**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ   
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра АСУ

Отчет

о лабораторной работе № 1

по дисциплине «Интеллектуальный анализ данных»

# на тему: «применение библиотеки Scikit-Learn для решения задач классификации и регрессии.»

Выполнил:

студент группы ИСТ-19б

Карпович В. Д.

Проверили:

Шуватова Е.А.

Васяева Т.А.

Донецк – 2022

**Цель работы:** изучение алгоритмов построения деревьев и правил решений; создание и исследование классификационных и моделей с помощью деревьев и правил решений

**Ход работы**

1. Выбрать одну из баз данных (согласовать с преподавателем), список которых представлен на сайте http://archive.ics.uci.edu/ml/. Скачать данные с сайта.
2. Исходные данные должны храниться в файле. Программным способом считать эти данные и сформировать обучающую, проверочную и тестовую выборки. Обучающая выборка должна включать достаточное количество примеров для обучения. Тестовая выборка может быть вполовину меньше обучающей. Примеры из тестовой выборки не должны быть включены в обучающую. Для оценки обучения модели можно использовать проверочную (валидационную) выборку (при достаточном количестве исходных данных) или же использовать кросс-валидацию (если данных мало).
3. Разработать программу на языке Python с использованием библиотек Pandas, Scikit-Learn. Программа должна обеспечивать: построение дерева (правил) алгоритмом согласно варианту на обучающей выборке; для деревьев решений реализовать механизм упрощения деревьев; тестирование дерева (правил); вычисление точности и ошибки классификации; сохранение / считывание модели (деревья или правила); вывод на экран структуры дерева, можно в виде правил. 28
4. Согласно варианту заданий (таблица 1.4) построить классификатор таким образом, чтобы точность на обучающей выборке была не 100%, но более чем 90% (желательно около 97-98%). В процессе построения классификатора, записывайте в таблицу, какие у вас получались точности для различных моделей на обучающей и тестовой выборках в процессе подбора гиперпараметров.

**Метод:** Random Forests

**Датасет:** набор различных комплектующих для ПК и их параметров – имя реализатора, название модели, объем кэш-памяти и т.д. Целевым параметром является выходное КПД детали

**Тип задачи:** регрессия

**Исходный код**

import pandas

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

# # #

vendors\_nums = { "vendor\_name": { "adviser": 1, "amdahl": 2, "apollo": 3, "basf": 4,"bti": 5,"burroughs": 6,"c.r.d": 7,

"cambex": 8,"cdc": 9,"dec": 10,"dg": 11,"formation": 12,"four-phase": 13, "gould": 14, "honeywell": 15, "hp": 16,

"ibm": 17,"ipl": 18,"magnuson": 19,"microdata": 20,"nas": 21,"ncr": 22,"nixdorf": 23,"perkin-elmer": 24,"prime": 25,"siemens": 26,"sperry": 27,

"sratus": 28,"wang": 29, "harris": 30

}}

db = pandas.read\_csv('machine.data').replace(vendors\_nums).drop(columns = ['model\_name'], axis = 1)

print('Raw Data:')

print(db)

print('\n\n')

x = db.loc[:, 'vendor\_name':'prp']

y = db['erp']

(x\_train, x\_test, y\_train, y\_test) = train\_test\_split(x, y, test\_size = 0.4)

forest\_model = RandomForestClassifier().fit(x\_train, y\_train)

y\_test\_predict = forest\_model.predict(x\_test)

print('Predict:')

print(y\_test\_predict)

print("\nTrain set accuracy: {:.3f}".format(forest\_model.score(x\_train, y\_train)))

print("Test set accuracy: {:.3f}".format(forest\_model.score(x\_test, y\_test)))

estimators = 0

plt.rcParams.update({'figure.figsize': (12.0, 8.0)})

plt.rcParams.update({'font.size': 14})

plt.figure(figsize=(20, 20))

\_ = tree.plot\_tree(forest\_model.estimators\_[estimators], feature\_names=x.columns, filled=True)

**Результат работы**



