Auto-WEKA

Программа Auto-WEKA представляет собой приложение для автоматического выбора подходящего алгоритма машинного обучения, наиболее эффективного для обработки заданного набора данных, и осуществляющее автоматическую подстройку гиперпараметров выбранного алгоритма.

Для определения наиболее эффективных значений гиперпараметров Auto-WEKA использует алгоритм SMAC (Sequential Model-Based Algorithm Configuration – последовательная настройка алгоритма по модели) базирующийся на подходе SMBO (Sequential Model-Based Optimization - последовательная оптимизация по модели) основанном на байесовской оптимизации. На каждом шаге работы SMBO: 1. Строится вероятностная модель целевой функции. 2. Подбираются гиперпараметры, которые лучше всего подходят для вероятностной модели. 3. Подобранные гиперпараметры применяются к целевой функции. 4. Вероятностная модель перестраивается.

Установка Auto-WEKA

Алгоритмы, среди которых Auto-WEKA выбирает подходящий, берутся им из приложения WEKA. Фактически Auto-WEKA является надстройкой, дополнительным программным пакетом в приложении WEKA. Поэтому для использования Auto-WEKA вначале нужно установить WEKA, после чего добавить в него Auto-WEKA. Приложение WEKA написано на языке Java и доступно для использования в различных операционных системах.

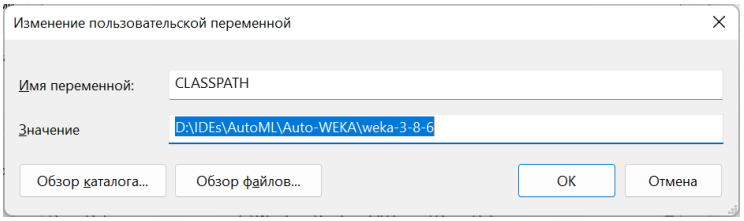
Скачать установщик WEKA можно со страницы проекта на GitHub - https://waikato.github.io/weka-wiki/downloading_weka/

Для операционной системы Windows предлагается два варианта установки:

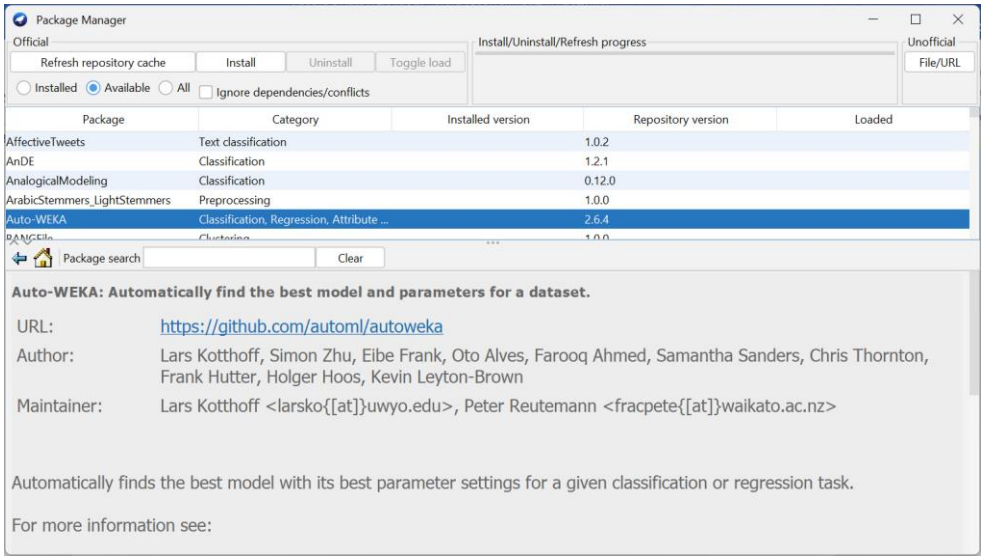
1. Самодостаточный, не требующий наличия установленной заранее виртуальной машины Java, а включающий виртуальную машину как дополнительную часть установщика (weka-3-8-6-azul-zulu-windows.exe)
2. Использующий уже установленную виртуальную машину Java (архив weka-3-8-6.zip)

В последнем случае весь процесс установки состоит из разархивирования zip файла, после чего приложение можно сразу использовать, запустив файл weka.jar.

Однако в случае использования заранее установленной виртуальной машины, для корректной работы Auto-WEKA необходимо установить переменную среды CLASSPATH, так, чтобы она указывала либо на сам файл weka.jar, либо на полный путь к нему.



Добавление дополнительных пакетов, в том числе Auto-WEKA производится с помощью менеджера пакетов, доступного через меню Tools -> Package manager.

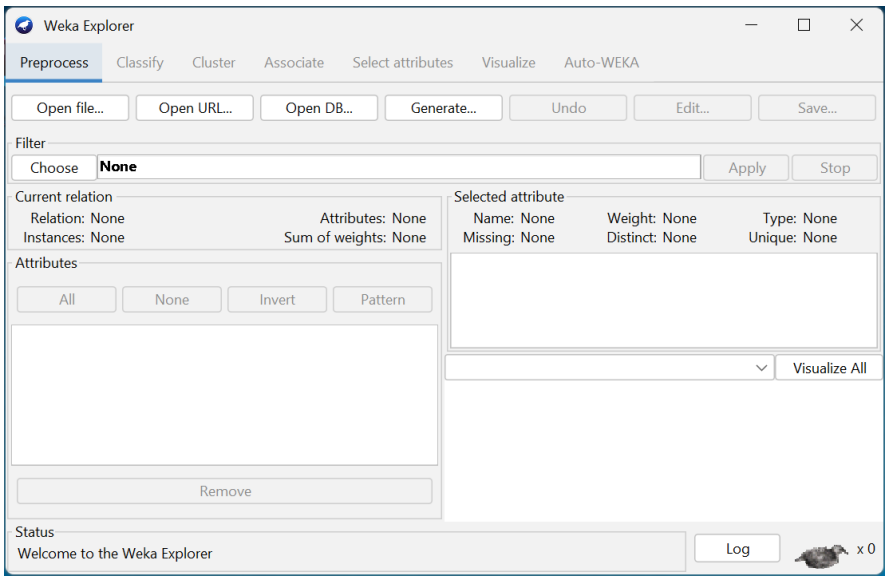


По умолчанию дополнительные файлы, относящиеся к приложению WEKA, находятся по адресу C:\Users\ИмяПользователя\wekafiles, а установленные менеджером пакетов пакеты соответственно в папке C:\Users\ИмяПользователя\wekafiles\packages

Исследование данных

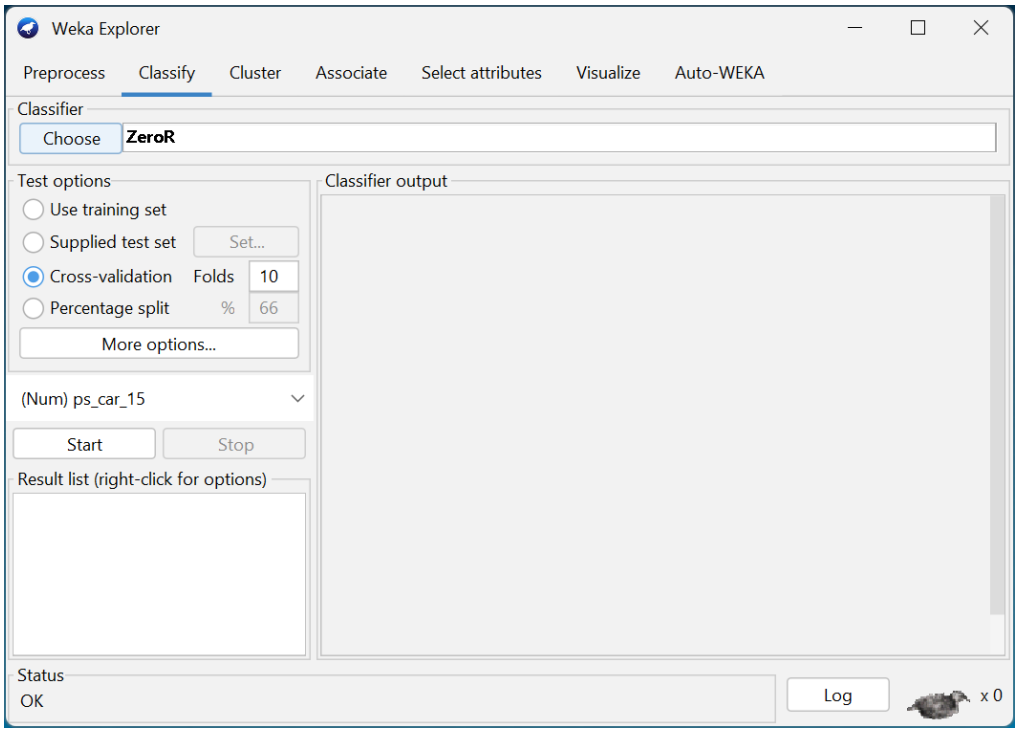
Приложение WEKA состоит из пяти частей, каждая из которых предоставляет свои возможности по работе с данными. Это Explorer, Experimenter, KnowledgeFlow, Workbench и Simple CLI. Компонент Explorer предназначен для исследования исходных данных, и именно там находится вкладка для работы с Auto-WEKA после установки этого пакета.

Прежде всего Explorer необходимо обеспечить данными, с которыми он будет работать. Это делается на единственной активной в начале вкладке - Preprocess. Доступны варианты загрузки данных из локального файла, по url, из базы данных, или самостоятельно сгенерированных.



К загруженным данным можно применять различные фильтры, выбрав соответствующий в дереве фильтров, и указав к каким атрибутам их следует применить. Так в частности можно изменить тип атрибута.

После загрузки данных становятся активными остальные вкладки. На вкладке Classify (классифицировать) можно выбрать классификатор из дерева классификаторов, настроить его гиперпараметры, и запустить для обработки загруженного набора данных.



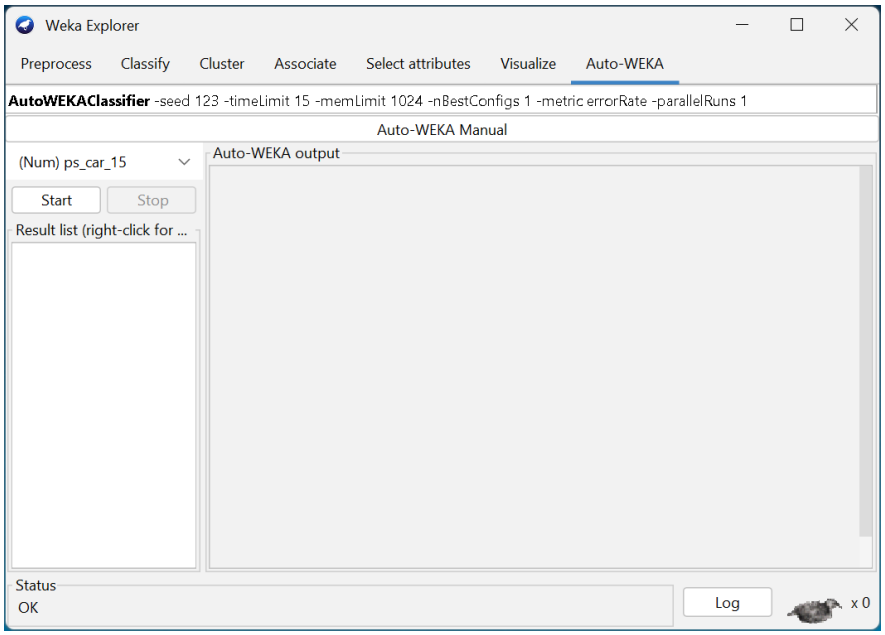
По умолчанию атрибут по которому производится обучение классификатора, и который он затем должен будет определять, является последним атрибутом в наборе данных. Однако это значение можно изменить, выбрав целевой атрибут из выпадающего списка.

Для тестирования обученного классификатора предлагаются четыре варианта: 1)повторное использование обучающей выборки, 2)использование тестовой выборки из отдельного файла,

3)кросс-валидация с указанием числа разбиений, 4)использование только части данных для тестирования, с указанием процентного соотношения используемой для тестирования части.

Установленный компонент Auto-WEKA также находится в дереве классификаторов и может быть выбран в ветке classifiers->meta->AutoWEKAClassifier. Но в этом случае рекомендуется использовать в качестве варианта тестирования повторное использование обучающей выборки, поскольку разбиение данных на обучающую и тестовую части уже заложено в самом механизме Auto-WEKA.

Кроме того, для компонента Auto-WEKA имеется также отдельная вкладка, на которой его тоже можно запустить.

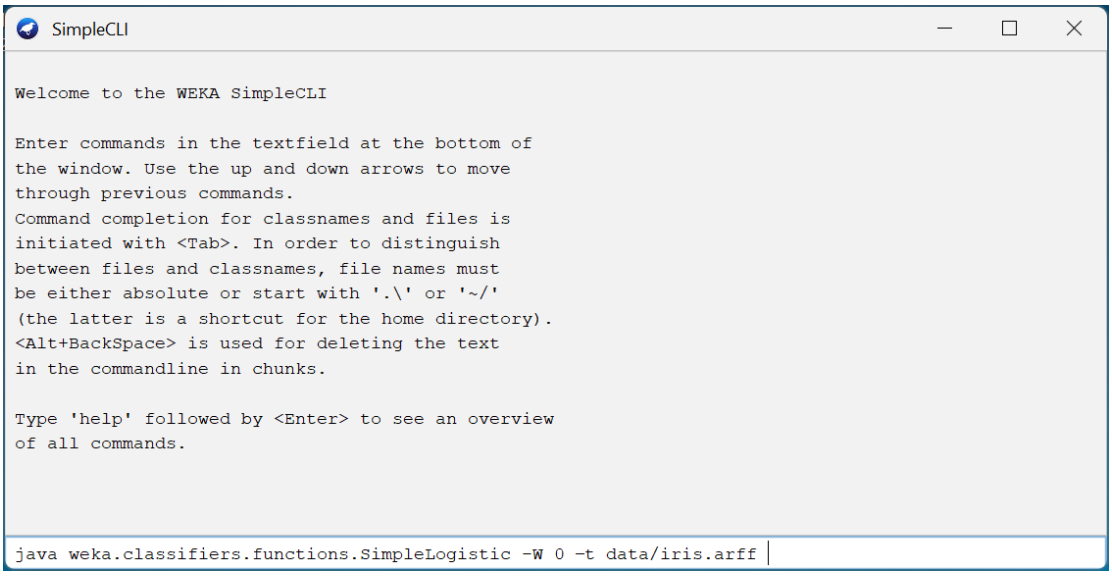


Командная строка SimpleCLI

Кроме обработки данных в компоненте Explorer доступна также работа с ними с помощью командной строки, находящейся в компоненте SimpleCLI.

Формат вызова классификатора из командной строки SimpleCLI:

java имяКлассификатора параметрыКлассификатора параметрыЗапуска



Наиболее употребимые параметры запуска:
-t файл с обучающей выборкой в формате arff
-T файл с тестовой выборкой в формате arff
-x количество делений для кросс-валидации
-d файл для сохранения обученной модели
-l файл для загрузки обученной модели

Пример запуска классификатора SimpleLogistic для обработки данных из файла iris.arff:
java weka.classifiers.functions.SimpleLogistic -W 0 –t data/iris.arff

Задания

Задание 1.

Набор данных porto-seguro (Porto Seguros Safe Driver Prediction) содержит данные о 595212 водителях, каждый из которых описан с помощью 39 атрибутов. Задача состоит в том, чтобы спрогнозировать подаст ли водитель страховой иск в следующем году. Информация о том, подал водитель иск или нет содержится в атрибуте с названием target (0 – не подал, 1 – подал).

1. Загрузить данные в систему WEKA.
2. С помощью компонента Auto-WEKA определить подходящий классификатор и его гиперпараметры.
3. Определить количество испробованных конфигураций.
4. У выбранной конфигурации изучить:
    a. Процент правильных и неправильных предсказаний.
    b. Среднюю абсолютную ошибку (Mean absolute error).
    c. Матрицу ошибок (confusion matrix).
5. Использовать тот же классификатор, который предложил Auto-WEKA, но со значениями гиперпараметров по умолчанию. Сравнить результаты.
6. Использовать другой классификатор. Сравнить результаты.

Задание 2.

Набор данных mnist_784 содержит образцы рукописного написания цифр, размером 28x28 пикселей, соответственно каждый образец имеет 784 атрибута. Задача состоит в том, чтобы распознать по атрибутам к какой цифре относится данный образец.

1. Загрузить данные в систему WEKA.
2. С помощью компонента Auto-WEKA определить подходящий классификатор и его гиперпараметры.
3. Определить количество испробованных конфигураций.
4. Какое значение распознается неправильно чаще всего?
5. Изменить настройки Auto-WEKA чтобы было проверено больше конфигураций. Поменялась ли оптимальная конфигурация?
6. Сохранить обученную модель в файл.
7. Протестировать сохраненную модель на тестовой выборке.
8. Использовать в качестве тестовой выборки данные из csv файла, выполнив необходимые преобразования (количество, порядок, тип и названия атрибутов должны совпадать и в обучающей и в тестовой выборке).