AITECH
TRAINING & SOLUTIONS

"CardioCare: A Machine Learning-Enabled Android App for Heart Disease Risk prediction"

Problem Statement

- Develop a machine learning enabled app using a dataset comprising 12 features to forecast mortality risk in individuals with heart failure, aiming to enhance patient care and outcomes in cardiovascular healthcare.

Objectives

- The objective of developing a predictive model for forecasting mortality risk in individuals with heart failure is to provide healthcare professionals with a tool that can assist in identifying patients who are at higher risk of mortality.
- **Early Identification:** Identifying patients at higher risk of mortality allows healthcare providers to intervene early and provide targeted care to improve patient outcomes.
- **Personalized Care:** Predictive models can help tailor treatment plans to individual patients based on their specific risk profiles

AITECH
TRAINING & SOLUTIONS

Description

1.Age: Numerical. Represents the age of the individual.

2.Sex: Categorical. Indicates the gender of the individual. (M = Male, F = Female)

3.ChestPainType: **Categorical.** Describes the type of chest pain experienced.
(ATA = Atypical Angina, NAP = Non-Anginal Pain, ASY = Asymptomatic)

4. RestingBP: Numerical. Represents the resting blood pressure of the individual
(in mmHg).

5.Cholesterol: Numerical. Represents the cholesterol level of the individual (in mg/dL).

6.FastingBS: Categorical. Indicates whether fasting blood sugar is above 120 mg/dl or not. (0 = No, 1 = Yes)

AITECH
TRAINING & SOLUTIONS

7.RestingECG: Categorical. Describes the resting electrocardiographic results. (Normal, ST = ST-T Wave Abnormality)

8.MaxHR: Numerical. Represents the maximum heart rate achieved.

9.ExerciseAngina: Categorical. Indicates whether angina was induced by exercise. (N = No, Y = Yes)

10.Oldpeak: Numerical. ST depression induced by exercise relative to rest.

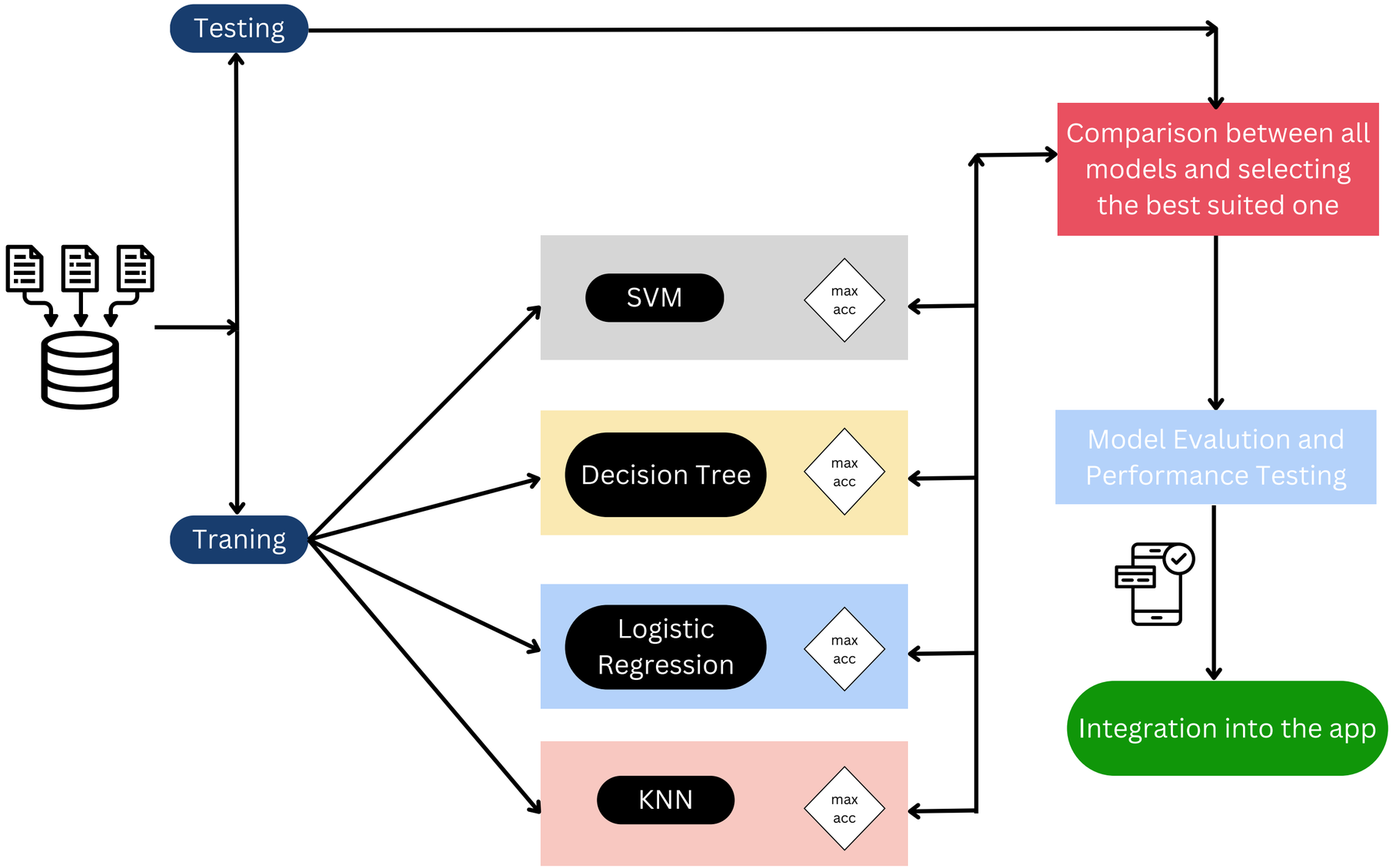
11.ST_Slope: Categorical. Describes the slope of the peak exercise ST segment.
(Up = Upsloping, Flat = Flat, Down = Downsloping)

12.HeartDisease: Categorical. Indicates the presence of heart disease. (0 = No, 1 = Yes)

Description

AITECH
TRAINING & SOLUTIONS

Flow-Diagram



AITECH TRAINING & SOLUTIONS

Step 1 : Summary Statistics

```
stats = df.describe(include = 'all')
stats
```

	Age	Sex	ChestPainType
count	918.000000	918	918
unique	NaN	2	4
top	NaN	M	ASY
freq	NaN	725	496
mean	53.510893	NaN	NaN
std	9.432617	NaN	NaN
min	28.000000	NaN	NaN
25%	47.000000	NaN	NaN
50%	54.000000	NaN	NaN
75%	60.000000	NaN	NaN
max	77.000000	NaN	NaN

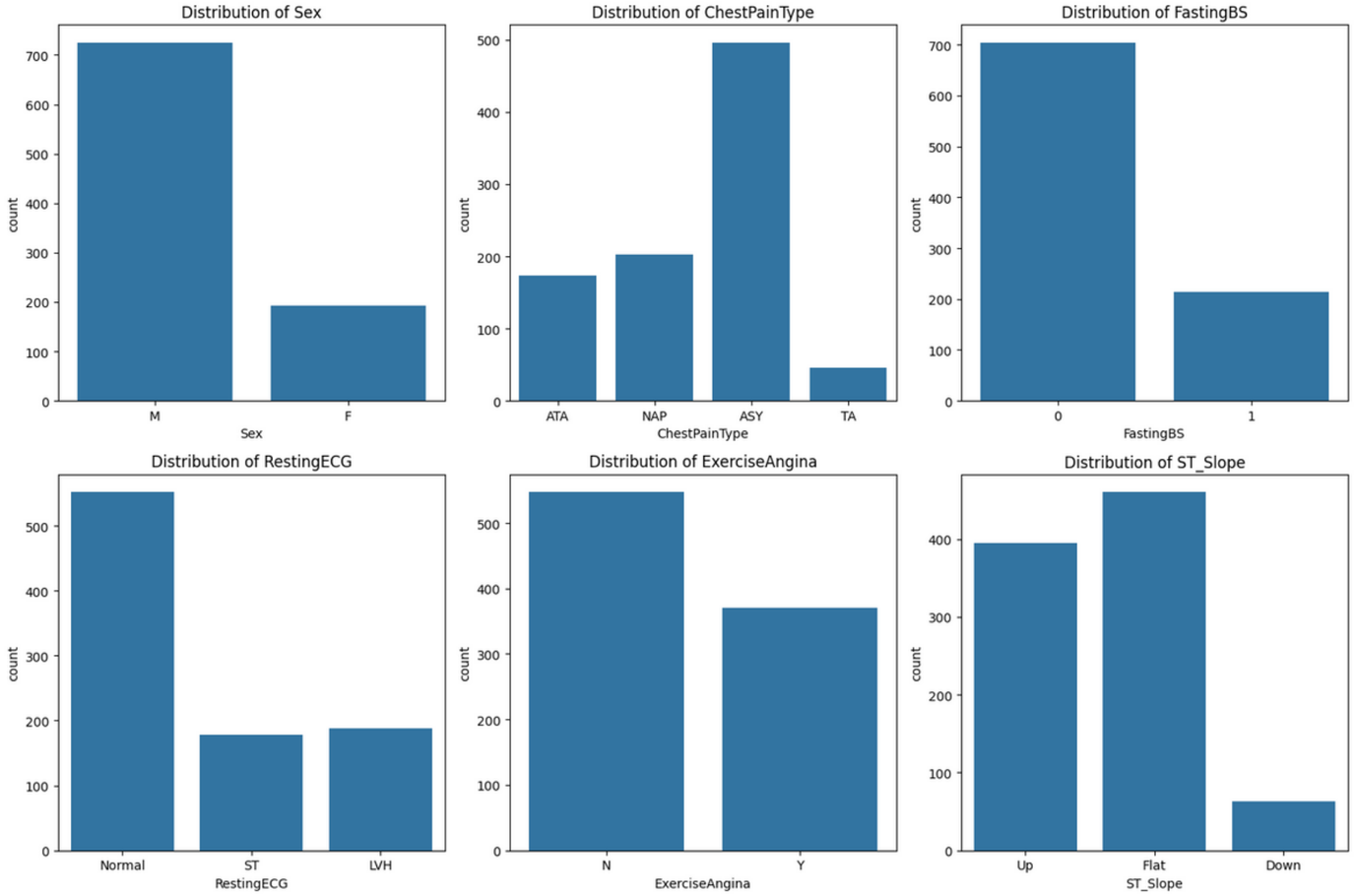
```
df.dtypes
```

```
Age                int64
Sex               object
ChestPainType     object
RestingBP        float64
Cholesterol      float64
FastingBS          int64
RestingECG        object
MaxHR            float64
ExerciseAngina    object
Oldpeak          float64
ST_Slope          object
HeartDisease       int64
dtype: object
```

- Gained insights of the dataset, including
- non-numerical columns

Data Exploration

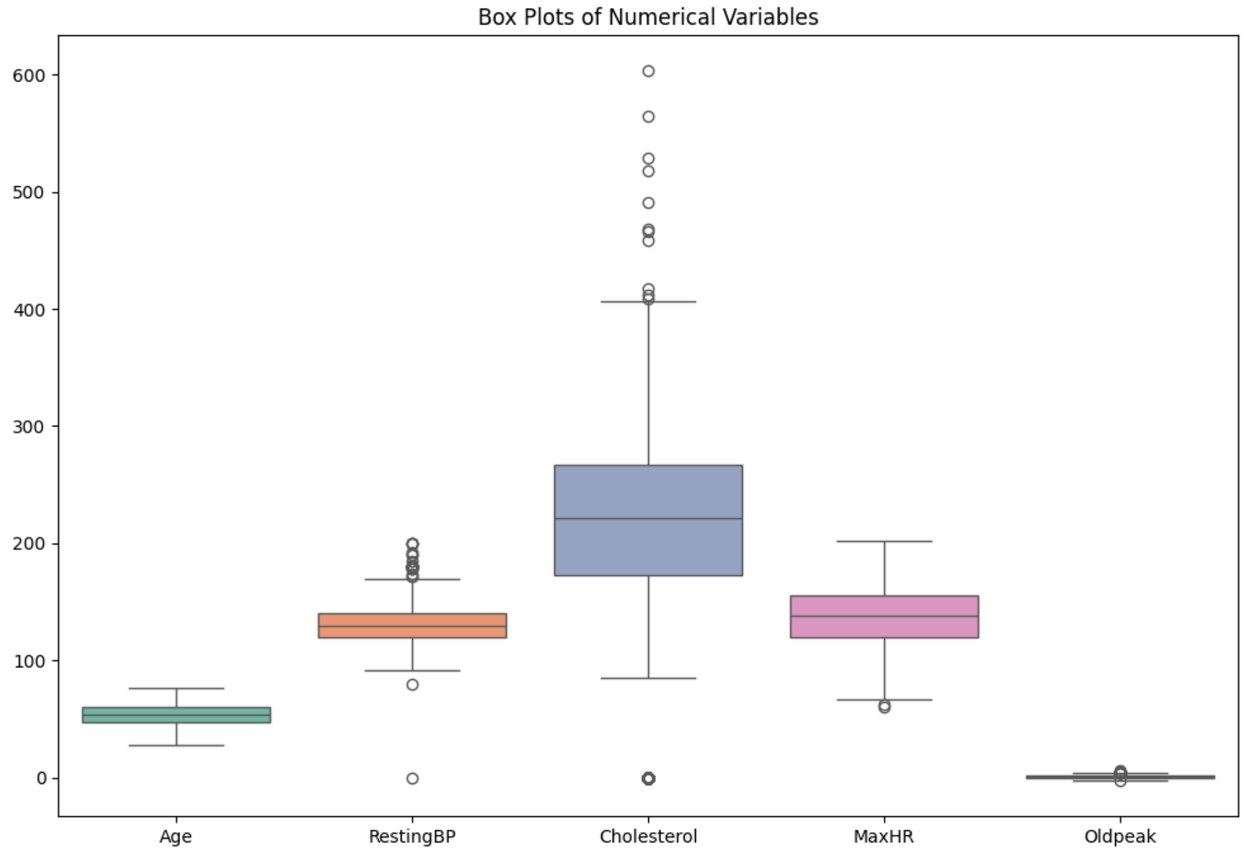
Step 2 : Data Visualization



- Obtaining Comparative count of categories for each Categorical Columns
- For ex: - In the attribute “Sex” records are present much more for the category “M” than “F”. “ExerciseAngina” is generally less observed in the dataset.

Data Visualization

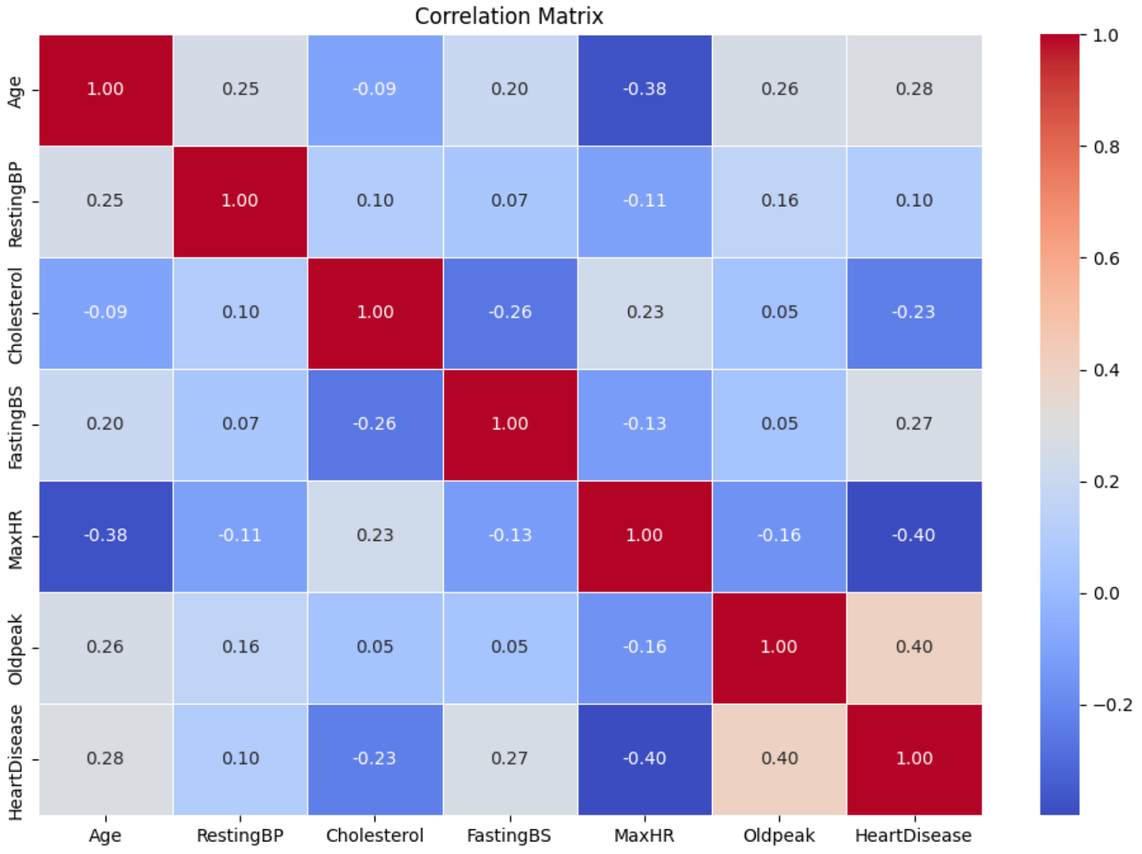
Step 3 : Data Distribution



- Observed how the data is distributed across Numerical Columns.
- In a box-plot , point beyond the whiskers typically indicate outliers
- We can observe , columns RestingBP , “Cholesterol” , “MaxHR” “Oldpeak” has outliers present.

Data Distribution

Step 4 : Correlation Analysis



- Plotting a heatmap to understand the relationships between various attributes.
- Positive co-relation usually indicates a directly proportional relationship and vice versa.
- It can be observed that, there is a proportional relationship all.

Correlation Analysis

Data Exploration

- Outliers are extreme values in a dataset, differing substantially from the majority of data points. There exist several method fo detecting outlier , we shall use Z-score Method
- Z-Score Method: Calculate the Z-score for each data point, which measures how many standard deviations a data point is from the mean. Set a threshold (e.g., Z > 3 or Z < -3) to identify data points beyond this range as outliers.

```
import pandas as pd
from scipy import stats
z_scores = stats.zscore(df[['Age', 'RestingBP', 'Cholesterol', 'MaxHR', 'Oldpeak']])
threshold = 3
outlier_indices = (abs(z_scores) > threshold).any(axis=1)
columns_with_outliers = df.loc[outlier_indices, ['Age', 'RestingBP', 'Cholesterol', 'MaxHR', 'Oldpeak'
columns_with_outliers
```

Outlier Detection

AITECH
TRAINING & SOLUTIONS

Step 5: Outlier Detection

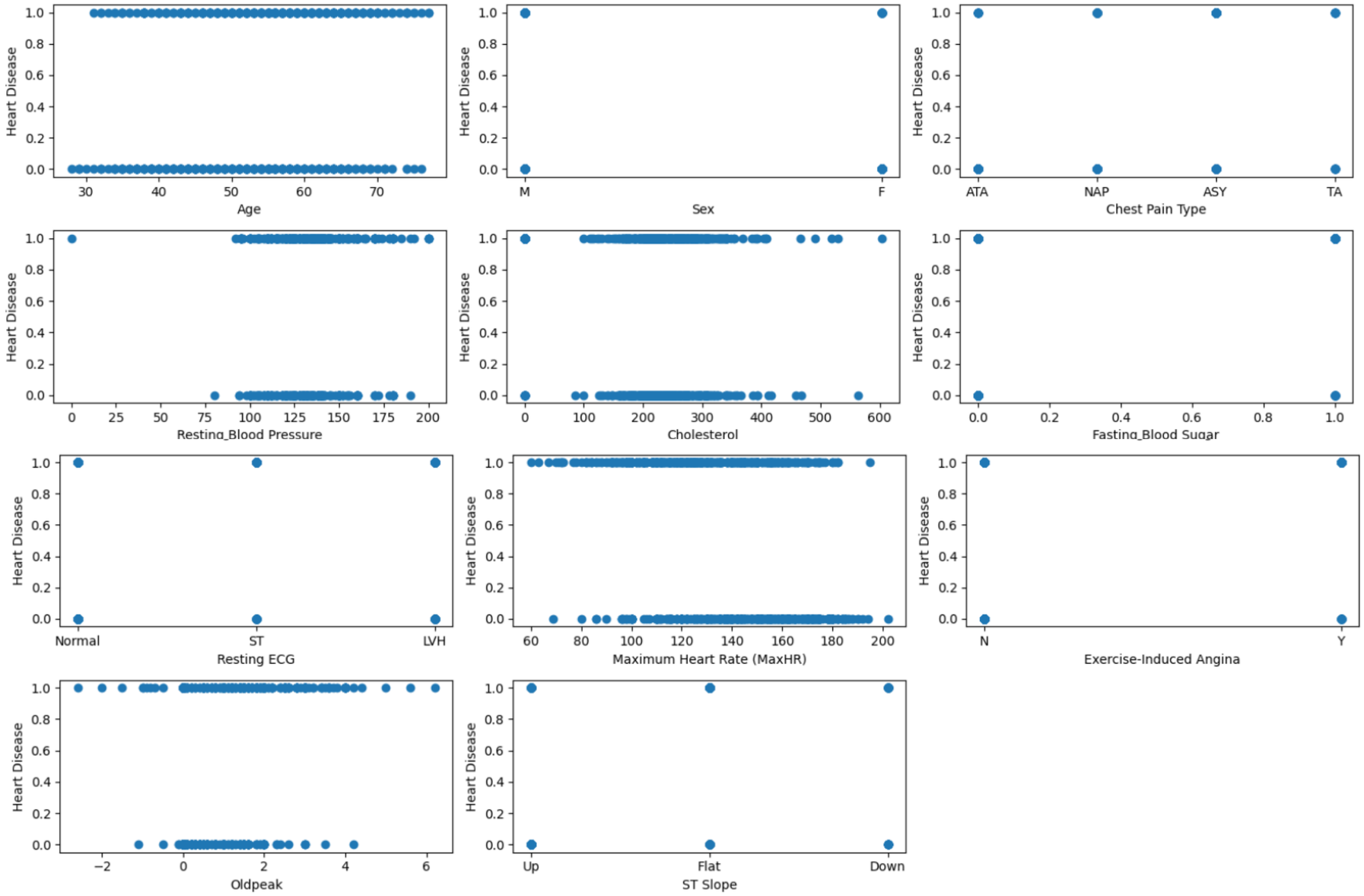
```
count = columns_with_outliers.shape[0]
print(count)
```

```
19
```

- It is necessary to handle outliers in accordance with business goals however since the amount of outliers being low , with no significant impact on the overall statistics in our dataset , we may choose to ignore them

Outlier handling

AITECH
TRAINING & SOLUTIONS



Feature-Target Relationship Visualization

AITECH
TRAINING & SOLUTIONS

Data Preprocessing

Step 1 : Data Cleaning

```
df['RestingBP'].fillna(120, inplace=True)
```

```
df['MaxHR'].fillna(220-df['Age'] , inplace=True)
```

```
df['Cholesterol'].fillna(df['Cholesterol'].median(), inplace=True)
```

```
df.isnull().sum()
```

```
Age               0
Sex               0
ChestPainType     0
RestingBP         0
Cholesterol       0
FastingBS         0
RestingECG        0
MaxHR             0
ExerciseAngina    0
Oldpeak           0
ST_Slope          0
HeartDisease      0
dtype: int64
```

- RestingBP's null values will be replaced by Normal RestingBP's value.
- MaxHR will be replaced by 220 - age.
- Cholesterol will be replaced by median as median is less affected by extreme values

AITECH
TRAINING & SOLUTIONS

Step 2 : Label Encoding

```
from sklearn.preprocessing import LabelEncoder
categorical_variables = ['Sex', 'ChestPainType', 'RestingECG', 'ExerciseAngina', 'ST_Slope']
label_encoder = LabelEncoder()
for var in categorical_variables:
    df[var] = label_encoder.fit_transform(df[var])

df.head()
```

- Encoding is done to convert categorical data into a numerical format suitable for machine learning algorithms. Label encoding assigns unique numeric labels to categories

- The categorical features, namely Sex, ChestPainType, FastingBS, RestingECG, ExerciseAngina, and ST_Slope, have been transformed using a label encoder for encoding purposes.

Data Preprocessing

AITECH
TRAINING & SOLUTIONS

Step 3: Data Splitting and Transformation.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
X = df.drop('HeartDisease', axis=1)
y = df['HeartDisease']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature Scaling using StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

- Data splitting ensures a model's performance is assessed on unseen data, while scaling or transforming data enhances algorithm efficiency by normalizing or standardizing features, improving model convergence and accuracy.

Data Splitting

AITECH
TRAINING & SOLUTIONS

Model Training

```
from sklearn.linear_model import LogisticRegression
lc= LogisticRegression(random_state=42)
lc.fit(X_train_scaled, y_train)
y_pred_lr = lc.predict(X_test_scaled)
```

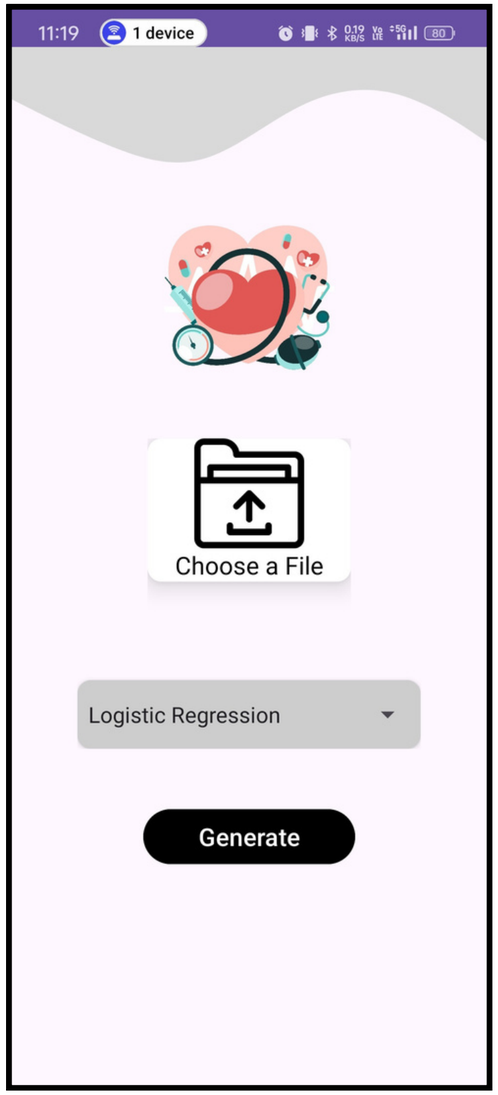
Results of all models

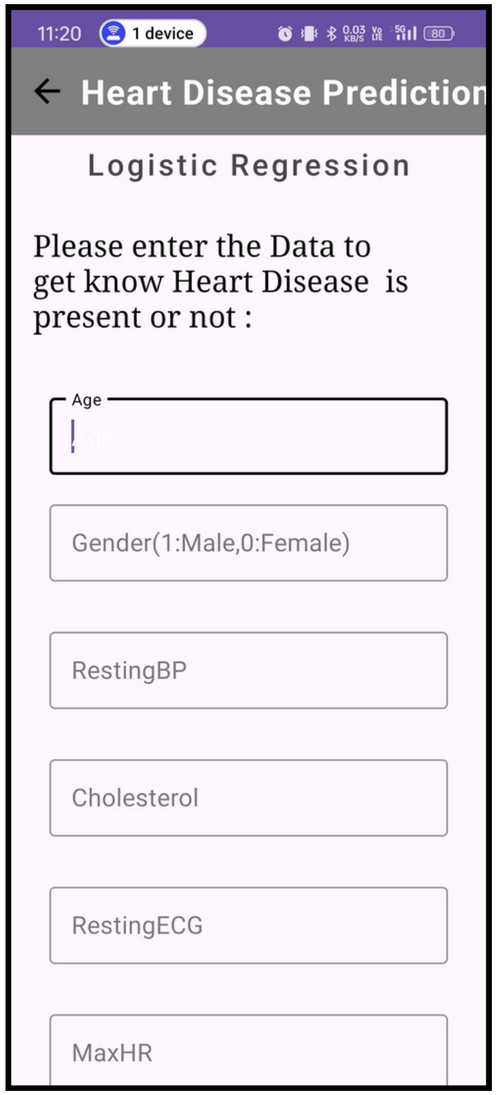
```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
def evaluate_model(y_true, y_pred, model_name):
    accuracy = accuracy_score(y_true, y_pred)
    print(f"\n{model_name} Accuracy: {accuracy:.4f}")
    print("Confusion Matrix:")
    print(confusion_matrix(y_true, y_pred))
    print("Classification Report:")
    print(classification_report(y_true, y_pred))

# Evaluate each model
evaluate_model(y_test, y_pred_lr, "Logistic Regression")
evaluate_model(y_test, y_pred_dt, "Decision Tree")
evaluate_model(y_test, y_pred_svm, "Support Vector Machine (SVM)")
evaluate_model(y_test, y_pred_knn, "K-Nearest Neighbors (KNN)")
```

Model Training

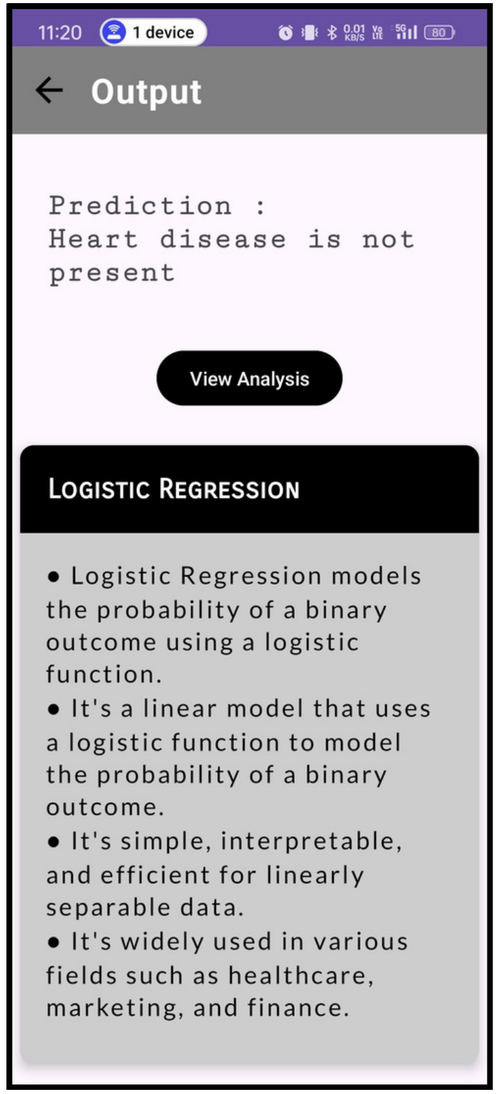
AITECH
TRAINING & SOLUTIONS

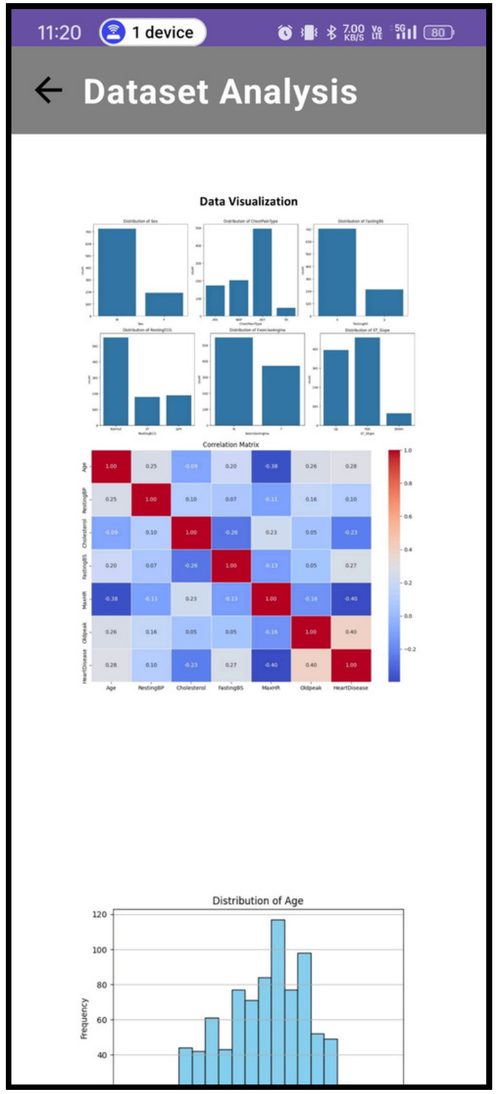




App Integration

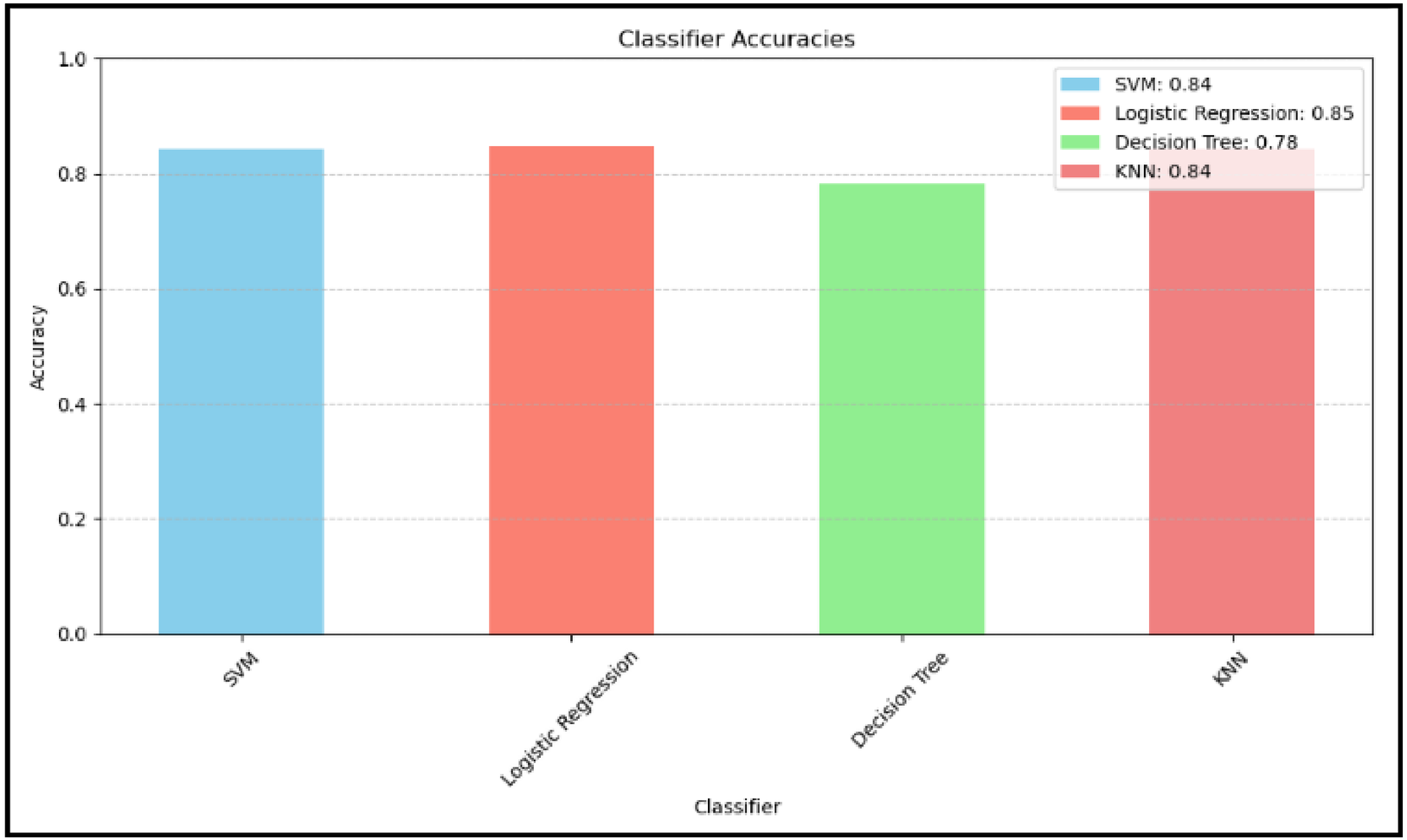
AITECH
TRAINING & SOLUTIONS





App Integration

Accuracy Of different models



Results

AITECH
TRAINING & SOLUTIONS

Results

Overall, the Support Vector Machine (SVM) model emerges as a formidable and precise tool for mortality risk prediction within the realm of cardiovascular healthcare. Its robust performance promises to significantly augment patient care and outcomes within clinical practice, ushering in a new era of proactive and personalized healthcare interventions.

Conclusion

In conclusion, CardioCare represents a significant step towards leveraging machine learning and mobile technology to empower individuals in managing their heart health and promoting overall well-being.

Thank You