```python
##########################################
# Required Python Packages
##########################################
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier


##########################################
# File Paths
##########################################
INPUT_PATH = "breast-cancer-wisconsin.data"
OUTPUT_PATH = "breast-cancer-wisconsin.csv"


##########################################
# Headers
##########################################
HEADERS = ["CodeNumber", "ClumpThickness", "UniformityCellSize", "UniformityCellShape", "MarginalAdhesion",
        "SingleEpithelialCellSize", "BareNuclei", "BlandChromatin", "NormalNucleoli", "Mitoses", "CancerType"]



##########################################
# Function name : read_data
# Description : Read the data into pandas dataframe
# Inpt :  path of CSV file
# Output : Gives the data
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
##########################################
def read_data(path):
    data = pd.read_csv(path)
    return data


##########################################
# Function name : get_headers
# Description :    dataset headers
# Input : dataset
# Output : Returns the header
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
##########################################
def get_headers(dataset):
    return dataset.columns.values


##########################################
# Function name : add_headers
# Description :    Add the headers to the dataset
# Input : dataset
# Output : Updated dataset
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
##########################################
def add_headers(dataset, headers):
    dataset.columns = headers
    return dataset


##########################################
```

```python
# Function name : data_file_to_csv
# Input : Nothing
# Output : Write the data to CSV
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
####################################
def data_file_to_csv():
    # Headers
    headers = ["CodeNumber", "ClumpThickness", "UniformityCellSize", "UniformityCellShape",
"MarginalAdhesion",
            "SingleEpithelialCellSize", "BareNuclei", "BlandChromatin", "NormalNucleoli", "Mitoses",
            "CancerType"]
    # Load the dataset into Pandas data frame
    dataset = read_data(INPUT_PATH)
    # Add the headers to the loaded dataset
    dataset = add_headers(dataset, headers)
    # Save the loaded dataset into csv format
    dataset.to_csv(OUTPUT_PATH, index=False)
    print ("File saved ...!")


####################################
# Function name : split_dataset
# Description :     Split the dataset with train_percentage
# Input : Dataset with related information
# Output : Dataset after splitting
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
####################################
def split_dataset(dataset, train_percentage, feature_headers, target_header):
    # Split dataset into train and test dataset
    train_x, test_x, train_y, test_y = train_test_split(dataset[feature_headers], dataset[target_header],
                                        train_size=train_percentage)
    return train_x, test_x, train_y, test_y


####################################
# Function name : handel_missing_values
#  Description :    Filter missing values from the dataset
# Input : Dataset with mising values
# Output : Dataset by remocing missing values
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
####################################
def handel_missing_values(dataset, missing_values_header, missing_label):
    return dataset[dataset[missing_values_header] != missing_label]


####################################
# Function name : random_forest_classifier
#  Description :   To train the random forest classifier with features and target data
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
####################################
def random_forest_classifier(features, target):
    clf = RandomForestClassifier()
    clf.fit(features, target)
    return clf


####################################
# Function name : dataset_statistics
```

```python
#  Description :  Basic statistics of the dataset
# Input : Dataset
# Output : Description of dataset
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
###################################
def dataset_statistics(dataset):
    print(dataset.describe())


###################################
# Function name : main
# Description :  Main function from where execution starts
# Author : Piyush Manohar Khairnar
# Date : 01/02/2022
###################################
def main():
    # Load the csv file into pandas dataframe
    dataset = pd.read_csv(OUTPUT_PATH)
    # Get basic statistics of the loaded dataset
    dataset_statistics(dataset)

    # Filter missing values
    dataset = handel_missing_values(dataset, HEADERS[6], '?')
    train_x, test_x, train_y, test_y = split_dataset(dataset, 0.7, HEADERS[1:-1], HEADERS[-1])

    # Train and Test dataset size details
    print("Train_x Shape :: ", train_x.shape)
    print("Train_y Shape :: ", train_y.shape)
    print("Test_x Shape :: ", test_x.shape)
    print("Test_y Shape :: ", test_y.shape)

    # Create random forest classifier instance
    trained_model = random_forest_classifier(train_x, train_y)
    print("Trained model :: ", trained_model)
    predictions = trained_model.predict(test_x)

    for i in range(0, 205):
        print("Actual outcome :: {} and Predicted outcome :: {}".format(list(test_y)[i], predictions[i]))

    print("Train Accuracy :: ", accuracy_score(train_y, trained_model.predict(train_x)))
    print("Test Accuracy  :: ", accuracy_score(test_y, predictions))
    print(" Confusion matrix ", confusion_matrix(test_y, predictions))

###################################
# Application starter
###################################
if __name__ == "__main__":
    main()
```