# Pseudocodice: Models

```
FUNCTION save_confusion_matrix(model, x_test, y_test, class_names, title,
filename)
    saves the confusion matrix for the desired model into a png file

FUNCTION cross_validation_score(model, x_data, y_data, folds)
    calculates the cross_validation_score for the provided model

    RETURN the score's mean and standard deviation

FUNCTION model_accuracy(model, x_test, y_test)
    RETURN the model accuracy score

FUNCTION calculate_and_print_metrics(model, x_test, y_test, title)
    calculates precision and recall scores

    RETURN precision and recall
```

```
FUNCTION train_tree_model(x_train, y_train):
    trains the Decision Tree on the training data
    with sklearn library with max_depth = 5 to avoid overfitting

    RETURN the Decision Tree

FUNCTION train_knn_model(x_train, y_train):
    trains the K-NN on the training data with sklearn library

    RETURN the K-NN classifier

FUNCTION train_rf_model(x_train, y_train)
    trains the Random Forest with 10 trees on the training data
    with sklearn library

    RETURN the Random Forest
```

```
FUNCTION preprocess_data_mushrooms()
    transform the class column with LabelEncoder // valori 0 e 1 per
commestibile e velenoso
    remove veil-type and stalk-root columns due to poor impact
    convert the attributes with one-hot-encoding and scale the information
    RETURN attributes and classes

FUNCTION preprocess_data_rice()
    convert the input data into utf-8 strings
    transform the class colum with LabelEncoder
    scale the features

    RETURN attributes and classes
```

```
FUNCTION visualize_decision_tree(model, feature_names, class_names, filename)
    saves the Decision Tree graph in a .png file

FUNCTION visualize_knn_boundaries(x_train, y_train, k, class_names, filename)
    saves the 2 dimentional K-NN graph in a .png file

FUNCTION plot_feature_importance(model, feature_names, title, filename,
x_data, y_data, top_n)
    IF the model is a Decision Tree or a Random Forest
        gets the model's feature importance
    ELSE IF the model is K-NN
        calcolates the model's permutation_importance and it's mean

    sort the features in descending order based on importance, create the bar
graph and save
    it in a .png file
```

```
FUNCTION main()
    The main enables the user to choose one of the datasets to train and test
the models,
    then calls all the functions and prints all the results and statistics
```