Darwin™ Run Time Engine User Guide

A SparkCognition™ Education Document

v. 3.0.0 - January 2020

Darwin Run-Time Engine User Guide

Contents

Overview 1

Darwin documentation 2

Prerequisites 2

Creating an initial model and data profiler 2

Darwin RTE installation and use 2

Windows 3

Linux and MacOS 4

Contact Support 5

Revision Table 5

Overview

Darwin™ is a SparkCognition™ tool that automates model building processes to solve specific problems. This tool enhances data scientist potential because it automates various tasks that are often manually performed. These tasks include data cleaning, latent relationship extraction, and optimal model determination. Darwin promotes rapid and accurate feature generation through both automated windowing and risk generation. Darwin quickly creates highly-accurate, dynamic models using both supervised and unsupervised learning methods.

The Darwin Runtime Engine (RTE) is an interface that enables users to run Darwin models outside of the Darwin cloud or in an on-premises environment. The Darwin RTE handles all data preprocessing and scoring in order to create predictions on a run-time dataset. Users are able to use the run-time engine library, a dataset, and a data profiler file and can run a model or analyze a model without connecting to the internet. Darwin RTE is currently designed for running supervised and NBM learning algorithms. It does not support unsupervised models.

The Darwin RTE is now delivered as an executable. Contact SparkCognition support for download instructions for the appropriate executable (Windows or Linux/MacOS).

Darwin documentation

The documentation for this version of the Darwin RTE includes:

- The *Darwin Release Notes*, version 2.0.5
- The *Darwin User Interface Guide*, version 2.0.5
- The *Darwin API User Guide*, version 1.36.1
- The *Darwin Python SDK User Guide*, version 1.46.0
- The *Darwin RTE User Guide*, version 3.0.0

All of these documents are available for download from the Darwin support portal.

Prerequisites

This section describes the hardware and dataset requirements to run the Darwin RTE:

- **Hardware**
  - At least 2GB of internal RAM
  - At least 5GB of storage
  - Includes AVX, AVX2 instruction sets
  - Operating systems supported:
    - Windows 10 Professional
    - CentOS 7.5, 7.7 and later
    - Red Hat Enterprise Linux (RHEL) version 7
- Dataset must be less than 1GB and its path should not contain any spaces.
- Only Darwin 2.x models are supported in this version of the RTE.

Creating an initial model and data profiler

Before you can use the Darwin RTE, you need to create an initial model and data profiler. You can create a model in the Darwin UI or through the Darwin SDK. When you invoke the *create_model* command in the SDK, Darwin trains the initial model and also creates a data profiler. After creating the model and data profiler, you can download them using the *download_model* command. For more information on using the SDK to create and download models, see the *Darwin Python SDK Guide*.

Note: If you don't want to go through the process of creating an initial model and data profiler using your dataset, there is a sample dataset, model, and data profiler available for inspection and use.

Darwin RTE installation and use

The Darwin RTE is distributed as a compressed file (.zip file for Windows, .tar file for Linux and MacOS) that contains an executable as well as all the necessary files needed for running the RTE. See the appropriate instructions for your operating system.

Windows

After downloading the .zip file to your computer, make sure that it is in a location where you have permission to run the executable. You can uncompress the file by right-clicking on it and selecting **Extract All**.

The RTE executable is run on the command line. A script will set the necessary environmental variables and invoke the executable with the defined arguments. Ensure that the current directory for your command line is inside the folder that was uncompressed. This is the assumption with the following commands.

Open the command prompt or Powershell and run the following command. There are two functions you can perform. They are detailed below.

To run a prediction, run the following command:

```
$> run.bat predict  <path-to-model-and-data_profile-folder>  /
 <path-to-dataset>  --char_encoding utf-8 --out_path /
 <path-to-outfile.csv>
```

To analyze a model, run the following command:

```
$> run.bat analyze_model  <path-to-model-and-data_profile-folder>  /
 <path-to-dataset>  --char_encoding utf-8 --out_path /
 <path-to-outfile.json>
```

The arguments of these commands include:

- **Location of model and data_profile** - Use the absolute path to the folder that contains the `model` and `data_profile` files. Do not include any trailing slashes.
- **Location of dataset** - Use the absolute path to the folder that contains the dataset that will be fed into a prediction or that will be used to analyze the model. Do not include any trailing slashes.
- **Character encoding of dataset** *(optional)* - The character encoding of the dataset. The default value is `utf-8` and will be used if this parameter is not set. If your dataset has a different encoding, set the value with this parameter. For a list of possible values, click here.
- **Location of output file** *(optional)*

  Optionally, you can specify the name and location containing the output file by entering the following argument as part of the run command: `--out_path <path-to-outfile>`

  If the output file path is not given in the command line argument, the output will be sent to the standard out, that is, the console.

  Note: The output file is a .csv file for `predict` and a .json file for `analyze_model`.

A more real-world example of these commands would be similar to the following:

```
$> run.bat predict C:\Users\my_home\model_folder /
C:\Users\my_home\dataset_folder\dataset.csv /
--char_encoding latin-1 /
--out_path C:\Users\my_home\dataset_folder\predictions.csv
```

```
$> run.bat analyze_model C:\Users\my_home\model_folder /
C:\Users\my_home\dataset_folder\dataset.csv /
--char_encoding cp1252 /
--out_path C:\Users\my_home\model_folder\analyze.json
```

Linux and MacOS

After downloading the .tar file, make sure that it is in a location where you have permission to run the executable. To uncompress the .tar file, execute the following command:

```
$ tar -xvf  <exe_name> .tar.gz
```

where `<exe_name>` is the name of the compressed file.

The RTE executable is run on the command line. A script will set the necessary environmental variables and invoke the executable with the defined arguements. Ensure that the current directory for your command line is inside the folder that was uncompressed. This is the assumption with the following commands.

Open the terminal run the following command. The arguments are discussed below.

To run a prediction, run the following command:

```
$ run.sh predict  <path-to-model-and-data_profile-folder>  \
 <path-to-dataset>  --char_encoding utf-8 \
--out_path  <path-to-outfile.csv>
```

To analyze a model, run the following command:

```
$ run.sh analyze_model  <path-to-model-and-data_profile-folder>  \
 <path-to-dataset>  --char_encoding utf-8 \
--out_path  <path-to-outfile.json>
```

The arguments of these commands include:

- **Location of model and data_profile** - Use the absolute path to the folder that contains the `model` and `data_profile` files. Do not include any trailing slashes.
- **Location of dataset** - Use the absolute path to the folder that contains the dataset that will be fed into a prediction or that will be used to analyze the model. Do not include any trailing slashes.
- **Character encoding of dataset** *(optional)* - The character encoding of the dataset. The default value is `utf-8` and will be used if this parameter is not set. If your dataset has a different encoding, set the value with this parameter. For a list of possible values, click here.
- **Location of output file** *(optional)*

  Optionally, you can specify the name and location containing the output file by entering the following argument as part of the run command: `--out_path <path-to-outfile>`

  If the output file path is not given in the command line argument, the output will be sent to the standard out, that is, the console.

Note: The output file is a .csv file for `predict` and a .json file for `analyze_model`.

A more real-world example of these commands would be similar to the following:

```
$ run.sh predict /home/my_home/model_folder \
/home/my_home/dataset_folder/dataset.csv \
--char_encoding utf-16 \
--out_path /home/my_home/dataset_folder/predictions.csv
```

```
$ run.sh analyze_model /home/my_home/model_folder \
/home/my_home/dataset_folder/dataset.csv \
--char_encoding utf-16 \
--out_path /home/my_home/model_folder/analyze.json
```

Contact Support

The following methods enable you to research issues, create a support ticket, or contact SparkCognition:

- **Use the Darwin support portal** - Read Frequently Asked Questions (FAQ), download documentation, or log your issue.
- **Email Support** - Send email to darwin_support@sparkcognition.com.
- **Phone Support** - The SparkCognition support line is +1-512-400-2001.

Revision Table

Version	Date	Notes
v 1.5	15-Oct-2018	Initial Release
v 1.6.1	25-Feb-2019	Clarification of Python environment
v 1.6.2	22-Mar-2019	Version number change only
v 1.7	17-May-2019	Note on ipython use and python 3.7 use
v 2.0.3	14-Nov-2019	Complete change to delivery mechanism by executable file and new command-line functionality
v 2.0.4	19-Dec-2019	Added compatibility requirement with version 2.x models
v 3.0.0	30-Jan-2020	Added optional parameter to set character encoding: --char_encoding utf-8