Journal of Finance, Economics and Data Analytics
Vol. 1 No. 1 (Jan-June 2026); pp 32-45

Machine Learning for Predictive Modelling: Why Data Quality Matters More Than Model Choice

Dr. Ritu Vashistha

Assistant professor

School of Digital Health, IIHMR University, Jaipur

Dr. Harshita Bhargava

Dr. Rajneesh Chaturvedi

Department of Computer Science & IT, IIS (deemed to be) University, Jaipur

Abstract: The study is done on the early prediction of Dengue fever by using demographic data and haematological features. Various Machine learning models are used in the study to have a comparative analysis between performance of different predictive models. The study also highlights that only using good Machine Learning models are not sufficient, the data also needs to support. The data collected should be containing required fields and adequate information in it.

Keywords: Machine learning, Predictive Modelling, XG Boost, Random Forest, Support Vector Machine (SVM)

Introduction

As per WHO "Dengue is a viral infection caused by the dengue virus (DENV), which is transmitted to humans through the bite of infected mosquitoes." Masoud et al., (2024), Schaefer et al., (2024) and Sanjay et al, (2019), in their research said that Dengue is one of the most dangerous viruses which can lead to death. There is no specific treatment for that and only prevention, awareness, early diagnosis and right treatment on right time can lead to prevention from fatality. Another team of Authors do agree with this that Dengue can cause unrepairable damage and only prevention is the cure. But still its diagnosis can be done with the help of various physical examinations which covers Platelet counts, White blood cell count, Haemoglobin and platelet distribution width these tests can warn patients about the severity of dengue and treatment can be done accordingly. These physical examinations play a very crucial role in understanding the condition of a patient and how severe the condition is and how much worse it could be in near future. Nyenke et al., (2023), Zhang et al., (2014).

Machine Learning predictive models can also be helpful in early detection of Dengue cases as per physical examination reports, it can predict whether a person is having Dengue or not based on the similar Datasets on which model can be trained. William et al., (2021) in his study has discussed about Logistic Regression, linear regression and general linear model and how it can be helpful in prediction of Dengue cases, the highest accuracy ratio was covered by General linear model with 70% accuracy score. Mayrose et al., (2023) developed a model based on SVM with accuracy score of 95 percent, they have used blood smear images to see the effectiveness of haematological parameters for Dengue detection. They have used the clinical test data like white blood cell count and haemoglobin to support their study.

Recently the ML models have been used largely because it covers complex and non-linear relationships also and able to provide insights accordingly. Chen and Moraga (2025) use Machine algorithmic model to predict the dengue cases and shown how climate variables plays and important role in the study and produced better results than traditional statistical methods and using only one kind of parameters.

Ferreira et al. (2019) in his study uses the clinical diagnosis parameters to develop classification models such as platelet count and white blood cell count for Dengue diagnosis. Potts and Rothman (2008) also supported the above study results by extending the same in his study, who identified thrombocytopenia and leukopenia as main and important factors for predicting dengue infection.

The above studies highlighted how Machine learning models help in dengue prediction. And how clinical parameters can make the prediction more effective. This study will use the clinical test parameters for early prediction of Dengue cases and will compare the results of various predictive models to see the efficiency level of different models on used dataset.

Methodology And Model Development

This study uses quantitative method and data driven approach for developing various models to detect dengue using the clinical parameters along with demographic data as well. All models were developed using Python code and appropriate Machine learning libraries.

The data set has been taken from Kaggle, and it contains patients' demographic information like age and gender and clinical parameters such as haemoglobin, white blood cell count, differential count, RBC count, platelet count, platelet distribution width and then dengue label, which has 0 and 1. 0 means dengue not detected and 1 means detected.

Table 1: Dengue Detection Dataset

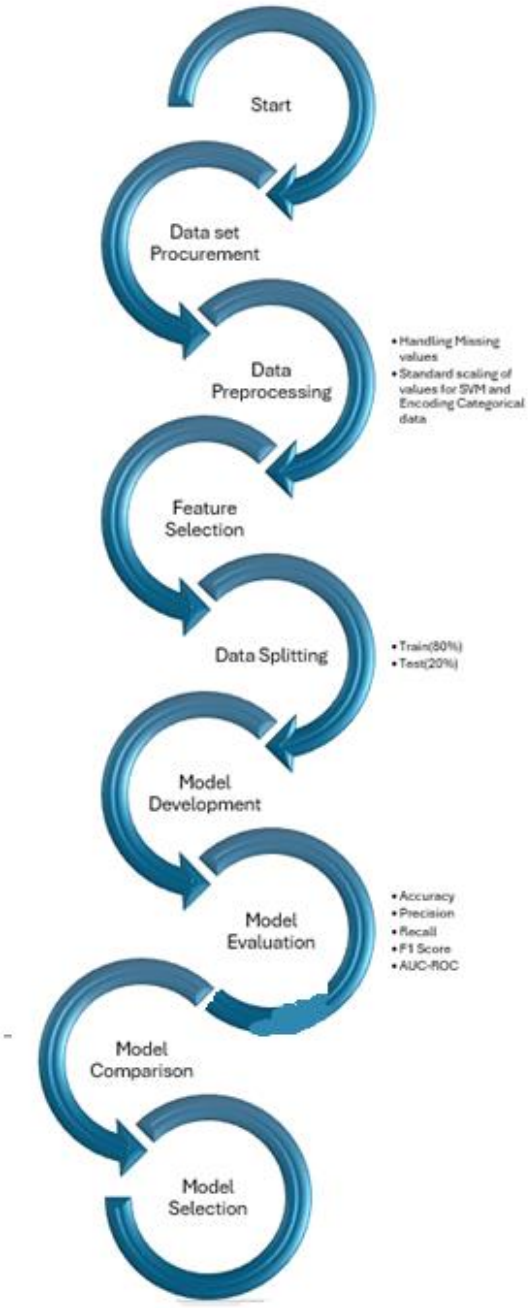
age	gender	hemoglobin_g_dl	wbc_count	differential_count	rbc_count	platelet_count	platelet_distribution_widt	dengue_label
43	Male	12.6	2200	1	1	62000	11	1
45	Male	13.2	3000	0	1	17000	17	1
50	Female	11	3300	1	1	19000	16.3	1
57	Female	11.9	3500	1	0	29000	14	1
51	Female	13	3100	0	1	30000	14.5	1
61	Male	15	3300	1	1	34000	20	1
6	Child	11	2300	1	0	69000	12.5	1
21	Male	14	2500	1	1	77000	13.3	1
29	Male	15	2400	1	1	78000	14.5	1

Source: https://www.kaggle.com/datasets/aravind3505/dengue-detection-dataset-clinical-data

The link of the dataset is provided below the table for experimentation purpose of readers. The first step before starting actual predictive modelling is to procure the dataset. For study purposes the author has taken from Kaggle, but the same model can be applied on any other primary data or secondary data collected by researchers.

The flow of study will be:

Figure 1: The flow of Predicting Dengue Cases by Machine Learning Models



Source: Self

Data set procurement and Data Description

To understand the dataset at first place we need to understand its structure and the types of values they are storing and how many Null values are there, so that it can be treated at first.

Command:

```
import pandas as pd

df = pd.read_csv("D:/research/dengue predictiction/Dengue_diseases_dataset_modified.csv")
print(df.shape)
print(df.info())
print(df.describe())
print(df['dengue_label'].value_counts())
```

Output:

```
(989, 9)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 989 entries, 0 to 988
Data columns (total 9 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   age                          989 non-null    int64
 1   gender                       989 non-null    object
 2   hemoglobin_g_dl              989 non-null    float64
 3   wbc_count                    965 non-null    float64
 4   differential_count           989 non-null    int64
 5   rbc_count                    989 non-null    int64
 6   platelet_count               973 non-null    float64
 7   platelet_distribution_width  970 non-null    float64
 8   dengue_label                 989 non-null    int64
dtypes: float64(4), int64(4), object(1)
memory usage: 69.7+ KB
None
                age  hemoglobin_g_dl     wbc_count  differential_count  \
count  989.000000       989.000000    965.000000          989.000000
mean    42.199191        13.712942   4338.031088            0.939333
std     20.941111         1.484111   2344.529755            0.238840
min      3.000000        11.000000   2000.000000            0.000000
25%     27.000000        12.600000   2600.000000            1.000000
50%     40.000000        13.700000   3200.000000            1.000000
75%     55.000000        15.000000   6200.000000            1.000000
max    120.000000        25.000000  10900.000000            1.000000

         rbc_count  platelet_count  platelet_distribution_width  dengue_label
count  989.000000      973.000000                   970.000000    989.000000
mean     0.938322   114702.241521                    22.848866      0.651163
std      0.240692    89421.766107                    14.692872      0.476843
min      0.000000    10000.000000                     1.000000      0.000000
25%      1.000000    46000.000000                    14.000000      0.000000
50%      1.000000    93000.000000                    17.800000      1.000000
75%      1.000000   162500.000000                    28.200000      1.000000
max      1.000000   500000.000000                   215.000000      1.000000
dengue_label
1    644
0    345
Name: count, dtype: int64
```

In the above code we have imported the Library “Pandas” to handle CSV file “Dengue_diseases_dataset_modified”. “Read.csv” command helps in reading csv file from address provided and store the values in object named “df” for further manipulation. “df.shape” provides the information about how many rows and columns are present in the dataset, the result shows, in this dataset we have 989 rows and and 9 columns. Then the next command “df.info” specifies details about the structure , the datatype of each column and how many non null values are present in each column.

Then the next command “df.describe” it takes all the numeric values and calculates all basic ststistical measures for them, it is also called EDA(Exploratory Data Analysis), it gives the idea about each column which helps in understanding distribution and detecting outliers if any.

The next command is very important for our dataset, “dengue_label.value counts , it tells how many are dengue patients and how many are not, it also gives the idea that, is our data is having equal kind of cases of dengue or non_dengue or not, it also helps in understanding is our data biased or fair.

Data preprocessing

The raw data which is either procured or collected may have issues like missing values, or categorical values which needs to be transformed or treated. In the code below gender values have been transformed from categorical to numerical for handling statistically. And before treating missing values it needs to be checked in which columns we have missing values and how many missing values each column have.

Command

```
# Convert categorical to numeric
df['gender'] = df['gender'].map({'Male': 1, 'Female': 0})

# Check missing values
print(df.isnull().sum())
```

Output

```
age                           0
gender                       34
hemoglobin_g_dl               0
wbc_count                    24
differential_count            0
rbc_count                     0
platelet_count               16
platelet_distribution_width  19
dengue_label                  0
dtype: int64
```

The output shows that in gender column 34 null values are there and similarly in wbc_count, platelet_count and platelet_distribution_width having 24, 16 and 19 missing values. It is very easy that we remove all those records which have Null values, but it is not a good practice, and it causes data loss as well. But all the fields/columns’ values can’t be treated equally.

In the code below, where missing values are handled it shows that for Gender “mode” function is used and for others “median” function is used instead of mean function. For gender “mode” is used so that most frequent repeated value can be used to fill missing value, as it is representing categorical value. For rest of columns “mean” is not used as it can be impacted by outliers present in the fields. After performing these operations all Null values have been replaced by appropriate values in respective columns.

Command

```
# Fill gender (categorical)
df['gender'].fillna(df['gender'].mode()[0], inplace=True)

# Fill numerical columns using median
df['wbc_count'].fillna(df['wbc_count'].median(), inplace=True)
df['platelet_count'].fillna(df['platelet_count'].median(), inplace=True)
df['platelet_distribution_width'].fillna(df['platelet_distribution_width'].median(), inplace=True)

# Verify again
print(df.isnull().sum())
```

Output

```
age                            0
gender                         0
hemoglobin_g_dl                0
wbc_count                      0
differential_count             0
rbc_count                      0
platelet_count                 0
platelet_distribution_width    0
dengue_label                   0
dtype: int64
```

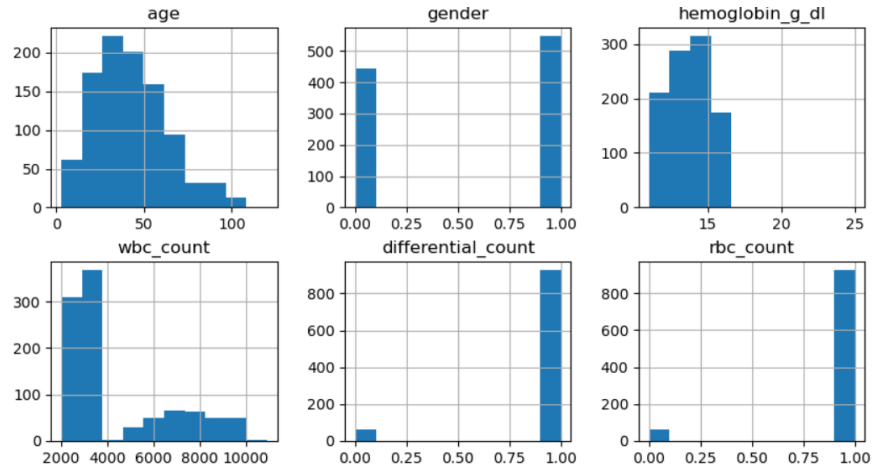
Code :

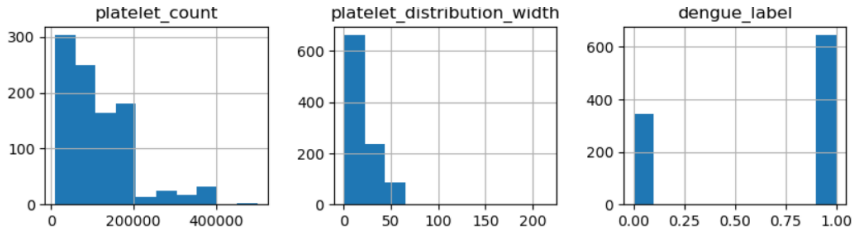
```
import matplotlib.pyplot as plt

df.hist(figsize=(10,8))
plt.show()

# Correlation
import seaborn as sns
sns.heatmap(df.corr(), annot=True)
plt.show()
```

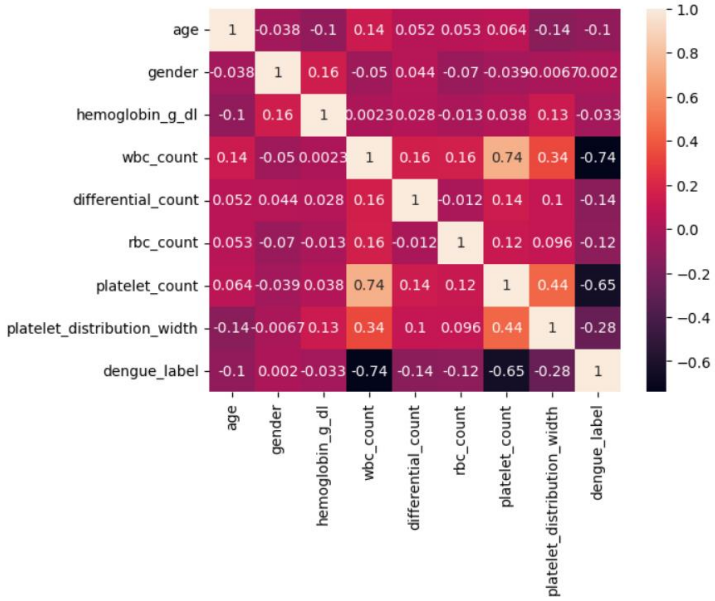
This part shows the EDA via Histogram and correlation Heatmap format. It is visible with these Histograms that the numerical variables, which are white blood cell count and platelet count, are right skewed which shows the presence of outliers and non-normality, which is obvious while dealing with clinical datasets.

The heatmaps indicates the Correlation analysis between different variables such as platelet count and white blood cell count have strong positive correlations and it has negative correlation with dengue diagnosis label, which showcases their importance as predictive features. In contrast to it age and gender show weak correlation, which has weak predictive power.

Output:







Data splitting

In our example we explicitly don't require feature selection, so as per flow diagram the next step is Data splitting between Training and Testing dataset.

Code:

```
from sklearn.model_selection import train_test_split

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

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

After importing required library for splitting data into training and testing, dependent and independent variable are created in form of X and Y. And the ratio of training and testing is kept 80:20. The 80 percent will be used for training the dataset and then 20 percent will be used to test the outputs from dataset. So as per above code "Dengue Label" is to be predicted by using rest of the columns from the dataset.

Model Development

To predict the Dengue infection multiple models have been used in this study, which includes Logistic Regression, Random Forest, XGBoost and Support Vector Machine (SVM).

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

model1 = LogisticRegression(max_iter=1000)
model1.fit(X_train, y_train)

y_pred1 = model1.predict(X_test)
print("Logistic Accuracy:", accuracy_score(y_test, y_pred1))
```

```
Logistic Accuracy: 0.5858585858585859
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier

model2 = RandomForestClassifier()
model2.fit(X_train, y_train)

y_pred2 = model2.predict(X_test)
print("RF Accuracy:", accuracy_score(y_test, y_pred2))
```

```
RF Accuracy: 0.5959595959595959
```

XGBoost

```
from xgboost import XGBClassifier

model3 = XGBClassifier()
model3.fit(X_train, y_train)

y_pred3 = model3.predict(X_test)
print("XGB Accuracy:", accuracy_score(y_test, y_pred3))
```

```
XGB Accuracy: 0.5909090909090909
```

Support Vector Machine (SVM)

For using SVM, the data needs to standardize, therefore StandardScaler have been used before applying the model.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
from sklearn.svm import SVC

model_svm = SVC(probability=True)  # probability=True needed for ROC

model_svm.fit(X_train_scaled, y_train)

y_pred_svm = model_svm.predict(X_test_scaled)
```

```
from sklearn.metrics import accuracy_score, classification_report

print("SVM Accuracy:", accuracy_score(y_test, y_pred_svm))
print(classification_report(y_test, y_pred_svm))
```

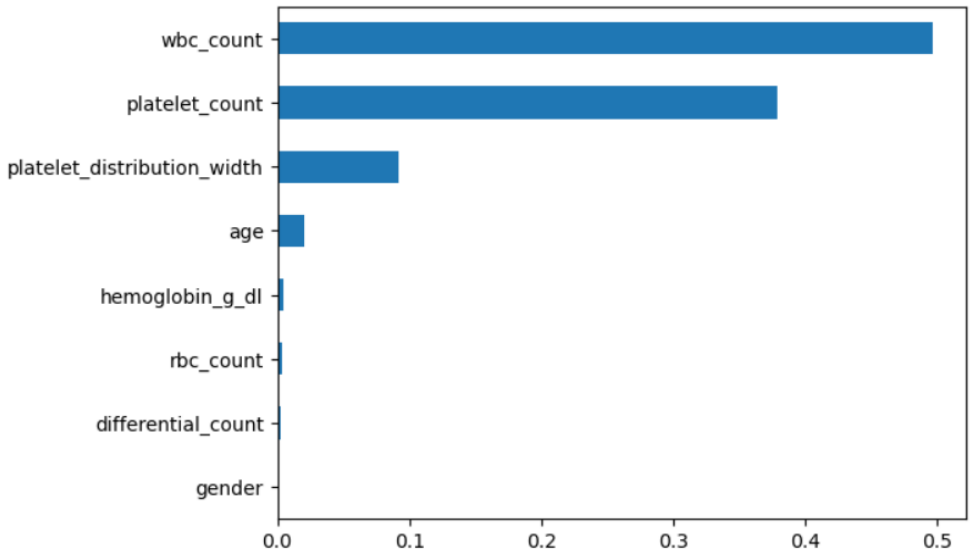
```
SVM Accuracy: 0.5909090909090909
```

With every model the accuracy score of that model is also being calculated and it have been noticed that all models fall from 58 to 60 percent approx. accuracy score, it means at max how accurately as per dataset training is done how and accurately it is predicting whether it is dengue case or not.

The highest Accuracy score belongs to Random Forest for this dataset, which is 59.59.

Feature Importance Analysis

```
import pandas as pd

feature_importance = pd.Series(model2.feature_importances_, index=X.columns)
feature_importance.sort_values().plot(kind='barh')
plt.show()
```



The feature importance was be seen by using tree-based decision structure, which is random forest in our study, by this we can analyse which feature play's important role in predicting the dengue. So, in our study we can see, wbc_count, platelet count and platelet distribution width plays important role in predicting Dengue cases and others have very low contribution or not at all.

Model Evaluation

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

def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    y_prob = model.predict_proba(X_test)[:,1]

    return {
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred),
        "Recall": recall_score(y_test, y_pred),
        "F1 Score": f1_score(y_test, y_pred),
        "AUC": roc_auc_score(y_test, y_prob)
    }

results = {
    "Logistic Regression": evaluate_model(model1, X_test, y_test),
    "Random Forest": evaluate_model(model2, X_test, y_test),
    "XGBoost": evaluate_model(model3, X_test, y_test),
    "SVM": evaluate_model(model_svm, X_test_scaled, y_test)
}

import pandas as pd
results_df = pd.DataFrame(results).T
print(results_df)
```

	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	0.585859	0.593985	0.738318	0.658333	0.549656
Random Forest	0.595960	0.603053	0.738318	0.663866	0.581596
XGBoost	0.590909	0.598485	0.738318	0.661088	0.614409
SVM	0.590909	0.597015	0.747664	0.663900	0.577488

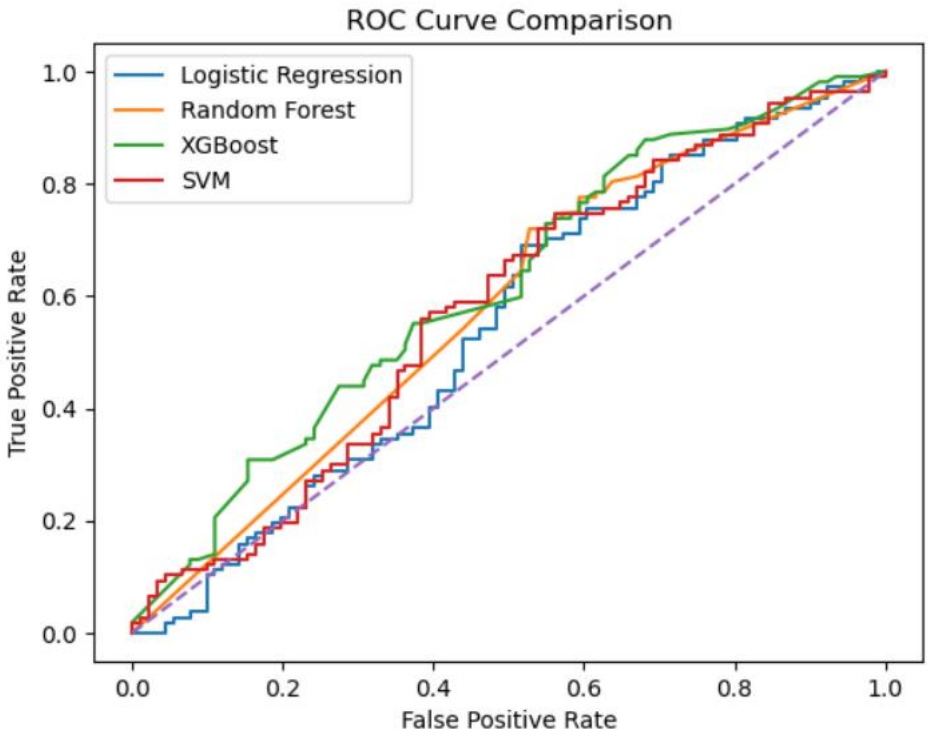
Although the Accuracy result is considered to select between different Predictive Models, but in our dataset example and models used, the Accuracy scores which are calculated by each model is very similar to each other, so to judge which model is performing best, we need to consider other parameters as well as the problem was not solved only by seeing Accuracy score.

The Recall values if we see, we can clearly say that SVM is performing better, it means predicting actual positive values, it is around 75%, still 25% Dengue cases are not detected but still SVM is better then other models here. In Precision column Random Forest is working best as there are less False Positive cases in comparison to other models. In F1 score which shows balance between Recall and precision, Random Forest and SVM are almost having similar values. In AUC, XGBoost performs best but still its value is less then 75% to have an impact, it means it is not able perform weak in distinguishing between Dengue and Non-Dengue cases.

So, in the above case all models are almost behaving equally but XGBoost performed a bit better if we consider AUC to select between models.

This study shows that in current dataset there are limited features to judge best model or to have better predictions, so clearly the dataset structure needs to be improved.

```python
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt

# Get probabilities
y_prob1 = model1.predict_proba(X_test)[:,1]
y_prob2 = model2.predict_proba(X_test)[:,1]
y_prob3 = model3.predict_proba(X_test)[:,1]
y_prob_svm = model_svm.predict_proba(X_test_scaled)[:,1]
# ROC values
fpr1, tpr1, _ = roc_curve(y_test, y_prob1)
fpr2, tpr2, _ = roc_curve(y_test, y_prob2)
fpr3, tpr3, _ = roc_curve(y_test, y_prob3)
fpr_svm, tpr_svm, _ = roc_curve(y_test, y_prob_svm)
# Plot
plt.figure()

plt.plot(fpr1, tpr1, label="Logistic Regression")
plt.plot(fpr2, tpr2, label="Random Forest")
plt.plot(fpr3, tpr3, label="XGBoost")
plt.plot(fpr_svm, tpr_svm, label="SVM")

plt.plot([0,1], [0,1], linestyle='--')  # random line

plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve Comparison")
plt.legend()

plt.show()
```



The ROC curve shows the above comparison table results in visual form; in this diagram we can see that all are having very close approach but still XGBoost is bit better than others.

But to have some meaningful results, the dataset needs to have more meaningful and symptomatic fields for better prediction.

Conclusion

The study focuses on application of Machine learning techniques using the demographic data and haematological data of patients. The study focused on certain models which is used as per applicability on used Dataset, the models used were Logistic Regression, Random Forest, XGBoost and SVM. The models were compared to each other by using standard performance metrics which contains accuracy score, precision score, recall score, F1-score, and AUC. Although the results of all the used models were having scores in all the fields very close to each other. After seeing all the scores in various heads, it was evaluated that XGBoost performed marginally better than the other models.

The results highlighted a fact that only Machine learning models itself will not be able to support any study or can do a better predictive modelling. It is the Dataset which takes the first importance, that how we have collected it, stored it and have we collected all required features which will help in the study or not.

The future research done in this area or the related area must include the additional clinical symptoms, patient history, and environmental factors, along with the steps included in this study, to enhance the performance of the results and have a greater impact.

Reference:

Chen, X., & Moraga, P. (2025). *Assessing dengue forecasting methods: A comparative study of statistical models and machine learning techniques in Rio de Janeiro, Brazil*. **Tropical Medicine and Health, 53**(1), 52. https://doi.org/10.1186/s41182-025-00723-7

Ferreira, J. R., Gutiérrez, J. B., & Duarte, F. (2019). *Predictive models for the medical diagnosis of dengue: A case study in Paraguay*. **Computational and Mathematical Methods in Medicine, 2019**, Article 7307806. https://doi.org/10.1155/2019/7307806

Hoyos, W., Aguilar, J., & Toro, M. (2021). *Dengue models based on machine learning techniques: A systematic literature review*. **Artificial Intelligence in Medicine, 119**, 102157. https://doi.org/10.1016/j.artmed.2021.102157

Masoud Pourzangiabadi, M., Najafi, H., Fallah, A., Goudarzi, A., & Pouladi, I. (2024). *Dengue virus: Etiology, epidemiology, pathobiology, and developments in diagnosis and control—A comprehensive review*. **Virus Research, 339**, 199348. https://doi.org/10.1016/j.virusres.2024.199348

Mayrose, H., Bairy, G. M., Sampathila, N., Belurkar, S., & Saravu, K. (2023). *Machine learning-based detection of dengue from blood smear images utilizing platelet and lymphocyte characteristics*. **Diagnostics, 13**(2), 220. https://doi.org/10.3390/diagnostics13020220

Nyenke, C. U., Nnokam, B. A., Esiere, R. K., & Nwalozie, R. (2023). *Dengue fever: Etiology, diagnosis, prevention and treatment*. **Asian Journal of Research in Infectious Diseases, 14**(1), 26–33. https://doi.org/10.9734/ajrid/2023/v14i1279

Patil, S. T., K. L., V., & N. G., K. (2019). *A study of clinical manifestations and complications of dengue fever in medical college hospital*. **International Journal of Medical Research and Review, 7**(3), 224–230. https://doi.org/10.17511/ijmrr.2019.i03.13

Potts, J. A., & Rothman, A. L. (2008). Clinical and laboratory features that distinguish dengue from other febrile illnesses in endemic populations. *Tropical medicine & international health : TM & IH*, *13*(11), 1328–1340. https://doi.org/10.1111/j.1365-3156.2008.02151.x

Schaefer, T. J., Panda, P. K., & Wolford, R. W. (2024). *Dengue fever*. In **StatPearls**. StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK430732/

https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue

https://www.kaggle.com/datasets/aravind3505/dengue-detection-dataset-clinical-data