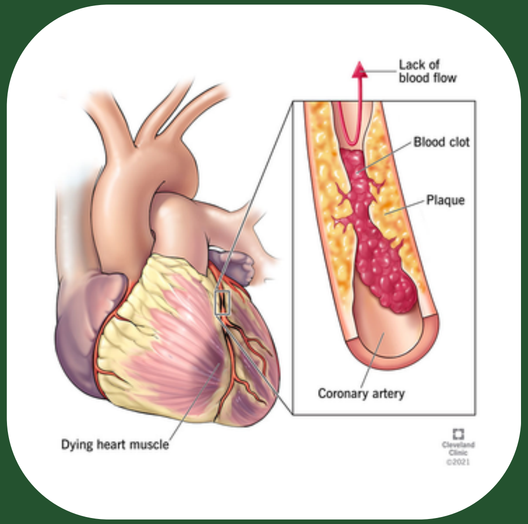
3 GOOD HEALTH AND WELL-BEING

Determination of Ventricular Fibrillation Complication after Myocardial Infarction

PATTERN RECOGNITION CLASS - FINAL PROJECT

Sasha Annabel 22/496780/PA/21361

Motivation



A myocardial infarction (MI), commonly known as a HEART ATTACK, occurs when blood flow decreases (or stops) to a part of the heart, thus causing damage to the heart muscle. The most common symptom is chest pain, but there are many occasions where symptoms do not develop at all.

" Among those over 75 years old, about 5% have had an MI with little or <u>no history of symptoms</u> "

This is **dangerous** because an MI may cause complications such as heart failure, irregular heartbeat, even cardiogenic shock or cardiac arrest. It is important to find the "priority signs" through recorded medical data to predict possible complications as soon as a patient is admitted for MI.

Project Goal

3 GOOD HEALTH AND WELL-BEING

Although there are many dangerous complications of heart attack (MI), this project will focus on **one of the most lethal complications** :

Ventricular Fibrillation

Called VF in short, it is a life-threatening *arrhythmia* that can lead to sudden cardiac death if not treated immediately, meaning that early detection and prevention are critical for patient survival.

The project's goal is to be able to identify
through certain medical data whether the patient will develop VF or not.

Proposed Methods

and their keywords.

DATA ACQUISITION

Kaggle, CSV, GoogleColab Notebook

PRE-PROCESSING

Null values, target feature

FEATURE SELECTION/EXTRACTION

Random Forest, relevant features

TRAINING/TESTING CLASSIFIER MODEL

KNN classifier, data splitting

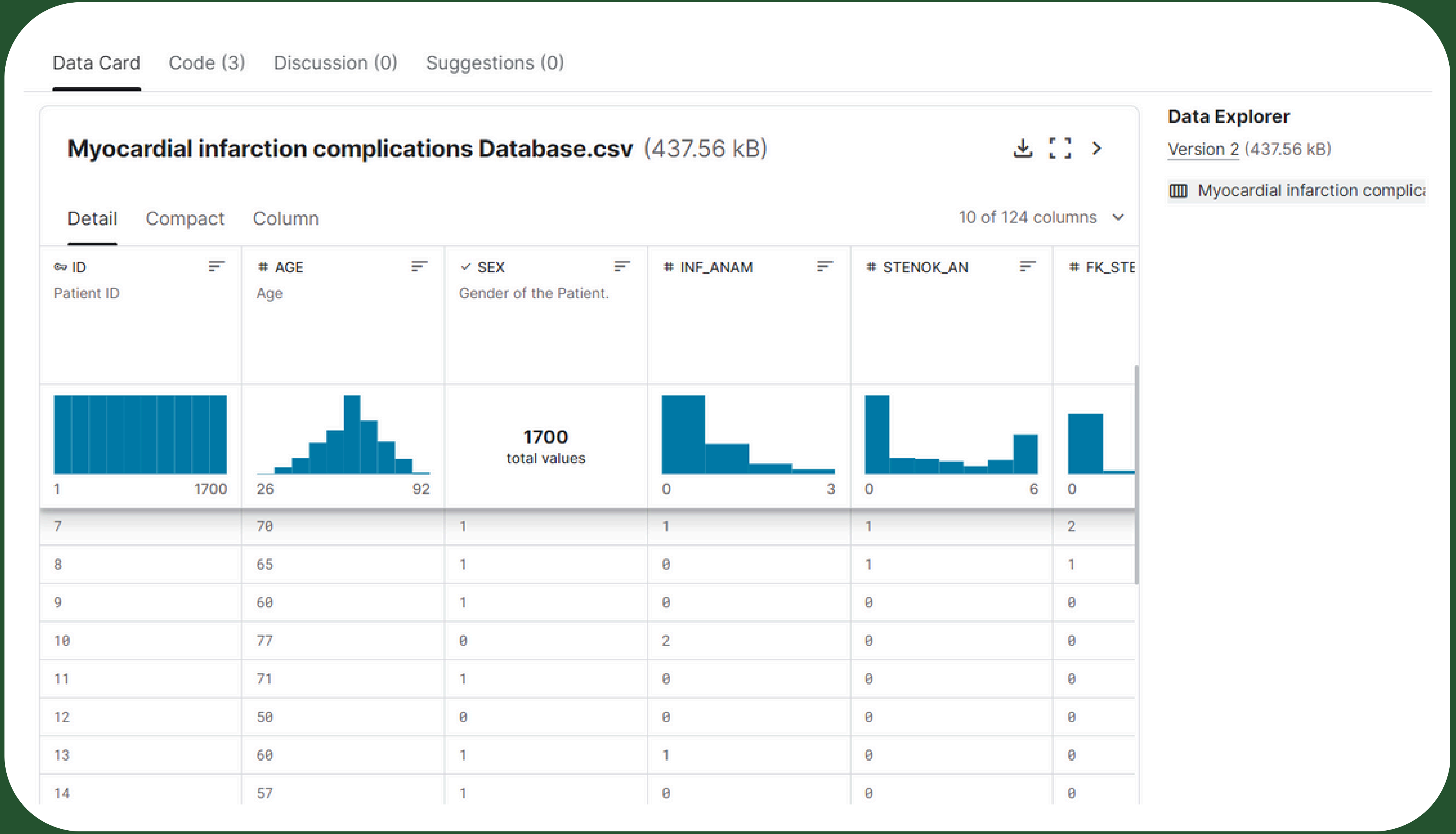
EVALUATION

Accuracy, grid search cross-validation

Part 1 : Data Acquisition

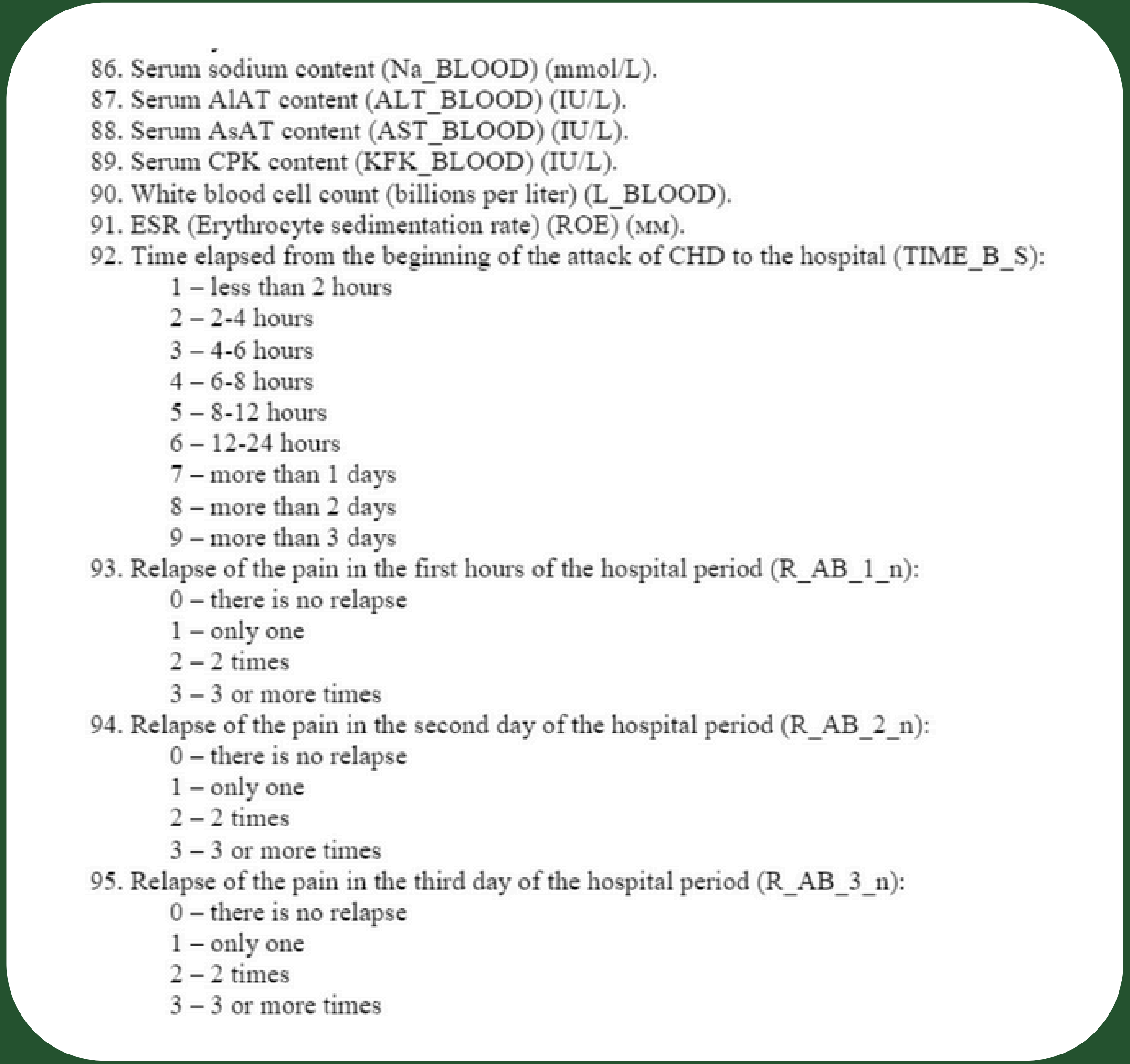
The dataset is taken from Kaggle, from a paper titled *Complications of Myocardial Infarction: A Database for Testing Recognition and Prediction Systems* (link) by authors from the University of Leicester.

The purpose of the dataset is to help predict complications of Myocardial Infarction (MI) based on information about the patient at the time of admission and on the third day of the hospital period.



Myocardial infarction complications Database.csv (437.56 kB)

ID	AGE	SEX	INF_ANAM	STENOK_AN	FK_STE
Patient ID	Age	Gender of the Patient.			
1 – 1700	26 – 92	1700 total values	0 – 3	0 – 6	0
7	70	1	1	1	2
8	65	1	0	1	1
9	60	1	0	0	0
10	77	0	2	0	0
11	71	1	0	0	0
12	50	0	0	0	0
13	60	1	1	0	0
14	57	1	0	0	0



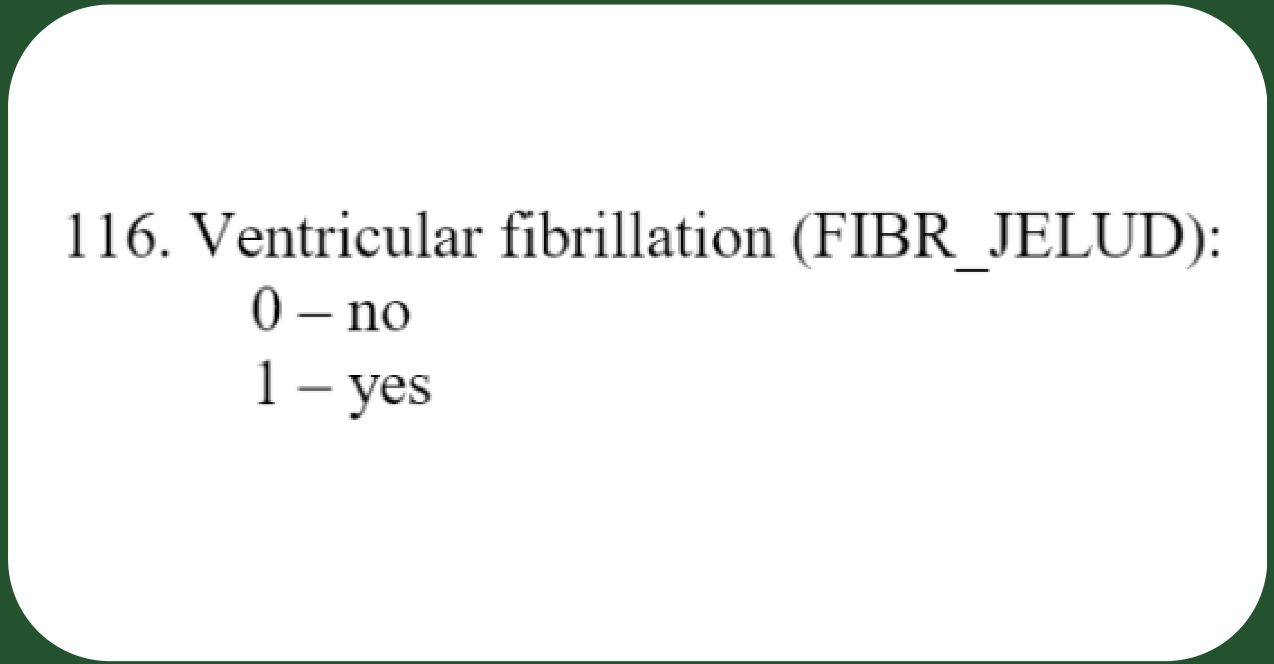
86. Serum sodium content (Na_BLOOD) (mmol/L).
87. Serum AlAT content (ALT_BLOOD) (IU/L).
88. Serum AsAT content (AST_BLOOD) (IU/L).
89. Serum CPK content (KFK_BLOOD) (IU/L).
90. White blood cell count (billions per liter) (L_BLOOD).
91. ESR (Erythrocyte sedimentation rate) (ROE) (мм).
92. Time elapsed from the beginning of the attack of CHD to the hospital (TIME_B_S):
    1 – less than 2 hours
    2 – 2-4 hours
    3 – 4-6 hours
    4 – 6-8 hours
    5 – 8-12 hours
    6 – 12-24 hours
    7 – more than 1 days
    8 – more than 2 days
    9 – more than 3 days
93. Relapse of the pain in the first hours of the hospital period (R_AB_1_n):
    0 – there is no relapse
    1 – only one
    2 – 2 times
    3 – 3 or more times
94. Relapse of the pain in the second day of the hospital period (R_AB_2_n):
    0 – there is no relapse
    1 – only one
    2 – 2 times
    3 – 3 or more times
95. Relapse of the pain in the third day of the hospital period (R_AB_3_n):
    0 – there is no relapse
    1 – only one
    2 – 2 times
    3 – 3 or more times

Part 2 : Data Pre-processing

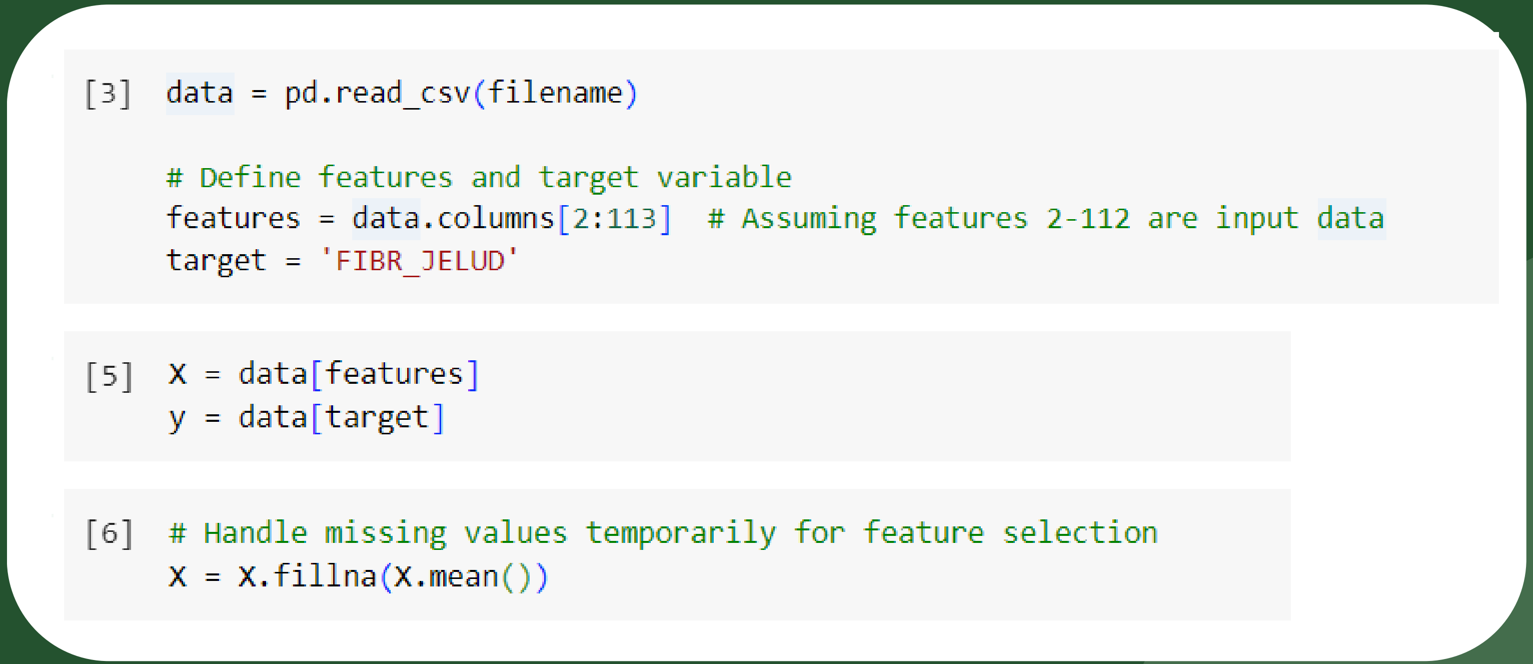
As previously mentioned, the project goal is to have the model help in deciding if Ventricular Fibrillation (VF) might occur as a complication of MI (1) or not (0) based on patient medical data.

In the dataset, authors mentioned that columns 2-112 are input data, while columns 113-124 are possible target features (different complications of MI). The corresponding column for the **target complication (feature) of VF is found to be 'FIBR_JELUD'.**

Before processing, there is also a handling of missing values for the remaining input features.



116. Ventricular fibrillation (FIBR_JELUD):
    0 – no
    1 – yes



```
[3]  data = pd.read_csv(filename)

     # Define features and target variable
     features = data.columns[2:113]  # Assuming features 2-112 are input data
     target = 'FIBR_JELUD'
```

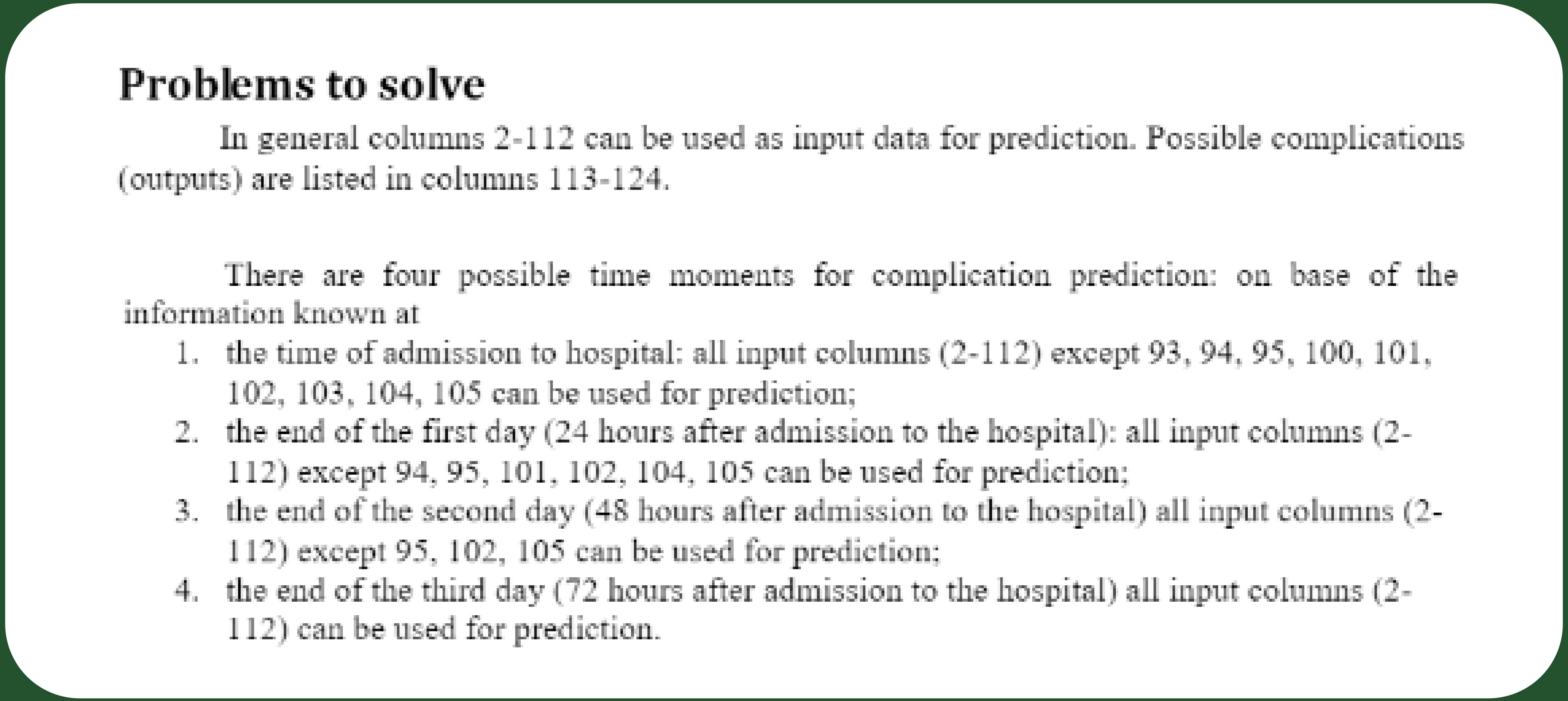
```
[5]  X = data[features]
     y = data[target]
```

```
[6]  # Handle missing values temporarily for feature selection
     X = X.fillna(X.mean())
```

Part 3 : Feature Selection/Extraction

According to creators of the dataset, columns 2-112 are for input features. Meanwhile, the project goal is to be able to determine possibility of VF complication after MI with as little number of signs as possible.

This means that the 111 input features must be reduced accordingly so that only the incredibly relevant features are included in the model. These top most relevant features should be features that can properly distinguish whether VF complication might occur or not.



Problems to solve

In general columns 2-112 can be used as input data for prediction. Possible complications (outputs) are listed in columns 113-124.

There are four possible time moments for complication prediction: on base of the information known at

1. the time of admission to hospital: all input columns (2-112) except 93, 94, 95, 100, 101, 102, 103, 104, 105 can be used for prediction;
2. the end of the first day (24 hours after admission to the hospital): all input columns (2-112) except 94, 95, 101, 102, 104, 105 can be used for prediction;
3. the end of the second day (48 hours after admission to the hospital) all input columns (2-112) except 95, 102, 105 can be used for prediction;
4. the end of the third day (72 hours after admission to the hospital) all input columns (2-112) can be used for prediction.

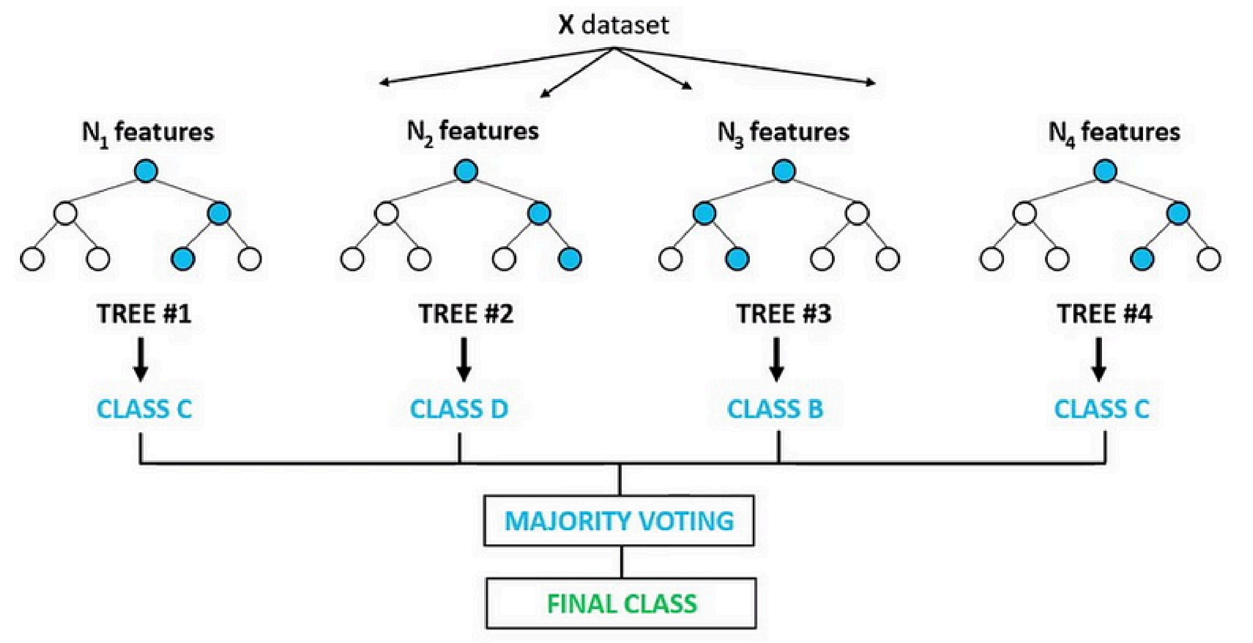
To find feature importance, this project will use Random Forest classifier.

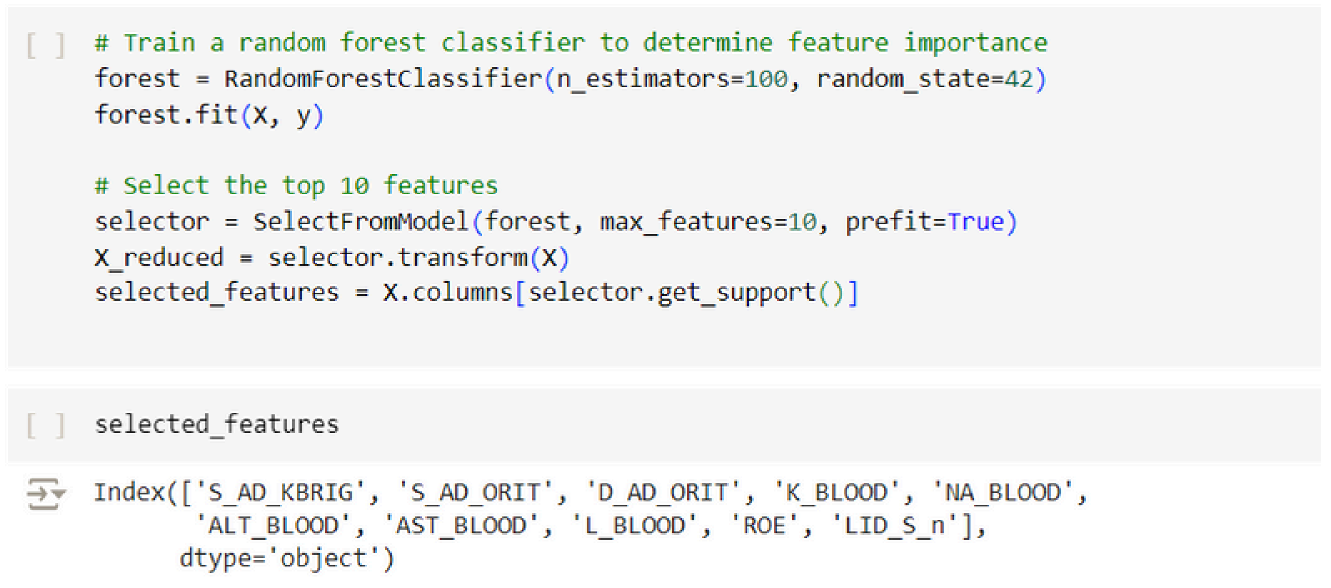
Part 3 : Feature Selection/Extraction

Random Forest (RF) classifier is popular in feature selection since it can rank features based on their importance.

The general process of how RF works :

1. RF builds several decision trees using different data and feature subsets.
2. On construction of each decision tree, RF tracks how much each feature reduces impurity at each split. More impurity reduced means feature is considered more important.
3. Importance scores from all trees are averaged for each feature (aggregation).
4. Then, Features are ranked by their overall importance scores.
5. The 10 most important features are chosen based on these rankings!





```
# Train a random forest classifier to determine feature importance
forest = RandomForestClassifier(n_estimators=100, random_state=42)
forest.fit(X, y)

# Select the top 10 features
selector = SelectFromModel(forest, max_features=10, prefit=True)
X_reduced = selector.transform(X)
selected_features = X.columns[selector.get_support()]
```

```
selected_features
```

```
Index(['S_AD_KBRIG', 'S_AD_ORIT', 'D_AD_ORIT', 'K_BLOOD', 'NA_BLOOD',
       'ALT_BLOOD', 'AST_BLOOD', 'L_BLOOD', 'ROE', 'LID_S_n'],
      dtype='object')
```

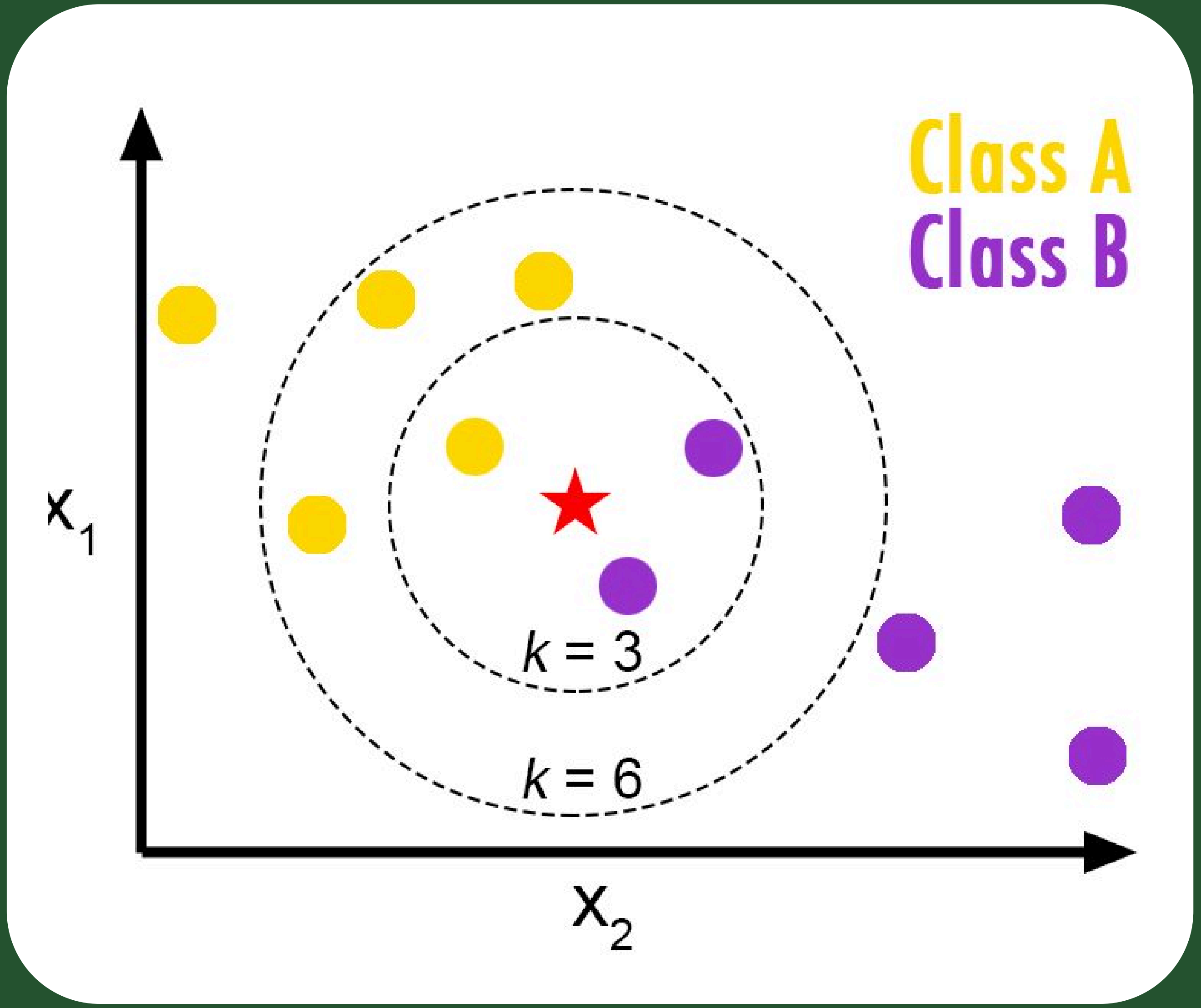
Part 3 : Feature Selection/Extraction

Here are the 10 most relevant (important) features in detecting 'FIBR_JELUD' (Ventricular Fibrillation) according to the Random Forest classifier.

S_AD_KBRIG	Systolic blood pressure according to Emergency Cardiology Team (mmHg)
S_AD_ORIT	Systolic blood pressure according to intensive care unit (mmHg)
D_AD_ORIT	Diastolic blood pressure according to intensive care unit (mmHg)
K_BLOOD	Serum potassium content (mmol/L)
NA_BLOOD	Serum sodium content (mmol/L)
ALT_BLOOD	Serum AlAT content (IU/L)
AST_BLOOD	Serum AsAT content (IU/L)
L_BLOOD	White blood cell count (billions per liter)
ROE	ESR (Erythrocyte sedimentation rate) (ROE) (мм)
LID_S_n	Use of lidocaine in the ICU

Part 4 : Training/Testing Model

? *K-Nearest Neighbors (KNN) works by classifying a data point based on the majority class of its k nearest neighbors in the feature space, determined by a distance metric such as **Euclidean distance**.* ?



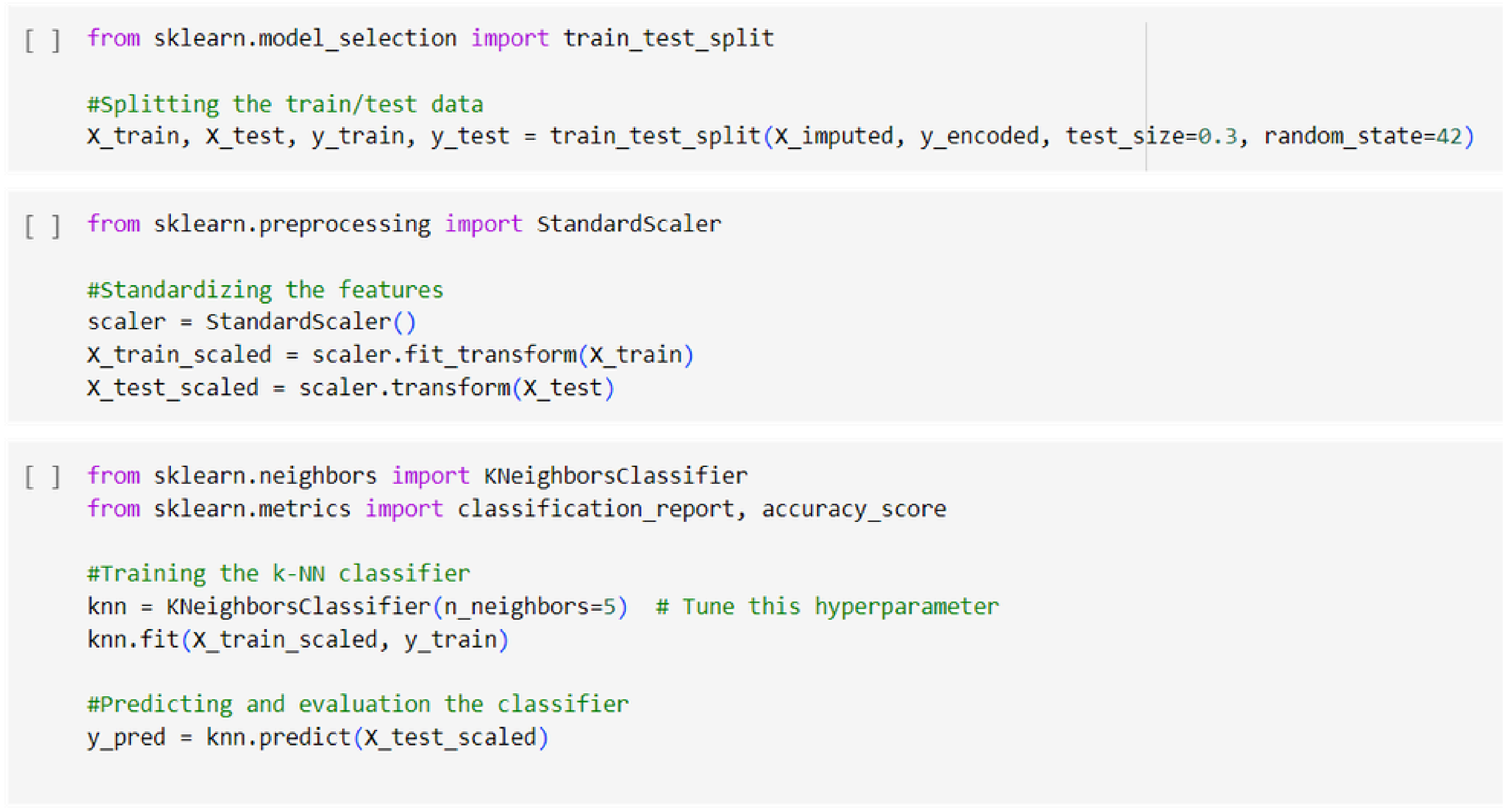
Steps on how KNN Model works :

1. Choose the number of neighbors (k = number of nearest neighbors to consider)
2. Calculate distance between the current point and all other points in the dataset using a chosen distance metric (e.g., Euclidean distance).
3. Select the k data points from the dataset that are closest to the given point in distance
4. Among these k nearest neighbors, count the number of data points belonging to each class.
5. The data point's class is the majority class among the k nearest neighbors.
6. Make predictions by using the assigned class for the given data point as the predicted output.

Part 4 : Training/Testing Model

Once the 10 selected features are retrieved, the next step is training the KNN classifier with thousands of patients data on the previously selected 10 features.

Later, the KNN model will take a new patient's medical data input (of the 10 features) and determine whether the patient will have VF complication or not.



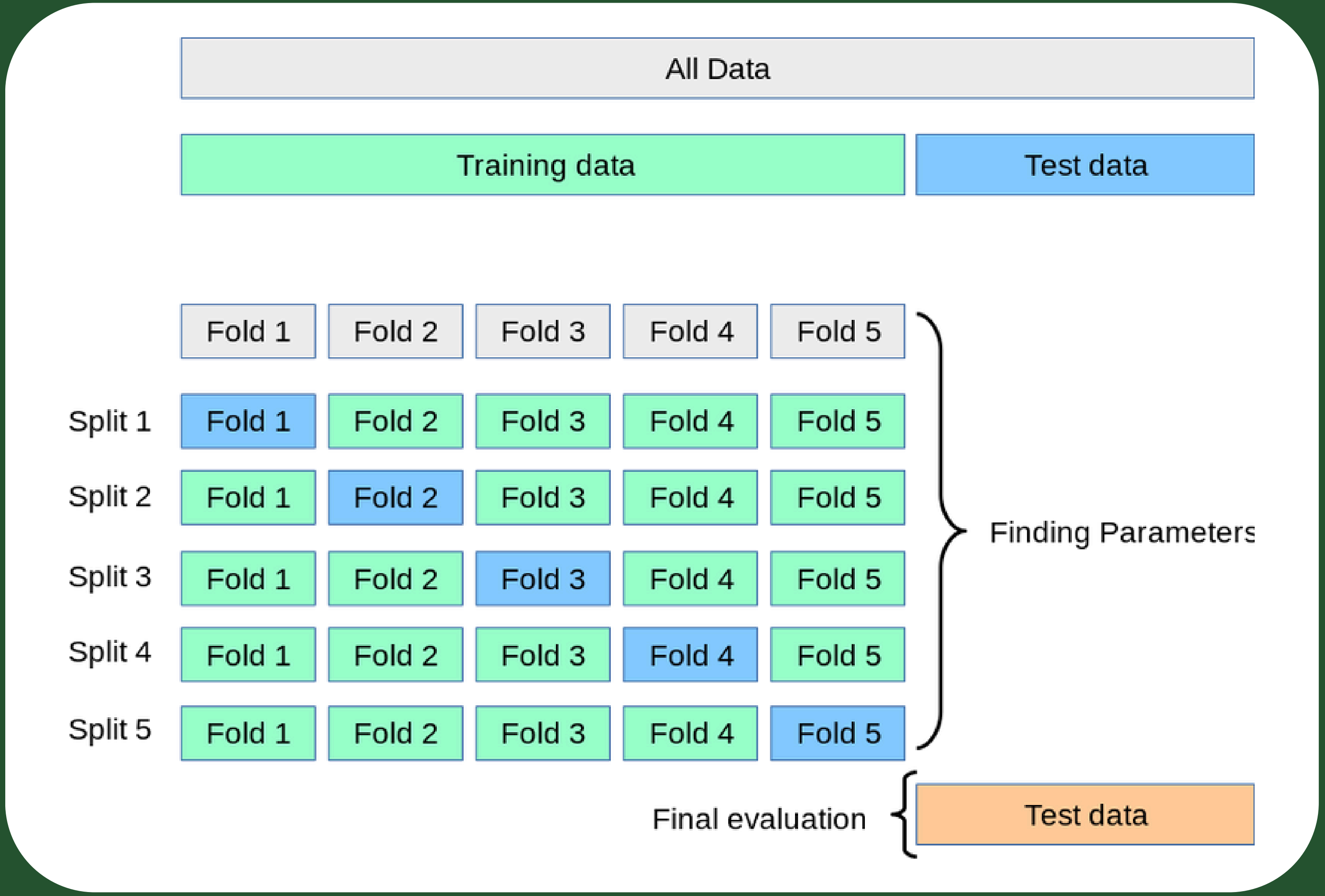
```
[ ] from sklearn.model_selection import train_test_split

    #Splitting the train/test data
    X_train, X_test, y_train, y_test = train_test_split(X_imputed, y_encoded, test_size=0.3, random_state=42)
```

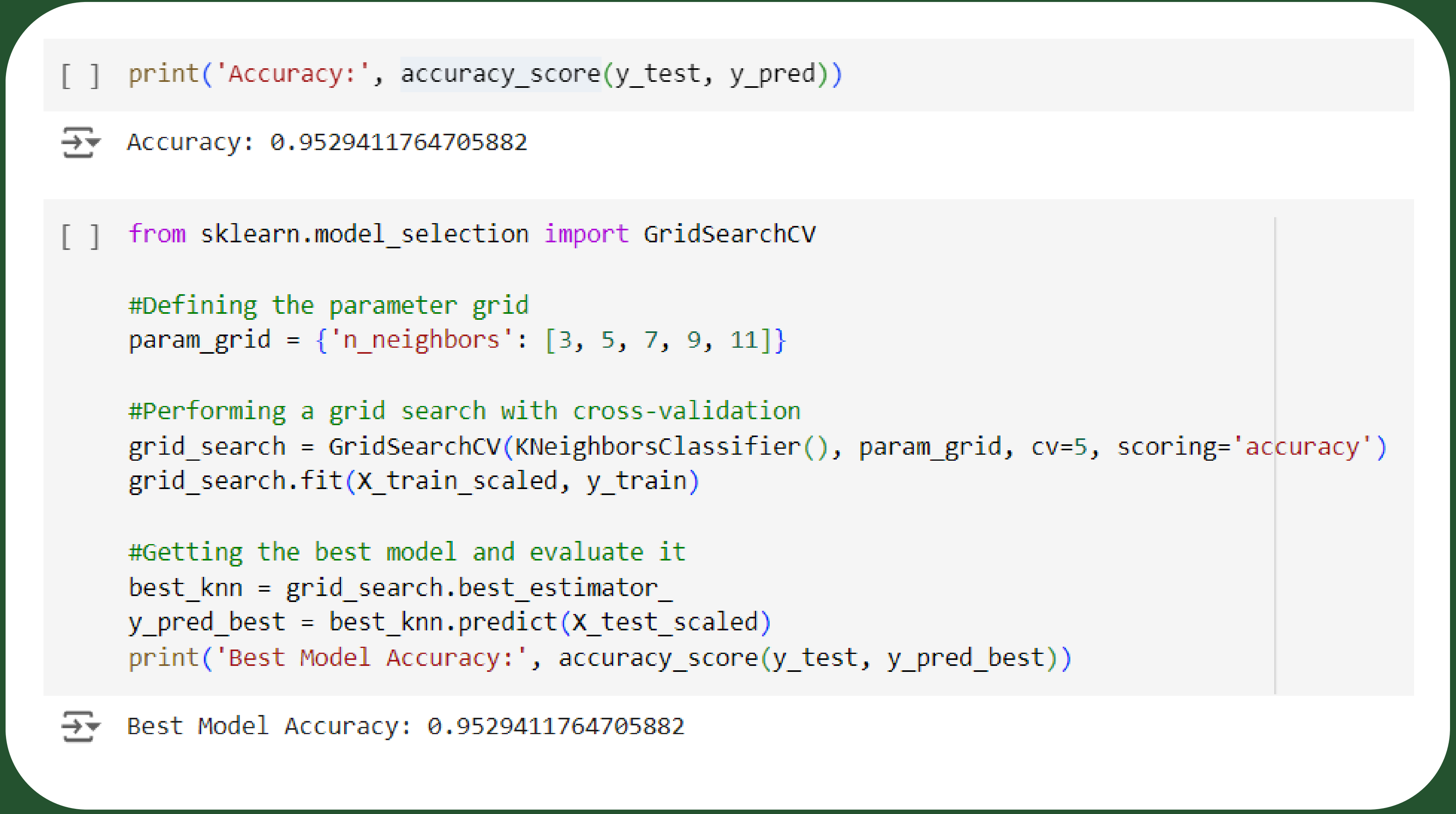
```
[ ] from sklearn.preprocessing import StandardScaler

    #Standardizing the features
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)
```

```
[ ] from sklearn.neighbors import KNeighborsClassifier
    from sklearn.metrics import classification_report, accuracy_score

    #Training the k-NN classifier
    knn = KNeighborsClassifier(n_neighbors=5)  # Tune this hyperparameter
    knn.fit(X_train_scaled, y_train)

    #Predicting and evaluation the classifier
    y_pred = knn.predict(X_test_scaled)
```

Part 5 : Evaluation Results & Discussion

After training/testing the KNN model, the evaluation shows the accuracy is 95%.
There is an additional evaluation by using grid search with cross validation, and the best model is still the model with 95% accuracy. This accuracy is deemed good for the particular project, although it is preferable to have more testings done manually by nurses/doctors who are on the field (and more knowledgable).





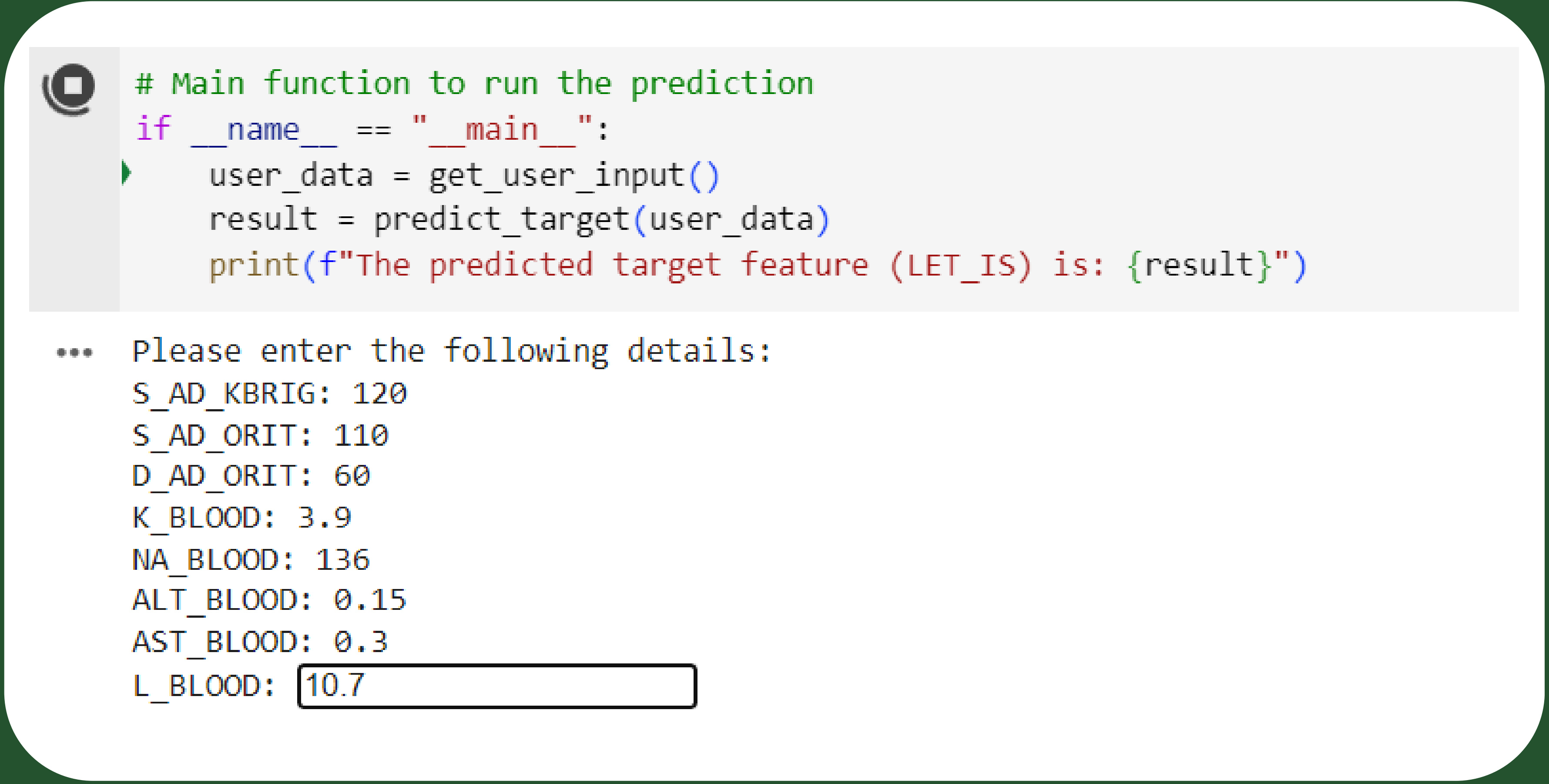
```
[ ] print('Accuracy:', accuracy_score(y_test, y_pred))

    Accuracy: 0.9529411764705882

[ ] from sklearn.model_selection import GridSearchCV

    #Defining the parameter grid
    param_grid = {'n_neighbors': [3, 5, 7, 9, 11]}

    #Performing a grid search with cross-validation
    grid_search = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5, scoring='accuracy')
    grid_search.fit(X_train_scaled, y_train)

    #Getting the best model and evaluate it
    best_knn = grid_search.best_estimator_
    y_pred_best = best_knn.predict(X_test_scaled)
    print('Best Model Accuracy:', accuracy_score(y_test, y_pred_best))

    Best Model Accuracy: 0.9529411764705882
```

User Implementation

As mentioned before, this program is intended to sort out the most important features that can decide for possibility of VF complications.

This is how a nurse/doctor might input the basic medical data to check for chances of VF.

To see the detailed code, please visit the GoogleColab at this link.



```
# Main function to run the prediction
if __name__ == "__main__":
    user_data = get_user_input()
    result = predict_target(user_data)
    print(f"The predicted target feature (LET_IS) is: {result}")
```

```
Please enter the following details:
S_AD_KBRIG: 120
S_AD_ORIT: 110
D_AD_ORIT: 60
K_BLOOD: 3.9
NA_BLOOD: 136
ALT_BLOOD: 0.15
AST_BLOOD: 0.3
L_BLOOD: 10.7
```

Conclusion

This project is inspired by the need for fast diagnosis of VF complication on a hospital patient admitted for MI (heart attack).

The combination of Random Forest for feature selection and KNN for classification model has worked well for this particular case, with quite a high accuracy. It may be improved further, however, by including not only binary classification for VF or not, but also taking into account other complications provided in the dataset.

Thank you very much.

Hope this project is informative enough!