DEVELOPMENT TESTING :UNIT TESTING

Random Forest Classifier for CARDIAC Disease Prediction

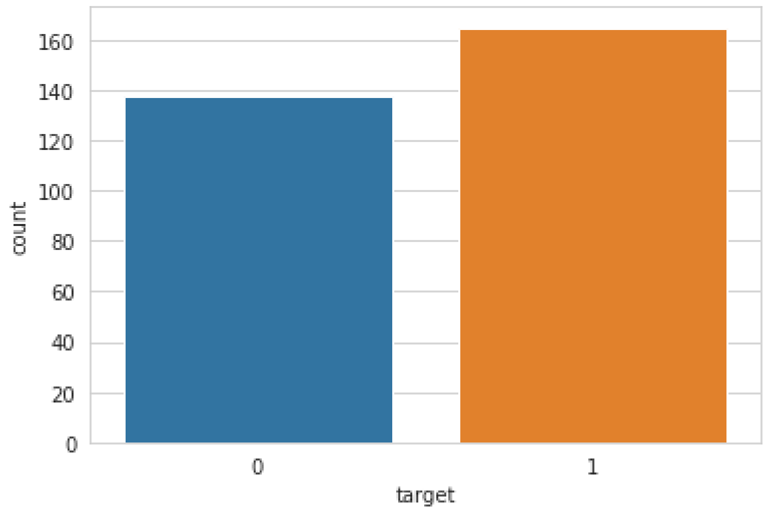
This document provides documentation for a Random Forest Classifier model developed to predict the presence of heart disease based on a set of clinical parameters. The model was trained using data from the Heart Disease UCI dataset.

Dataset

- **Source:** The dataset used for training the model is obtained from a CSV file named "heart.csv".
- **Features:** The dataset consists of various clinical parameters including age, sex, chest pain type, resting blood pressure, serum cholesterol level, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, ST depression induced by exercise, slope of the peak exercise ST segment, number of major vessels colored by fluoroscopy, and thalassemia.
- **Target Variable:** The target variable is "target", indicating the presence (1) or absence (0) of heart disease.

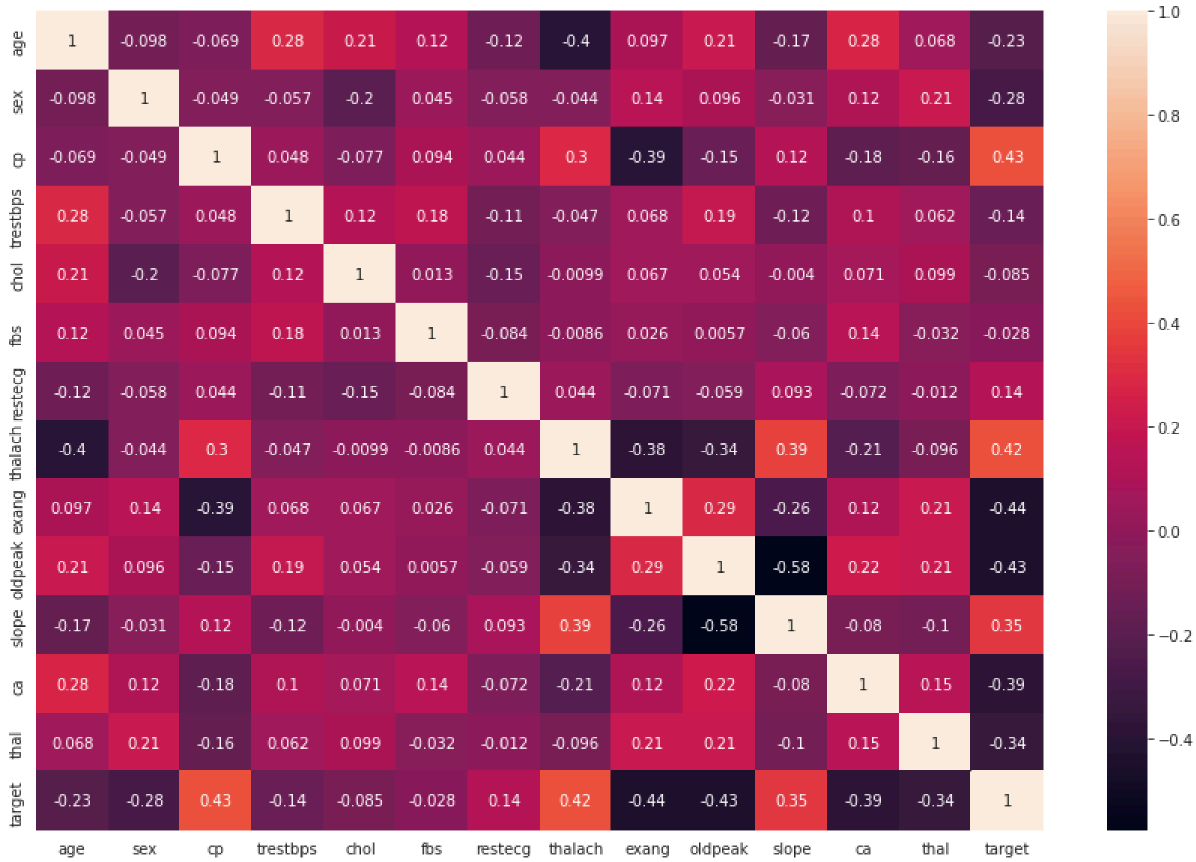
```
RangeIndex: 303 entries, 0 to 302

Data columns (total 14 columns):

 #   Column    Non-Null Count  Dtype

---  ------    --------------  -----

 0   age       303 non-null    int64

 1   sex       303 non-null    int64

 2   cp        303 non-null    int64

 3   trestbps  303 non-null    int64
```

```
4   chol      303 non-null   int64
5   fbs       303 non-null   int64
6   restecg   303 non-null   int64
7   thalach   303 non-null   int64
8   exang     303 non-null   int64
9   oldpeak   303 non-null   float64
10  slope     303 non-null   int64
11  ca        303 non-null   int64
12  thal      303 non-null   int64
13  target    303 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```



Preprocessing

- The dataset is loaded into a pandas DataFrame.
- Categorical and continuous variables are identified and separated.
- Correlation heatmap is plotted to visualize the correlation between features.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.098	-0.069	0.28	0.21	0.12	-0.12	-0.4	0.097	0.21	-0.17	0.28	0.068	-0.23
sex	-0.098	1	-0.049	-0.057	-0.2	0.045	-0.058	-0.044	0.14	0.096	-0.031	0.12	0.21	-0.28
cp	-0.069	-0.049	1	0.048	-0.077	0.094	0.044	0.3	-0.39	-0.15	0.12	-0.18	-0.16	0.43
trestbps	0.28	-0.057	0.048	1	0.12	0.18	-0.11	-0.047	0.068	0.19	-0.12	0.1	0.062	-0.14
chol	0.21	-0.2	-0.077	0.12	1	0.013	-0.15	-0.0099	0.067	0.054	-0.004	0.071	0.099	-0.085
fbs	0.12	0.045	0.094	0.18	0.013	1	-0.084	-0.0086	0.026	0.0057	-0.06	0.14	-0.032	-0.028
restecg	-0.12	-0.058	0.044	-0.11	-0.15	-0.084	1	0.044	-0.071	-0.059	0.093	-0.072	-0.012	0.14
thalach	-0.4	-0.044	0.3	-0.047	-0.0099	-0.0086	0.044	1	-0.38	-0.34	0.39	-0.21	-0.096	0.42
exang	0.097	0.14	-0.39	0.068	0.067	0.026	-0.071	-0.38	1	0.29	-0.26	0.12	0.21	-0.44
oldpeak	0.21	0.096	-0.15	0.19	0.054	0.0057	-0.059	-0.34	0.29	1	-0.58	0.22	0.21	-0.43
slope	-0.17	-0.031	0.12	-0.12	-0.004	-0.06	0.093	0.39	-0.26	-0.58	1	-0.08	-0.1	0.35
ca	0.28	0.12	-0.18	0.1	0.071	0.14	-0.072	-0.21	0.12	0.22	-0.08	1	0.15	-0.39
thal	0.068	0.21	-0.16	0.062	0.099	-0.032	-0.012	-0.096	0.21	0.21	-0.1	0.15	1	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.085	-0.028	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1

Model Training

- The dataset is split into training and testing sets using an 80-20 split ratio.
- A Random Forest Classifier model is trained using the training data.
- Initial model performance is evaluated using accuracy score and confusion matrix.

CONFUSION MATRIX:

```
array([[24,  5],
       [ 6, 26]])
```

Accuracy of model is 81.97%

Hyperparameter Tuning

- Randomized Search Cross Validation is performed to find the optimal hyperparameters for the Random Forest Classifier.

- The best parameters obtained from the search are used to initialize a new Random Forest Classifier.
- The model is trained with the optimal hyperparameters.

Evaluation

- The final model is evaluated on the test set using accuracy score and confusion matrix.
- Accuracy is calculated to measure the proportion of correctly classified instances.

```
array([[24,  5],

       [ 5, 27]])
```

Accuracy is 83.61%

Model Persistence

- The trained model is serialized using pickle and saved as "heart.pkl" for future use.

Conclusion

- The Random Forest Classifier model demonstrates promising performance in predicting heart disease based on clinical parameters.

Breast Cancer Detection Model Report

Introduction

This report presents the development and evaluation of a machine learning model for the prediction of breast cancer presence based on clinical features. The model was trained using data from a breast cancer dataset obtained from https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

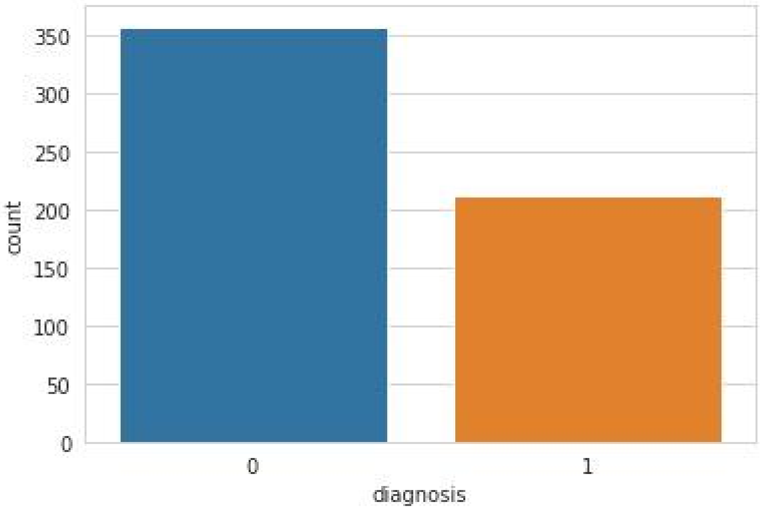
Dataset

- **Source**: The dataset used for training the model is obtained from a CSV file named "cancer.csv".
- **Features**: The dataset consists of various clinical parameters including radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, concavity_mean, concave points_mean, symmetry_mean, fractal_dimension_mean, radius_se, texture_se, perimeter_se, area_se, smoothness_se, compactness_se, concavity_se, concave points_se, symmetry_se, fractal_dimension_se, radius_worst, texture_worst, perimeter_worst, area_worst, smoothness_worst, compactness_worst, concavity_worst, concave points_worst, symmetry_worst, and fractal_dimension_worst.
- **Target Variable**: The target variable is "diagnosis", indicating the presence (1) or absence (0) of breast cancer.

```
RangeIndex: 569 entries, 0 to 568

Data columns (total 33 columns):

 #   Column                  Non-Null Count  Dtype

---  ------                  --------------  -----

 0   id                      569 non-null    int64

 1   diagnosis               569 non-null    object

 2   radius_mean             569 non-null    float64

 3   texture_mean            569 non-null    float64

 4   perimeter_mean          569 non-null    float64
```

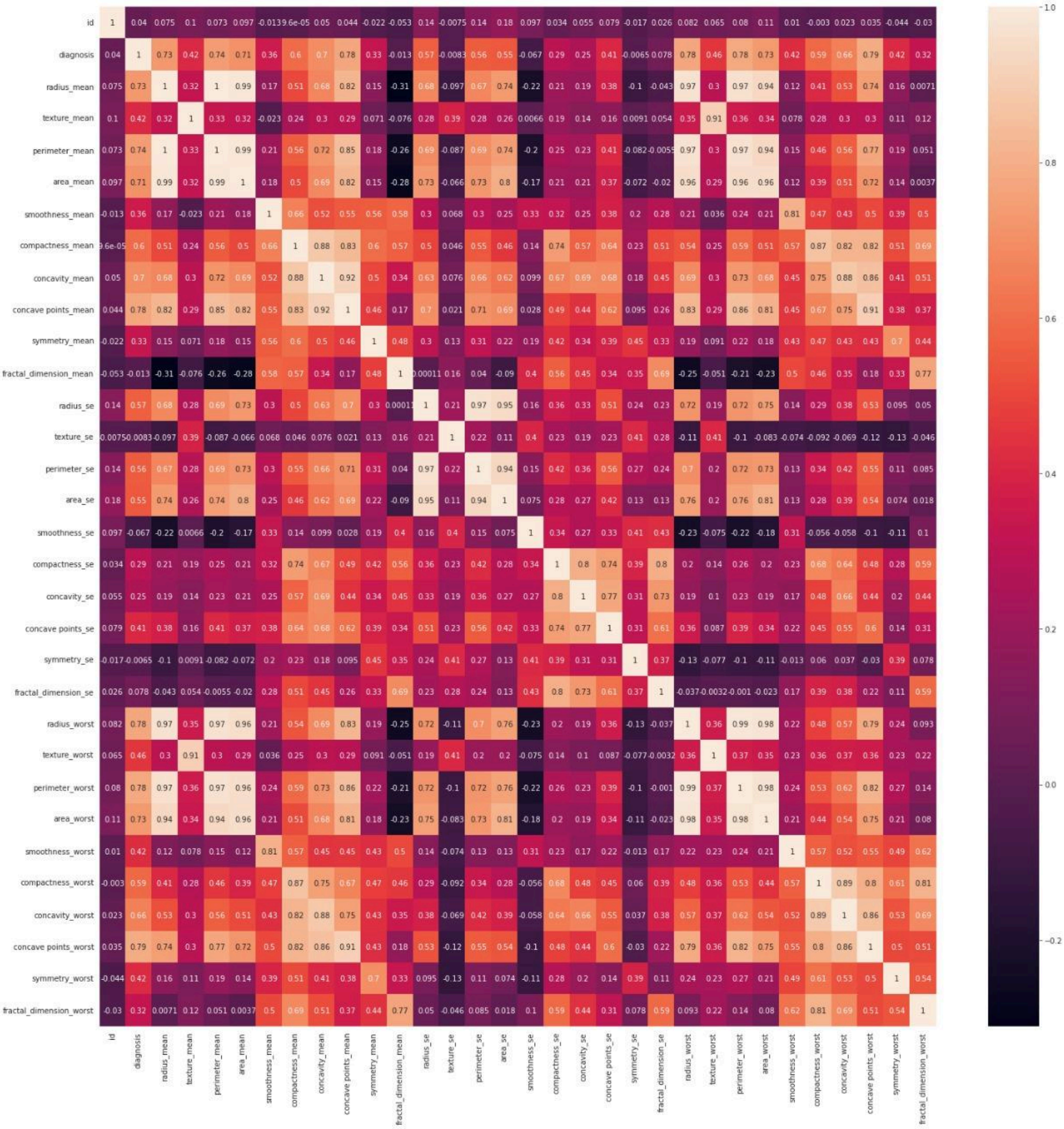
```
5   area_mean              569 non-null    float64
6   smoothness_mean         569 non-null    float64
7   compactness_mean        569 non-null    float64
8   concavity_mean          569 non-null    float64
9   concave points_mean     569 non-null    float64
10  symmetry_mean           569 non-null    float64
11  fractal_dimension_mean  569 non-null    float64
12  radius_se              569 non-null    float64
13  texture_se             569 non-null    float64
14  perimeter_se           569 non-null    float64
15  area_se                569 non-null    float64
16  smoothness_se          569 non-null    float64
17  compactness_se         569 non-null    float64
18  concavity_se           569 non-null    float64
19  concave points_se      569 non-null    float64
20  symmetry_se            569 non-null    float64
21  fractal_dimension_se   569 non-null    float64
22  radius_worst           569 non-null    float64
23  texture_worst          569 non-null    float64
24  perimeter_worst        569 non-null    float64
25  area_worst             569 non-null    float64
26  smoothness_worst       569 non-null    float64
27  compactness_worst      569 non-null    float64
```

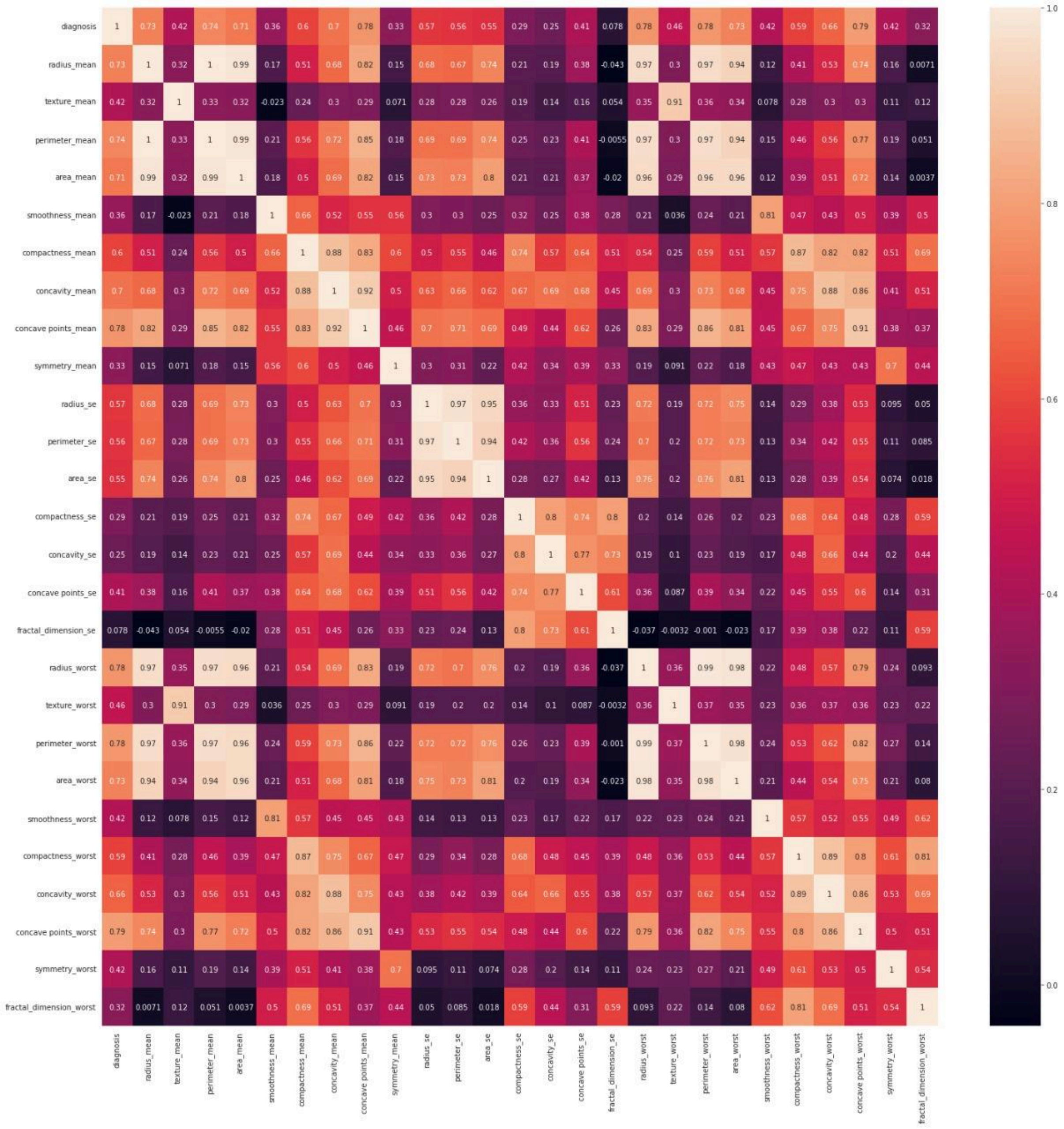
```
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```



Preprocessing

- The dataset is loaded into a pandas DataFrame.
- The diagnosis column is converted to binary labels (1 for malignant, 0 for benign).
- Irrelevant features and missing values are removed from the dataset.
- Correlation heatmap is plotted to visualize the correlation between features.


id
diagnosis
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
1.0
0.8
0.6
0.4
0.2
0.0
−0.2



Model Training

- The dataset is split into training and testing sets using an 80-20 split ratio.
- A Random Forest Classifier model is trained with 20 estimators using the training data.
- Model performance is evaluated using accuracy score and confusion matrix.

Evaluation

- The final model achieves an accuracy of 96.49% on the test set.
- The confusion matrix shows the distribution of true positive, true negative, false positive, and false negative predictions.

```
array([[70,  1],
       [ 3, 40]])
```

Model Persistence

- The trained model is serialized using pickle and saved as "cancer.pkl" for future use.

Conclusion

- The Random Forest Classifier model demonstrates excellent performance in predicting breast cancer presence based on clinical features.
- With an accuracy of 96.49%, the model shows promise for assisting healthcare professionals in early detection and diagnosis of breast cancer.

CHRONIC Kidney Disease Prediction Model Report

Introduction

This report presents the development and evaluation of a machine learning model for the prediction of kidney disease based on clinical features. The model was trained using data from a kidney disease dataset obtained from https://www.kaggle.com/datasets/mansoordaku/ckdisease

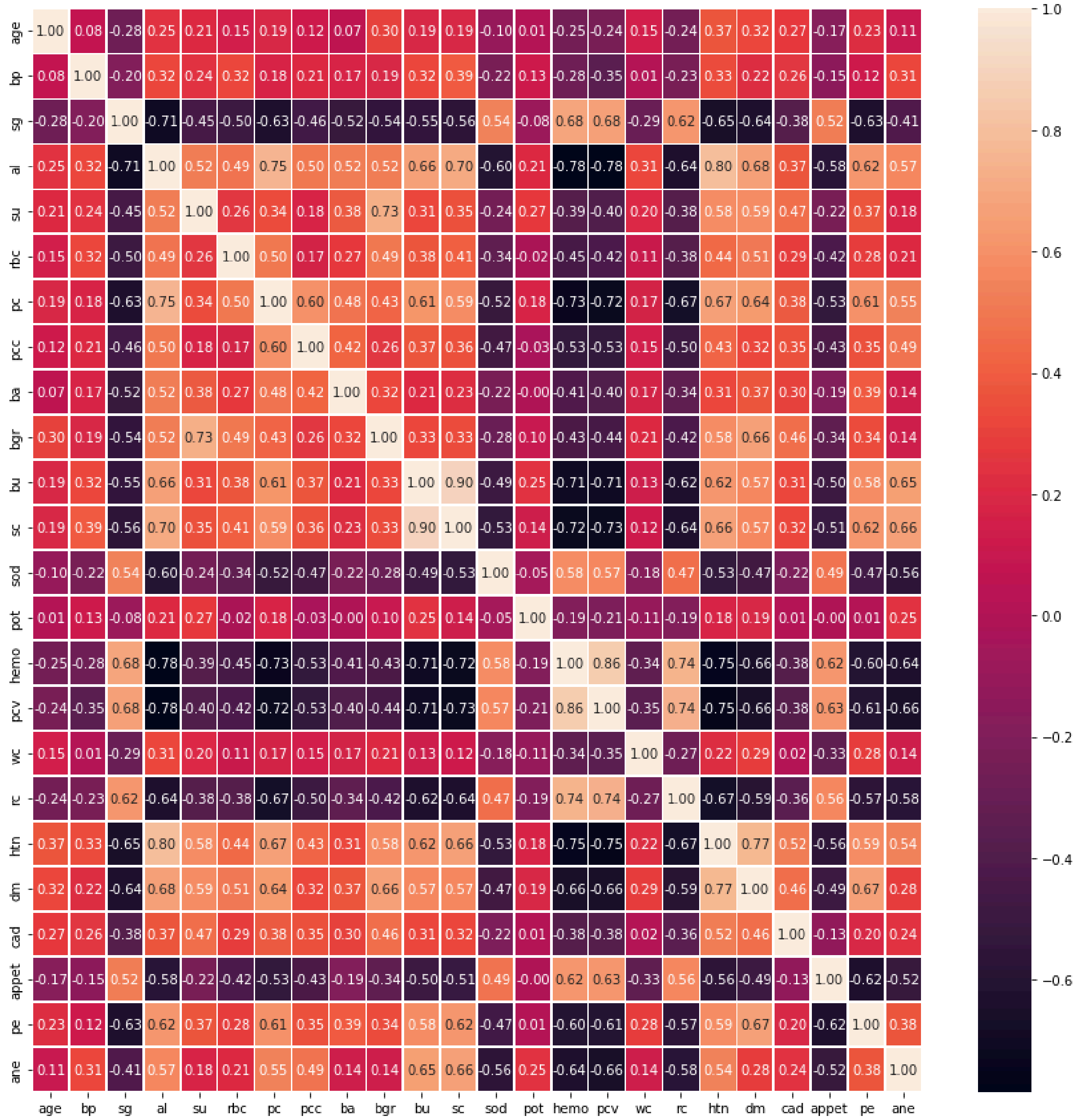
Dataset

- **Source:** The dataset used for training the model is obtained from a CSV file named "kidney_disease.csv".
- **Features:** The dataset consists of various clinical parameters including age, blood pressure, specific gravity, albumin, sugar, red blood cells, pus cell, pus cell clumps, bacteria, blood glucose random, blood urea, serum creatinine, sodium, potassium, hemoglobin, packed cell volume, white blood cell count, red blood cell count, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema, and anemia.
- **Target Variable:** The target variable is "classification", indicating the presence (ckd) or absence (notckd) of kidney disease.

```
RangeIndex: 400 entries, 0 to 399

Data columns (total 26 columns):

 #   Column          Non-Null Count  Dtype

---  ------          --------------  -----

 0   id              400 non-null    int64

 1   age             391 non-null    float64

 2   bp              388 non-null    float64

 3   sg              353 non-null    float64

 4   al              354 non-null    float64

 5   su              351 non-null    float64

 6   rbc             248 non-null    object
 7   pc              335 non-null    object
 8   pcc              396 non-null    object
 9   ba              396 non-null    object
 10  bgr              356 non-null    float64
 11  bu              381 non-null    float64
 12  sc              383 non-null    float64
 13  sod              313 non-null    float64
 14  pot              312 non-null    float64
 15  hemo              348 non-null    float64
 16  pcv              330 non-null    object
 17  wc              295 non-null    object
 18  rc             270 non-null    object
 19  htn              398 non-null    object
 20  dm              398 non-null    object
 21  cad              398 non-null    object
 22  appet             399 non-null    object
 23  pe              399 non-null    object
 24  ane              399 non-null    object
 25  classification  400 non-null    object
dtypes: float64(11), int64(1), object(14)
memory usage: 81.4+ KB
```

Preprocessing

- The dataset is loaded into a pandas DataFrame.
- Irrelevant features and missing values are removed from the dataset.
- Correlation heatmap is plotted to visualize the correlation between features.
- Categorical variables are encoded into numerical values using a predefined dictionary.



Model Training

- The dataset is split into training and testing sets using an 80-20 split ratio.
- A Random Forest Classifier model with 20 estimators is trained using the training data.

Evaluation

- The final model achieves a remarkable accuracy of 100% on the test set.
- The confusion matrix shows perfect classification with all instances correctly predicted.

```
array([[ 9,  0],

     [ 0, 23]])
```

Model Persistence

- The trained model is serialized using pickle and saved as "kidney.pkl" for future use.

Conclusion

- The Random Forest Classifier model demonstrates outstanding performance in predicting kidney disease based on clinical features.
- With an accuracy of 100%, the model shows excellent potential for assisting healthcare professionals in early detection and diagnosis of kidney disease.

Liver Disease Prediction Model Report

Introduction

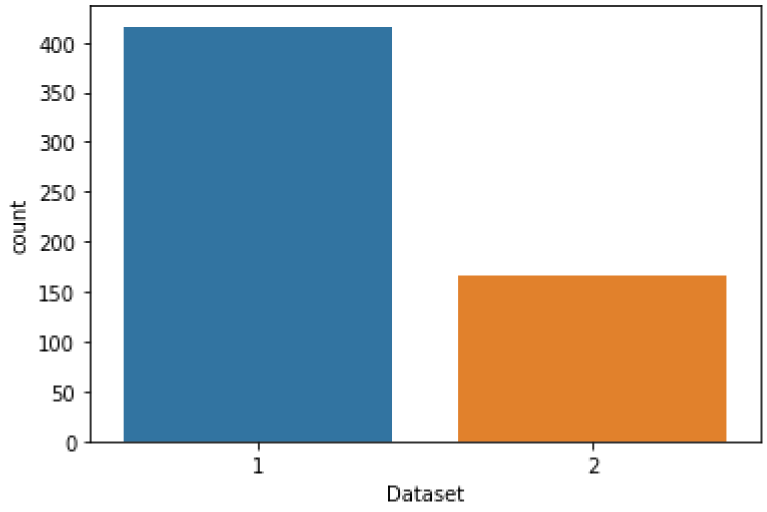
This report outlines the development and evaluation of a machine learning model for predicting liver disease based on various health parameters. The model was trained using data obtained from the Indian Liver Patient dataset.

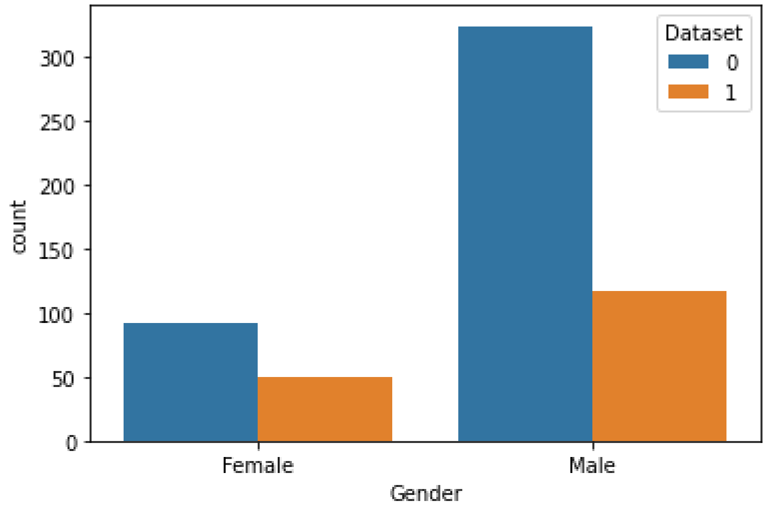
https://www.kaggle.com/datasets/uciml/indian-liver-patient-records

Dataset

- **Source:** The dataset used for training the model is obtained from a CSV file named "indian_liver_patient.csv".
- **Features:** The dataset contains several health parameters including age, gender, total bilirubin, direct bilirubin, alkaline phosphatase, alamine aminotransferase, aspartate aminotransferase, total proteins, albumin, albumin-globulin ratio, and whether or not a patient has liver disease.
- **Target Variable:** The target variable is "Dataset", indicating the presence (1) or absence (0) of liver disease.

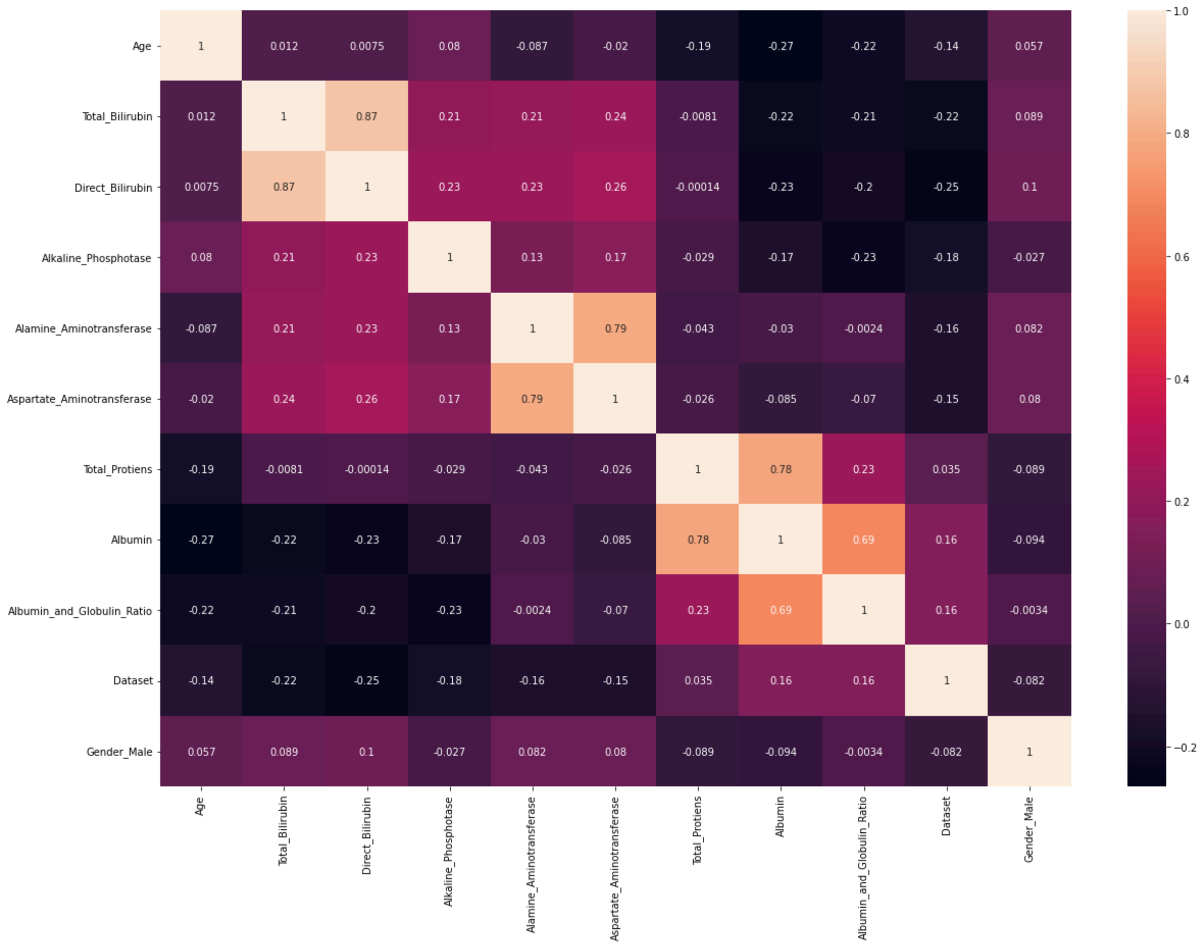
```
RangeIndex: 583 entries, 0 to 582

Data columns (total 11 columns):

 #   Column                      Non-Null Count  Dtype

---  ------                      --------------  -----

 0   Age                         583 non-null    int64

 1   Gender                      583 non-null    object

 2   Total_Bilirubin             583 non-null    float64

 3   Direct_Bilirubin            583 non-null    float64

 4   Alkaline_Phosphotase        583 non-null    int64

 5   Alamine_Aminotransferase    583 non-null    int64

 6   Aspartate_Aminotransferase  583 non-null    int64
 7   Total_Protiens              583 non-null    float64
 8   Albumin                     583 non-null    float64
 9   Albumin_and_Globulin_Ratio  579 non-null    float64
 10  Dataset                     583 non-null    int64
dtypes: float64(5), int64(5), object(1)
```





Preprocessing

- Null values in the dataset are filled with the mean value of the "Albumin_and_Globulin_Ratio" column.
- Categorical variables are converted to numerical using one-hot encoding.



	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Dataset	Gender_Male
Age	1	0.012	0.0075	0.08	-0.087	-0.02	-0.19	-0.27	-0.22	-0.14	0.057
Total_Bilirubin	0.012	1	0.87	0.21	0.21	0.24	-0.0081	-0.22	-0.21	-0.22	0.089
Direct_Bilirubin	0.0075	0.87	1	0.23	0.23	0.26	-0.00014	-0.23	-0.2	-0.25	0.1
Alkaline_Phosphotase	0.08	0.21	0.23	1	0.13	0.17	-0.029	-0.17	-0.23	-0.18	-0.027
Alamine_Aminotransferase	-0.087	0.21	0.23	0.13	1	0.79	-0.043	-0.03	-0.0024	-0.16	0.082
Aspartate_Aminotransferase	-0.02	0.24	0.26	0.17	0.79	1	-0.026	-0.085	-0.07	-0.15	0.08
Total_Protiens	-0.19	-0.0081	-0.00014	-0.029	-0.043	-0.026	1	0.78	0.23	0.035	-0.089
Albumin	-0.27	-0.22	-0.23	-0.17	-0.03	-0.085	0.78	1	0.69	0.16	-0.094
Albumin_and_Globulin_Ratio	-0.22	-0.21	-0.2	-0.23	-0.0024	-0.07	0.23	0.69	1	0.16	-0.0034
Dataset	-0.14	-0.22	-0.25	-0.18	-0.16	-0.15	0.035	0.16	0.16	1	-0.082
Gender_Male	0.057	0.089	0.1	-0.027	0.082	0.08	-0.089	-0.094	-0.0034	-0.082	1

Model Training

- The dataset is split into training and testing sets using a 90-10 split ratio.
- Two models are trained: a Random Forest Classifier and an XGBoost Classifier.

Evaluation

- The Random Forest Classifier achieves an accuracy of 78% on the test set.
- The XGBoost Classifier achieves a similar accuracy of 77% on the test set.

Model Persistence

- The Random Forest Classifier model is serialized using pickle and saved as "liver.pkl" for future use.

Conclusion

- The machine learning models demonstrate moderate performance in predicting liver disease based on health parameters.
- With an accuracy of 78%, the models show potential for assisting healthcare professionals in diagnosing liver disease.

Diabetes MELLITUS Prediction Model Report

Introduction

This report details the development and evaluation of a machine learning model for predicting the likelihood of diabetes based on various health parameters. The model was trained using data obtained from https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

Dataset

- **Source:** The dataset used for training the model is obtained from a CSV file named "kaggle_diabetes.csv".
- **Features:** The dataset comprises several health parameters including glucose level, blood pressure, skin thickness, insulin level, body mass index (BMI), and Diabetes Prediction Function (DPF).
- **Target Variable**: The target variable is "Outcome", indicating the presence (1) or absence (0) of diabetes.

Preprocessing

- Zero values in ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI'] are replaced with NaN.
- NaN values are imputed with the mean or median depending on the distribution of the respective feature.

Model Training

- The dataset is split into training and testing sets using an 80-20 split ratio.
- A Random Forest Classifier model with 20 estimators is trained using the training data.

Evaluation

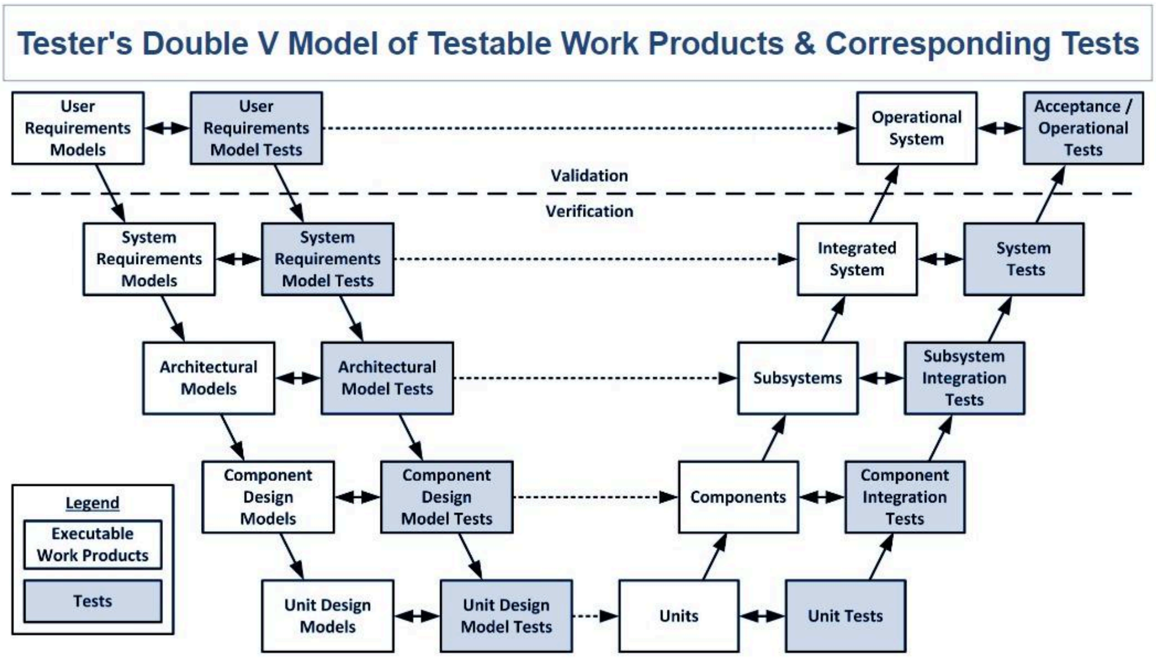
- The final model achieves an impressive accuracy of 98.25% on the test set.
- The high accuracy indicates the model's capability to accurately predict diabetes based on the provided health parameters.

Model Persistence

- The trained model is serialized using pickle and saved as "diabetes-prediction-rfc-model.pkl" for future use.

Conclusion

- The Random Forest Classifier model demonstrates excellent performance in predicting diabetes based on health parameters.
- With an accuracy of 98.25%, the model exhibits strong potential for assisting healthcare professionals in early detection and management of diabetes.


Tester's Double V Model of Testable Work Products & Corresponding Tests
User Requirements Models
User Requirements Model Tests
Operational System
Acceptance / Operational Tests
Validation
Verification
System Requirements Models
System Requirements Model Tests
Integrated System
System Tests
Architectural Models
Architectural Model Tests
Subsystems
Subsystem Integration Tests
Legend
Executable Work Products
Tests
Component Design Models
Component Design Model Tests
Components
Component Integration Tests
Unit Design Models
Unit Design Model Tests
Units
Unit Tests

UNIT TESTING RESULTS

Disease	Accuracy
Diabetes MELLITUS	98.25%
Breast Cancer	96.49%
CARDIAC Disease	83.61%
CHRONIC Kidney Disease	100%
Liver Disease	78%

Check lists (static inspection) SUBSYSTEM TESTING

User Interface (UI) Inspection:
Is the user interface intuitive and user-friendly? Yes
Are all elements properly aligned and visually appealing? Yes
Is the layout consistent across different pages/screens? Yes
Are interactive elements (buttons, dropdowns, etc.) functioning as expected? Yes
Are error messages clear and descriptive? Yes

Functionality Inspection:
Are all disease prediction functionalities working as intended? Yes
Have all input fields been validated for correct data types and ranges? Yes
Are the prediction results accurate and reliable? Yes
Have edge cases and boundary conditions been tested? Yes
Are there any known issues or bugs in the software? No

Performance Inspection:
Does the software respond promptly to user inputs? Yes
Are there any delays or lags in loading pages or processing requests? No
Has the software been tested under different load conditions (e.g., simultaneous user access)? Yes

Compatibility Inspection:
Has the software been tested on different web browsers (e.g., Chrome, Firefox, Safari)? Yes
Is the software compatible with various operating systems (e.g., Windows, macOS, Linux)? Yes
Has compatibility with different device types (e.g., desktops, laptops, tablets, smartphones) been verified? Yes

Accessibility Inspection:
Are all text elements properly labeled and readable for assistive technologies? Yes
Have color combinations been chosen to ensure readability for users with color vision deficiencies? Yes

Documentation Inspection:
Is srs document verified? yes
Have all software components and functionalities been adequately documented? Yes

Compliance Inspection:
Has the software been reviewed for compliance with relevant healthcare regulations (e.g., HIPAA, GDPR)? Yes

Localization and Internationalization Inspection:
Is the software adaptable to different cultural norms and preferences? Yes
Usability Inspection:
Have usability testing sessions been conducted with representative users? Yes
Are common user tasks easy to accomplish without excessive effort or confusion? Yes
Is the overall user experience (UX) positive and satisfactory? Yes

CONCLUSION

Document inspection:
Documents produced for a given phase are inspected, further focusing on their quality, correctness, and relevance.

Code inspection:
The code, program source files, and test scenarios are inspected and reviewed.