```python
import numpy
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import pandas as pd

# import dataset and setup dataframe
dataset = "breast-cancer.csv"
names = ['class', 'age', 'menopause', 'tumor-size', 'inv-nodes', 'node-caps', 'deg-malig', 'breast', 'breast-quad', 'irradiat']
df = pd.read_csv(dataset, names=names)

# label encode all of the categorical variables
label_encoders = {}
for column in df.columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# show dataframe
print(df.head())

# set regular x to all features
X = df.drop('class', axis=1)

# drop the less important features based on importance of model
X_limited = df.drop(['class', 'irradiat', 'menopause', 'breast', 'node-caps', 'inv-nodes'], axis=1)

# set y as class target
y = df['class']

# store list of accuracies from trained models
accuracies = []
accuracies_limited = []

# two separate models, one for all features and one for limited features
all_features_classifier = None
limited_features_classifier = None

# number of times to train the models
training_iterations = 500

# iteratively train model 500 times
for i in range(training_iterations):
    # split data set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

    # train model
    all_features_classifier = DecisionTreeClassifier()
    all_features_classifier.fit(X_train, y_train)

    # get predictions
    y_pred = all_features_classifier.predict(X_test)

    # store accuracy in the accuracies list
    accuracy = accuracy_score(y_test, y_pred)
    accuracies.append(accuracy)

# same process using limited features
for i in range(training_iterations):

    # using X_limited instead of X
    X_train, X_test, y_train, y_test = train_test_split(X_limited, y, test_size=0.3, random_state=42)

    limited_features_classifier = DecisionTreeClassifier()
    limited_features_classifier.fit(X_train, y_train)

    y_pred = limited_features_classifier.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    accuracies_limited.append(accuracy)


# generate bins within relevant range
bins = numpy.linspace(0.5, 0.7, 100)

# plot histogram to compare accuracies - interesting note: accuracies using less features tend to centralize around their mean with less deviation
plt.hist([accuracies, accuracies_limited], bins, label=['All features', 'Limited features'])
plt.legend(loc='upper right')
plt.xlabel("Decision Tree accuracy")
plt.ylabel("Instance Count")
plt.show()

# show bar graph of feature importances
plt.figure(figsize=(10, 6))
plt.barh(X.columns, all_features_classifier.feature_importances_, color='skyblue')
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title('Decision Tree Classifier Feature Importances')
plt.show()
```