A black and white logo

Description automatically generated

**TRIBHUVAN UNIVERSITY**

Faculty of Management

**National College of Computer Studies**

Paknajol, Kathmandu

**Project Report on**

**Risk Factors Cervical Cancer**

**Submitted By:** **Submitted To:**

Name: Rojen Dangol Mausam Rajbanshi

BIM 5th Semester

Sec: B

Roll no.:22

Date: ……………………

**Acknowledgement**

We would like to express our gratitude to all those who have contributed to the completion of this report. We would like to thank Mr. Mausam Rajbanshi for their guidance and support throughout the writing process. Their insights and expertise were invaluable in helping us to understand the topic and produce a high-quality report.

We are also grateful to National College of Computer Studies who gave us the opportunity to do this wonderful project. This report would not have been possible without the support and collaboration of so many people, and we are deeply grateful for their contributions.

**Table of Contents**

[1. Introduction 1](#_Toc144812906)

[2. Background 2](#_Toc144812907)

[3. Objectives 3](#_Toc144812908)

[4. Implementation 4](#_Toc144812909)

[5. Conclusion 6](#_Toc144812910)

# Introduction

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to learn and make predictions or decisions without being explicitly programmed. It is a powerful technology that has revolutionized various industries, from healthcare and finance to entertainment and autonomous vehicles. In this introduction to machine learning, we'll cover some fundamental concepts to help you understand its core principles. [1]

A screenshot of a computer

Description automatically generated

Key components of Machine learning:

**Data**: High-quality and representative data is crucial for training machine learning models. It serves as the foundation upon which the model learns patterns.

**Features**: Features are the specific attributes or characteristics used to represent data. Effective feature selection or engineering is essential for model performance.

**Algorithm**: The choice of machine learning algorithm depends on the problem type and dataset. Common algorithms include decision trees, neural networks, support vector machines, and k-means clustering.

**Training**: During training, the model adjusts its parameters to minimize the difference between its predictions and the actual outcomes in the training data.

**Testing**/**Evaluation**: After training, the model is tested on a separate dataset (test data) to assess its generalization performance.

**Model Deployment**: Once a model performs well, it can be deployed in real-world applications to make predictions or automate decision-making.

# Background

Data classification is a fundamental task in machine learning that involves training a model to classify data into predefined classes or categories. Cervical Cancer Risk Factors for Biopsy: This Dataset is Obtained from UCI Repository and kindly acknowledged. This file contains a List of Risk Factors for Cervical Cancer leading to a Biopsy Examination. [2]

The primary goal of this data classification task is to construct a machine learning model capable of accurately classifying Cervical Cancer Risk into leading to a Biopsy. To accomplish this, we use Python, a popular programming language for data analysis and machine learning, and a variety of libraries and techniques to handle the dataset, build a classification model, and evaluate its performance.

The following are the general steps involved in this data classification task:

* **Data Collection**: To begin, we obtain the Cervical Cancer Risk Factors dataset, which is easily accessible in Python via the scikit-learn library. This dataset will be the basis for training and testing our classification model.
* **Data Preparation**: To facilitate data manipulation, the dataset is organized into a structured format, typically a DataFrame. Furthermore, the data is divided into two subsets: one for model training and one for model evaluation. To ensure consistent scaling of features, data preprocessing steps such as standardization may be used.
* **Algorithm Selection**: We select an appropriate machine learning algorithm for this classification task. In this example, we use the k-nearest neighbors (KNN) algorithm, Logistic Regression Random Forest Classifier which are straightforward yet effective method for classification tasks. The algorithm chosen is determined by the nature of the data and the requirements of the problem.
* **Model Training**: Using the training dataset, the selected algorithm is trained. The model learns to recognize patterns and relationships in the data during this phase, allowing it to make predictions.
* **Model Evaluation**: The model is evaluated using the testing dataset after it has been trained. To evaluate the model's performance, metrics such as accuracy, precision, recall, and the F1-score are calculated. A confusion matrix is also generated to visualize the model's classification results.
* **Interpretation and Reporting**: Model evaluation results are interpreted and reported on to determine the model's accuracy and effectiveness in classifying Cervical Cancer Risk Factors. These findings are then documented, most commonly in the form of a classification report and a confusion matrix.

# Objectives

* To analyze and classify cervical cancer using machine learning algorithms that will help doctors accurately diagnose the cancer.
* To identify the correlations between the parameters that are likely to be responsible for cervical cancer.
* To conduct a survey that identifies women’s concerns about cervical cancer, and provides a message to the readers as well as the research community.

# Implementation

To carry out the classification task, we use Python, a versatile programming language for data analysis and machine learning. The following steps outline the key implementation elements:

1. Preparation of the Dataset:

The Cervical Cancer Risk Factor dataset is loaded to begin the implementation. This dataset is easily accessible in Python via the scikit-learn library, which is a powerful machine learning tool.

2. Data Division:

We divide the dataset into two subsets to facilitate model training and evaluation: a training set and a testing set. In this implementation, 80% of the data is allocated to the training set, with the remaining 20% used to test the model's performance. This division ensures that the model learns from a subset of the data and is tested on an independent dataset to determine its ability to generalize.

3. Standardization of Features:

While it is not always required, standardizing the features is a common practice to ensure that all attributes have the same scale. We use scikit-learn's StandardScaler to standardize the feature values, which can be especially useful when working with certain classification algorithms.

4. Choosing an Algorithm:

We use the Logistic Regression, k-nearest neighbors (KNN) , decision tree classifier and Random Forest Classifier algorithm for this classification task.

5. Model Education:

The training set is used to train the chosen classifiers. During this stage, the model learns to recognize patterns and relationships in data, allowing it to make predictions about Cervical Cancer.

6. Model Assessment:

Following model training, we assess its performance with a variety of classification metrics. Accuracy, precision, recall are all important metrics. These metrics reveal how well the model distinguishes Cervical Cancer.

7. Visualization of Results:

To improve the way the data are presented, we use visual aids like a confusion matrix. An easy-to-understand depiction of the model's classification results is provided by a confusion matrix, which shows the proportion of accurate and erroneous classifications.

All of these phases are included in the implementation code, which can be found below. It not only demonstrates how the classification task was technically completed, but it also provides the framework for later analysis and interpretation of the results.

**The codes used for the projects are given below:**

* **Imports**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import confusion\_matrix,classification\_report, accuracy\_score

from sklearn.metrics import precision\_score,recall\_score,f1\_score

from sklearn.metrics import plot\_roc\_curve [3]

* **Read the CSV file**

df=pd.read\_csv("./CSV/risk\_factors\_cervical\_cancer.csv")

df.head()

**A screenshot of a computer screen

Description automatically generated**

* Describing the Data

df.info() df.isna().sum()

A screenshot of a computer code

Description automatically generated A screenshot of a computer screen

Description automatically generated

df.describe()

A screenshot of a computer

Description automatically generated

* **Dropping the column with no useable values**

df[df=='?'].count()

A screenshot of a computer screen

Description automatically generated

df.drop(["STDs: Time since first diagnosis","STDs: Time since last diagnosis"],axis=1,inplace=True)

* **Filling the null values**

numerical = ['Age', 'Number of sexual partners', 'First sexual intercourse','Num of pregnancies', 'Smokes (years)', (packs/year)','Hormonal Contraceptives (years)','IUD (years)','STDs (number)']

df=df.replace('?',np.NaN)

for num in numerical:

    print(num,'',df[num].astype(str).astype(float).mean())

    feature\_mean = round(df[num].astype(str).astype(float).mean(),1)

    df[num] = df[ num].fillna(feature\_mean)A number on a white background

Description automatically generated

for num in numerical:

    df[numerical] = df[numerical].astype(float)

categorical = ['Smokes', 'Hormonal Contraceptives', 'IUD','STDs', 'STDs:condylomatosis','STDs:cervical condylomatosis','STDs:vaginal condylomatosis','STDs:vulvo-perineal condylomatosis','STDs:syphilis',:pelvic inflammatory disease', 'STDs:genital herpes', 'STDs:molluscum contagiosum', 'STDs:AIDS', 'STDs:HIV', 'STDs:Hepatitis B', 'STDs:HPV']

df[categorical]=df[categorical].fillna(df.mode().iloc[0])

df.isna().sum()

**A screenshot of a computer

Description automatically generated**

* **Correlation Matrix**

df.corr()**A screenshot of a computer

Description automatically generated**

corr\_matrix=df.corr()

fig, ax=plt.subplots(figsize=(20,15))

ax=sns.heatmap(corr\_matrix,annot=True, linewidths=0.5,fmt='.2f', cmap='Blues')

**A screenshot of a computer

Description automatically generated**

* **Splitting the data into train and test**

X=df.drop(['Biopsy'],axis=1)

y=df['Biopsy']

X\_train, X\_test, y\_train, y\_test= train\_test\_split(X, y,test\_size=0.2,random\_state=42)

# Put models in a dictionary

models={"Logistic Regression": LogisticRegression(),

        "KNN":KNeighborsClassifier(),

        "Random Forest": RandomForestClassifier()}

# Create a function to fit and score models

def fit\_and\_score(models, X\_train, X\_test, y\_train, y\_test):

    np.random.seed(42)

    model\_scores ={}

    for name, model in models.items():

        # Fit the model to the data

        model.fit(X\_train,y\_train)

        # Evaluate the model and append its score to model scores

        model\_scores[name]=model.score(X\_test,y\_test)

    return model\_scores

model\_scores = fit\_and\_score(models=models,

                             X\_train=X\_train,

                             X\_test=X\_test,

                             y\_train=y\_train,

                             y\_test=y\_test)

model\_compare=pd.DataFrame(model\_scores, index=["accuracy"])

model\_compare.T.plot.bar(figsize=(10,10));

**A blue and white striped square

Description automatically generated**

* **Visualizing Cross validated metrics**

cv\_metrics=pd.DataFrame({"Accuracy":cv\_acc,

                         "Precision": cv\_precision,

                         "Recall": cv\_recall,

                         "F1": cv\_f1},

                         index=[0])

cv\_metrics.T.plot.bar(title="Cross-validated classfication metrics",

                 legend=False);

**A bar graph with different numbers

Description automatically generated with medium confidence**

* **Visualize feature importance**

feature\_df=pd.DataFrame(feature\_dict,index=[0])

feature\_df.T.plot.bar(title="Feature Importance", legend=False, figsize=(15,10)); **A graph with blue lines and white text

Description automatically generated**

# Conclusion

In this study, the authors classified research based on the prediction of cervical cancer risks. The authors profiled data and performed comprehensive benchmarking to evaluate the performance of risks using predictive models based on precision, recall, F1-score, and support. The proposed deep learning model was implemented using the Python programming language with packages and libraries. The cervical cancer dataset was used to perform basic data analysis, then data standardization and visualization were performed. Finally, the model was trained for the accurate prediction of cervical cancer, and the accuracy and performance of the model were also evaluated. The dataset was chosen specifically to evaluate attributes such as smoking, STDs, STD, AIDS, first sexual intercourse, and cytology, which are the major risk factors of cervical cancer. Based on the computational results obtained, one hundred and nineteen instances were under the ‘class zero’ predicted class, while only three instances were found under ‘class one’ of the predicted class, which illustrated the proposed model performs very well for the cervical cancer dataset.

**References**

|  |  |
| --- | --- |
| [1] | "ChatGPT," [Online]. Available: https://chat.openai.com. |
| [2] | "Kaggle," [Online]. Available: https://www.kaggle.com/code/sashashirkhodaei/cervical-cancer-risk-prediction. |
| [3] | "scikit-learn," [Online]. Available: https://scikit-learn.org/stable/. |