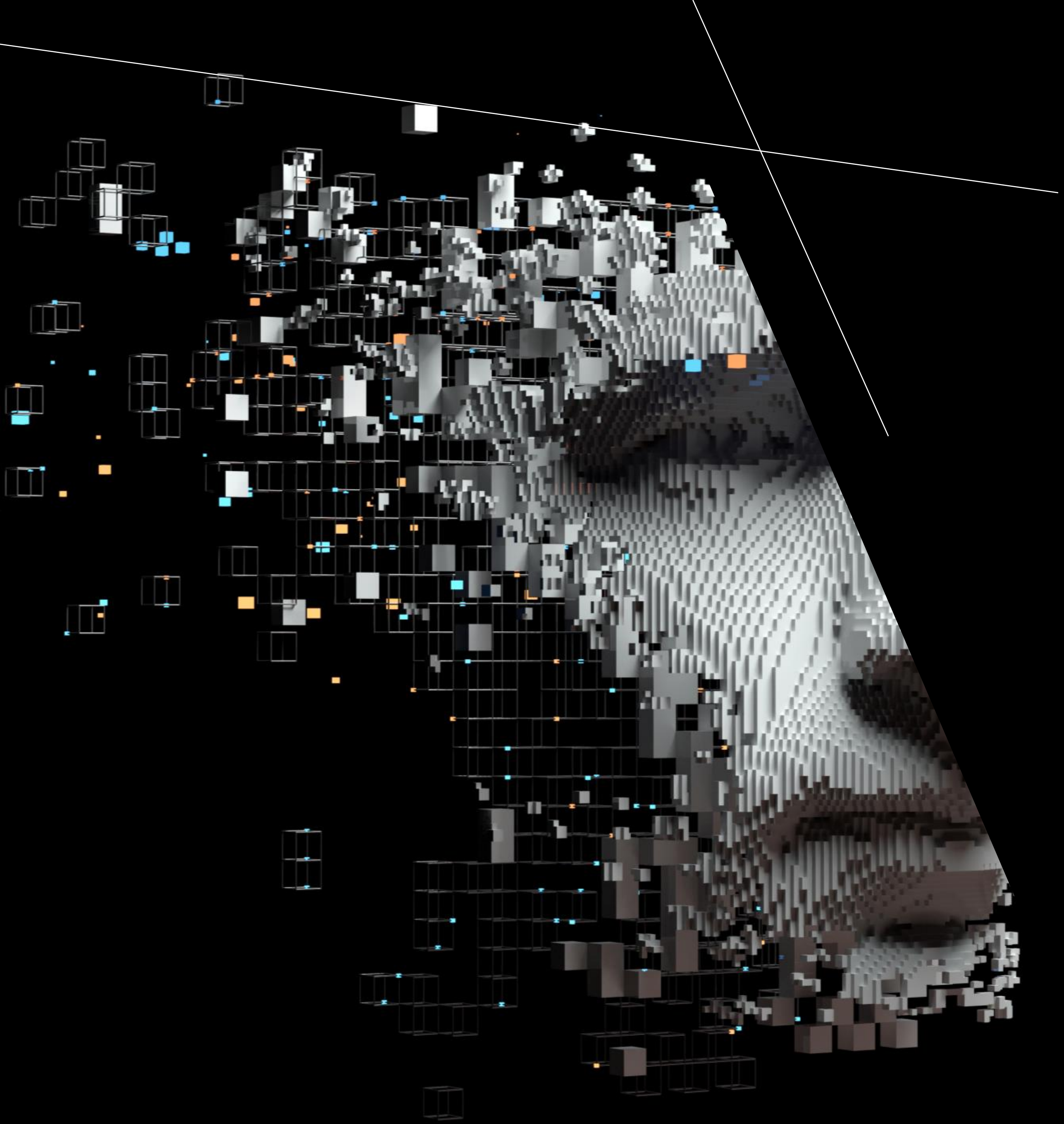
'?' was replaced with NaN to indicate missing values.

```
# Replace '?' with NaN
cancer_df = cancer_df.replace('?', np.nan)
```

Dropping Irrelevant Columns:

Columns such as 'STDs: Time since first diagnosis' were dropped as they were not relevant for the modeling task.

```
# Drop irrelevant columns
cancer_df = cancer_df.drop(columns=['STDs: Time
since first diagnosis'])
```



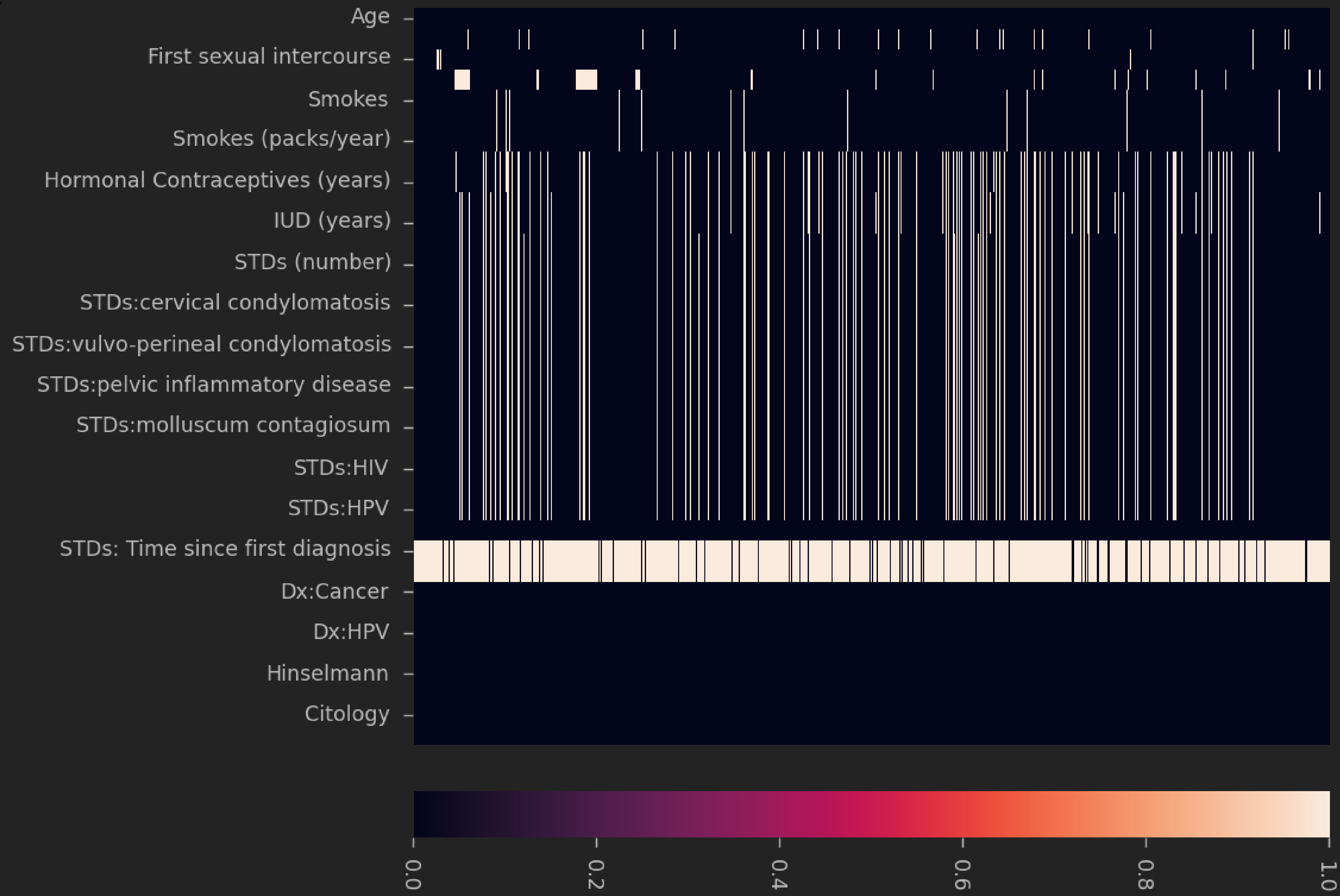
EXPLORATORY DATA ANALYSIS (EDA)

VISUALIZATIONS OF THE DATASET

Visualizations of the Dataset

Heatmap of Missing Values

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

plt.figure(figsize=(10, 8))
 sns.heatmap(cancer_df.isnull(), yticklabels=False, 
cmap='viridis', cbar=False)
plt.title('Heatmap of Missing Values’)
plt.show()
```



Age
First sexual intercourse
Smokes
Smokes (packs/year)
Hormonal Contraceptives (years)
IUD (years)
STDs (number)
STDs:cervical condylomatosis
STDs:vulvo-perineal condylomatosis
STDs:pelvic inflammatory disease
STDs:molluscum contagiosum
STDs:HIV
STDs:HPV
STDs: Time since first diagnosis
Dx:Cancer
Dx:HPV
Hinselmann
Citology
0.0
0.2
0.4
0.6
0.8
1.0

VISUALIZATIONS OF THE DATASET

Visualizations of the Dataset

Histograms of Features

```
cancer_df.hist(bins=20, figsize=(15, 10), color='skyblue')

plt.suptitle('Histograms of Features', fontsize=16)

plt.show()
```

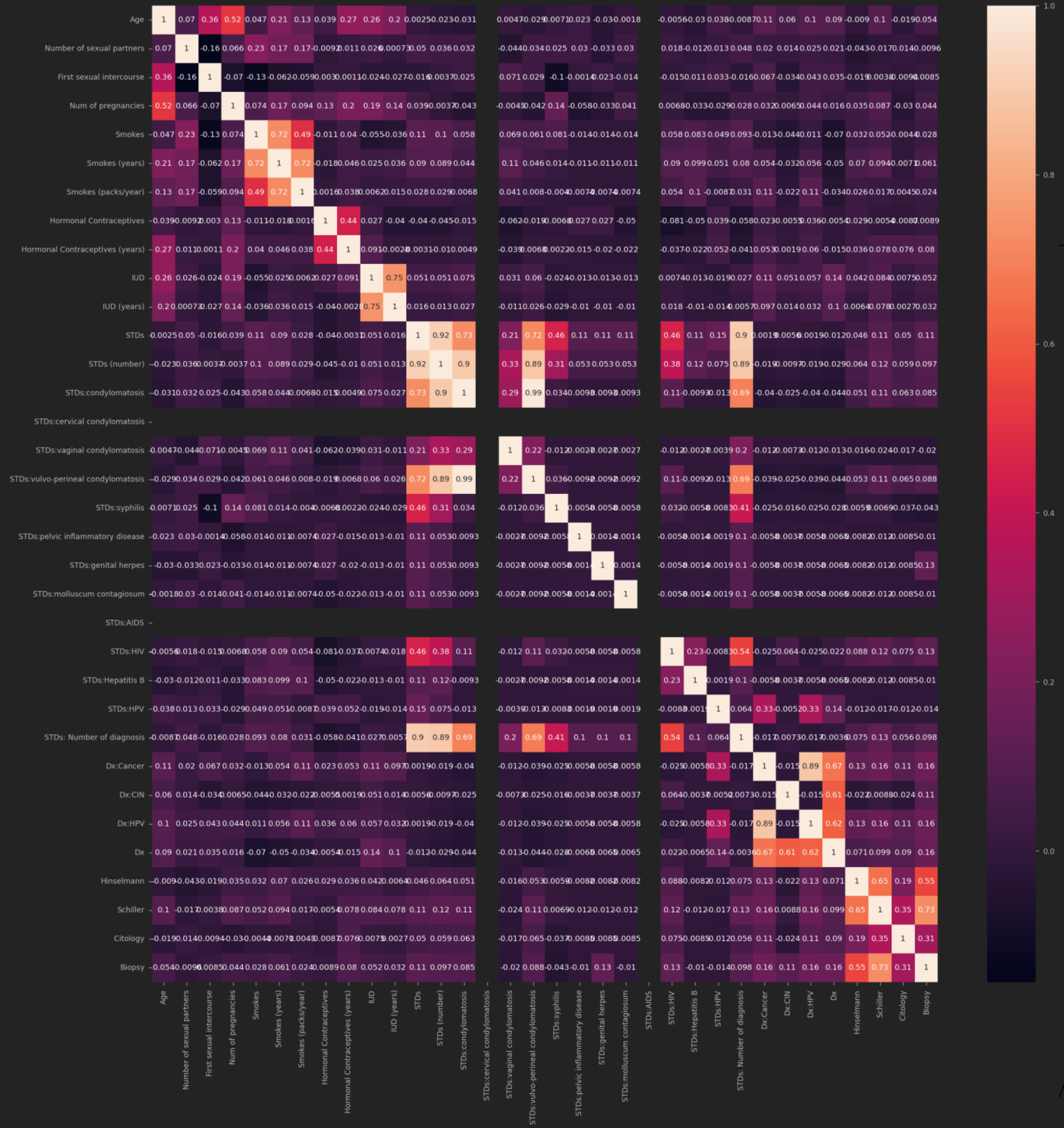

Age
Number of sexual partners
First sexual intercourse
Num of pregnancies
Smokes
Smokes (years)
Smokes (packs/year)
Hormonal Contraceptives
Hormonal Contraceptives (years)
IUD
IUD (years)
STDs
STDs (number)
STDs:condylomatosis
STDs:cervical condylomatosis
STDs:vaginal condylomatosis
STDs:vulvo-perineal condylomatosis
STDs:syphilis
STDs:pelvic inflammatory disease
STDs:genital herpes
STDs:molluscum contagiosum
STDs:AIDS
STDs:HIV
STDs:Hepatitis B
STDs:HPV
STDs: Number of diagnosis
Dx:Cancer
Dx:CIN
Dx:HPV
Dx
Hinselmann
Schiller
Citology
Biopsy

VISUALIZATIONS OF THE DATASET

Correlation Analysis

Correlation Matrix

```
corr_matrix = cancer_df.corr()


plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, 
cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```


Age
Number of sexual partners
First sexual intercourse
Num of pregnancies
Smokes
Smokes (years)
Smokes (packs/year)
Hormonal Contraceptives
Hormonal Contraceptives (years)
IUD
IUD (years)
STDs
STDs (number)
STDs:condylomatosis
STDs:cervical condylomatosis
STDs:vaginal condylomatosis
STDs:vulvo-perineal condylomatosis
STDs:syphilis
STDs:pelvic inflammatory disease
STDs:genital herpes
STDs:molluscum contagiosum
STDs:AIDS
STDs:HIV
STDs:Hepatitis B
STDs:HPV
STDs: Number of diagnosis
Dx:Cancer
Dx:CIN
Dx:HPV
Dx
Hinselmann
Schiller
Citology
Biopsy
1.0
0.8
0.6
0.4
0.2
0.0

MODEL BUILDING

DESCRIPTION OF XGBOOST ALGORITHM

XGBoost (Extreme Gradient Boosting)

- XGBoost is a powerful and efficient implementation of gradient boosting algorithms designed for speed and performance.
- It builds multiple weak learners (typically decision trees) sequentially, with each subsequent model correcting the errors of its predecessor.
- XGBoost uses gradient descent optimization techniques to minimize a specific loss function, making it highly effective for classification tasks like cervical cancer detection.

SPLITTING THE DATASET

Training, Validation, and Testing Sets

```
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
X_test, X_val, y_test, y_val = train_test_split(X_test, y_test, test_size=0.5, random_state=42)
```

TRAINING THE XGBOOST MODEL

Training the Model

```
import xgboost as xgb


model = xgb.XGBClassifier(learning_rate=0.1, max_depth=50, n_estimators=100)
model.fit(X_train, y_train)
```

MODEL EVALUATION METRICS

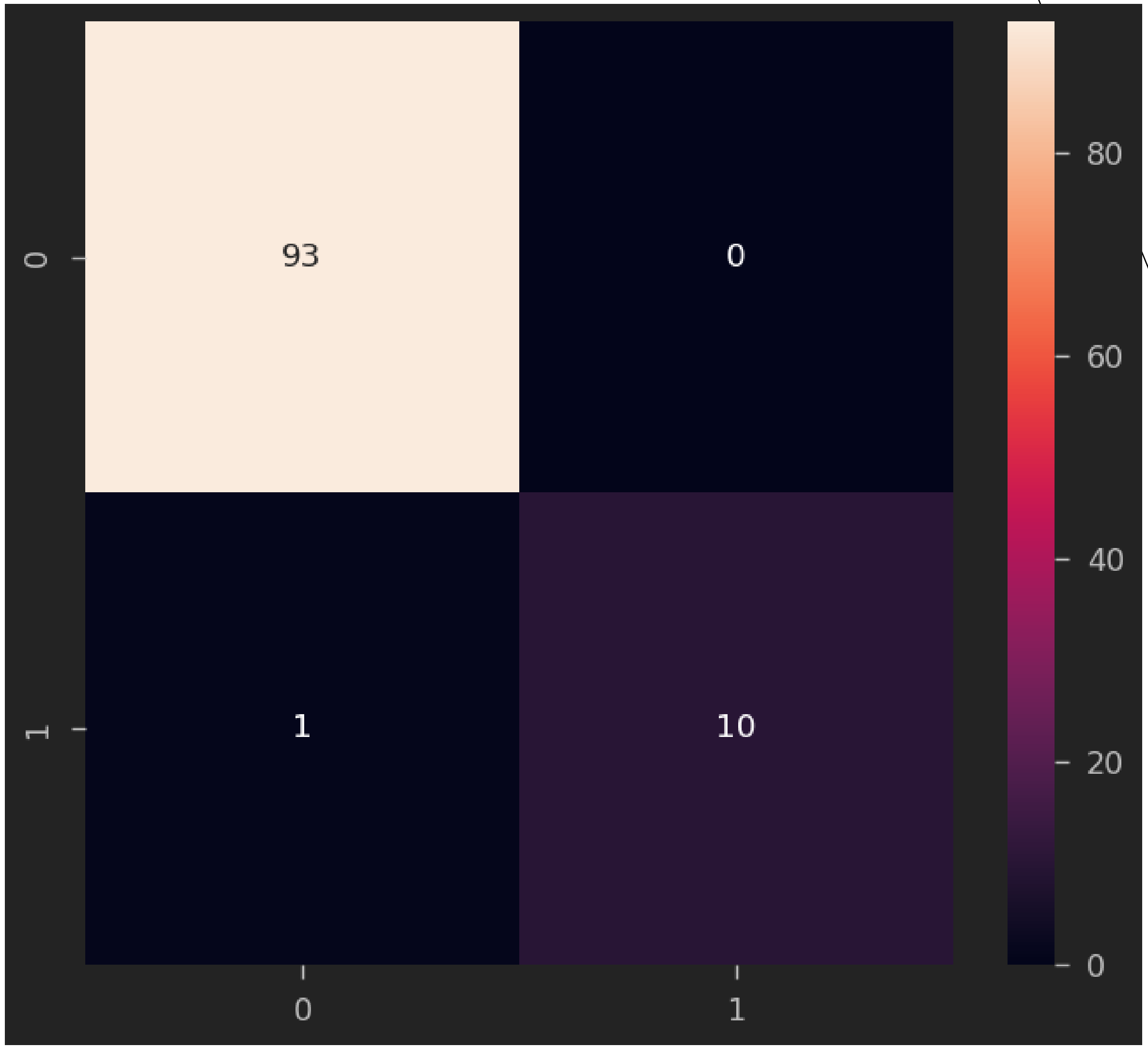
Accuracy, Precision, Recall, F1-Score

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score


y_pred = model.predict(X_val)
accuracy = accuracy_score(y_val, y_pred)
precision = precision_score(y_val, y_pred)
recall = recall_score(y_val, y_pred)
f1 = f1_score(y_val, y_pred)


print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-Score:", f1)
```

DISPLAY CONFUSION MATRIX

Confusion Matrix

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

cm = confusion_matrix(y_val, y_pred)
sns.heatmap(cm, annot=True, cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```



RESULTS

Training and Testing Accuracy Scores

- Training Accuracy Score
- Testing Accuracy Score

Classification Report

- Precision, Recall, F1-Score

TRAINING AND TESTING ACCURACY SCORES

Training Accuracy Score

```
train_accuracy = model.score(X_train, y_train)
print("Training Accuracy Score:", train_accuracy)
```

```
0.9952076677316294
```

TRAINING THE XGBOOST MODEL

Testing Accuracy Score

```
test_accuracy = model.score(X_test, y_test)
print("Testing Accuracy Score:", test_accuracy)
```

```
0.9903846153846154
```

CLASSIFICATION REPORT

Precision, Recall, F1-Score

```
from sklearn.metrics import classification_report


y_pred_test = model.predict(X_test)
classification_rep = classification_report(y_test, y_pred_test)
print("Classification Report:\n", classification_rep)
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.99      0.99        94
         1.0       0.91      1.00      0.95        10

    accuracy                           0.99       104
   macro avg       0.95      0.99      0.97       104
weighted avg       0.99      0.99      0.99       104
```

CONCLUSION

Summary of Findings

Potential Real-World Applications

Limitations and Future Work

SUMMARY OF FINDINGS

Model Performance: The XGBoost model achieved 0.9903846153846154 accuracy on the testing dataset, indicating remarkable accuracy.

Key Insights: The Correlation analysis uncovered good findings with 93 True positives and 10 true negatives along with the accuracy indicating good training.

POTENTIAL REAL-WORLD APPLICATIONS

- Early Detection: The developed model can potentially assist healthcare professionals in early detection and diagnosis of cervical cancer.
- Resource Allocation: By accurately identifying high-risk patients, healthcare resources can be allocated more efficiently, improving patient outcomes and reducing healthcare costs.
- Decision Support System: The model can serve as a decision support system for healthcare providers, aiding in treatment planning and patient management.

LIMITATIONS AND FUTURE WORK

Data Quality: Limited availability or quality of data may have impacted model performance. Future work could involve collecting more comprehensive and diverse datasets to improve model robustness.

Model Generalization: Further evaluation on diverse datasets and external validation in clinical settings are necessary to assess the generalizability and reliability of the model.

Feature Engineering: Exploration of additional features and advanced feature engineering techniques could enhance the model's predictive capabilities and interpretability.

Interpretability: Developing methods for interpreting model predictions and providing explanations to healthcare professionals is crucial for gaining trust and acceptance in clinical practice.

THANK YOU

Abhijith Nair

Haneesh Kenny

Aditya Vats

Mehul Karwa

Khush Chauhan