

# PySpark interpreter for PyCharm

# Spark install steps

- Install Python
  - miniconda recommended
    - <https://docs.conda.io/en/latest/miniconda.html>
- Install Spark
  - <https://phoenixnap.com/kb/install-spark-on-windows-10>
  - <https://medium.com/beeranddiapers/installing-apache-spark-on-mac-os-ce416007d79f>
  - <https://phoenixnap.com/kb/install-spark-on-ubuntu>
- Add PySpark package to Python
  - Simplified with conda environment
- PySpark interpreter for PyCharm

# Install Python

- miniconda recommended
  - <https://docs.conda.io/en/latest/miniconda.html>
- Anaconda works too
  - Larger installed base
- On Windows
  - Install in either
    - Windows
    - WSL 2

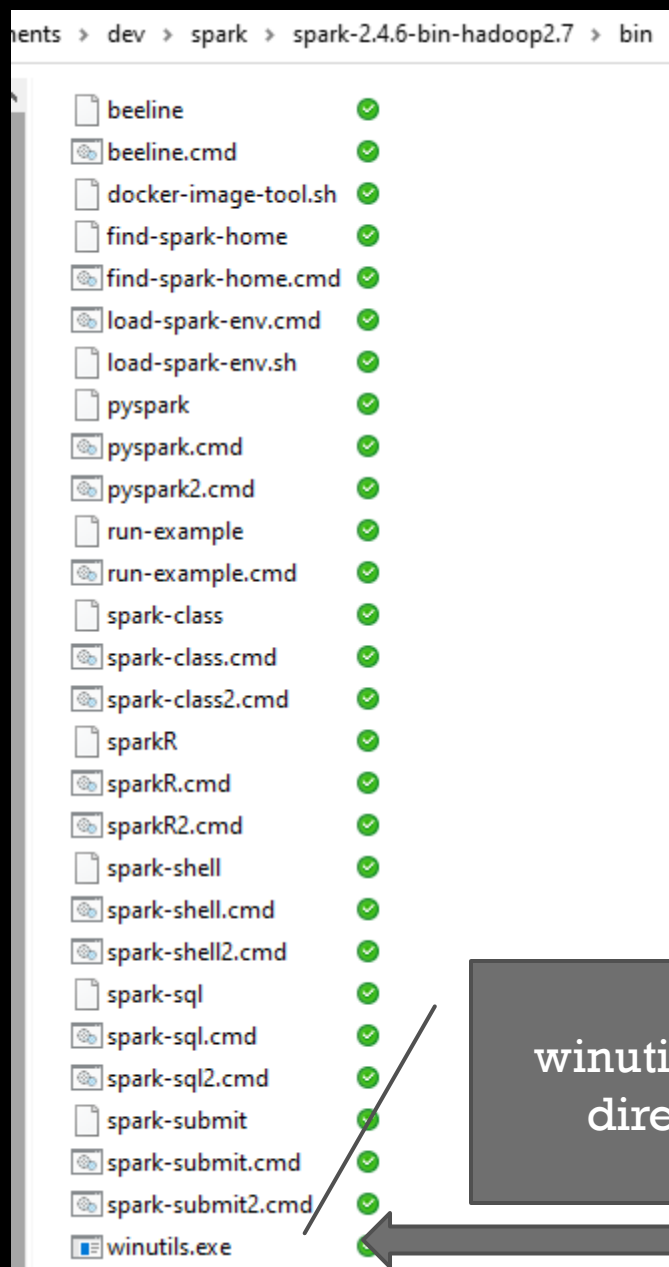
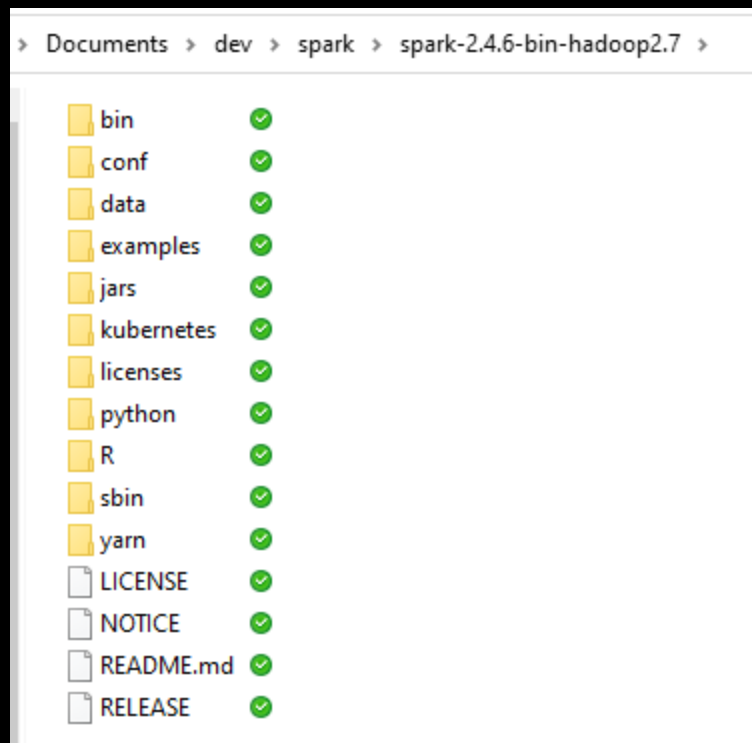
# Example windows install

Data (D:) > Users > robin > OneDrive > miniconda3			Search miniconda3		
condabin	✓	api-ms-win-core-sysinfo-l1-1-0.dll	✓		
conda-meta	✓	api-ms-win-core-timezone-l1-1-0.dll	✓		
DLLs	✓	api-ms-win-core-util-l1-1-0.dll	✓		
envs	✓	api-ms-win-crt-conio-l1-1-0.dll	✓		
etc	✓	api-ms-win-crt-convert-l1-1-0.dll	✓		
include	✓	api-ms-win-crt-environment-l1-1-0.dll	✓		
info	✓	api-ms-win-crt-file-system-l1-1-0.dll	✓		
Lib	✓	api-ms-win-crt-heap-l1-1-0.dll	✓		
Library	✓	api-ms-win-crt-locale-l1-1-0.dll	✓		
libs	✓	api-ms-win-crt-math-l1-1-0.dll	✓		
Menu	✓	api-ms-win-crt-multibyte-l1-1-0.dll	✓		
pkgs	✓	api-ms-win-crt-private-l1-1-0.dll	✓		
Scripts	✓	api-ms-win-crt-process-l1-1-0.dll	✓		
shell	✓	api-ms-win-crt-runtime-l1-1-0.dll	✓		
tcl	✓	api-ms-win-crt-stdio-l1-1-0.dll	✓		
Tools	✓	api-ms-win-crt-string-l1-1-0.dll	✓		
.nonadmin	✓	api-ms-win-crt-time-l1-1-0.dll	✓		
_conda.exe	✓	api-ms-win-crt-utility-l1-1-0.dll	✓		
api-ms-win-core-console-l1-1-0.dll	✓	concr140.dll	✓		
api-ms-win-core-datetime-l1-1-0.dll	✓	cwp.py	✓		
api-ms-win-core-debug-l1-1-0.dll	✓	LICENSE_PYTHON.txt	✓		
api-ms-win-core-errorhandling-l1-1-0.dll	✓	msvc140.dll	✓		
api-ms-win-core-file-l1-1-0.dll	✓	msvc140_1.dll	✓		
api-ms-win-core-file-l1-2-0.dll	✓	msvc140_2.dll	✓		
api-ms-win-core-file-l2-1-0.dll	✓	python.exe	✓		
api-ms-win-core-handle-l1-1-0.dll	✓	python.pdb	✓		
api-ms-win-core-heap-l1-1-0.dll	✓	python3.dll	✓		
api-ms-win-core-interlocked-l1-1-0.dll	✓	python38.dll	✓		
api-ms-win-core-libraryloader-l1-1-0.dll	✓	python38.pdb	✓		
api-ms-win-core-localization-l1-2-0.dll	✓	pythonw.exe	✓		
api-ms-win-core-memory-l1-1-0.dll	✓	pythonw.pdb	✓		
api-ms-win-core-namedpipe-l1-1-0.dll	✓	ucrtbase.dll	✓		
api-ms-win-core-processenvironment-l1-1-0.dll	✓	Uninstall-Miniconda3.exe	✓		
api-ms-win-core-processthreads-l1-1-0.dll	✓	vccorlib140.dll	✓		
api-ms-win-core-processthreads-l1-1-1.dll	✓	vcomp140.dll	✓		
api-ms-win-core-profile-l1-1-0.dll	✓	vcruntime140.dll	✓		
api-ms-win-core-rtlsupport-l1-1-0.dll	✓	venvlauncher.exe	✓		
api-ms-win-core-string-l1-1-0.dll	✓	venvwlauncher.exe	✓		
api-ms-win-core-synch-l1-1-0.dll	✓				
api-ms-win-core-synch-l1-2-0.dll	✓				

# Install Spark



# Example spark install



winutils in bin  
directory

# Ensure environment variables are set

Environment Variables



User variables for robin

Variable	Value
HADOOP_HOME	D:\Users\robin\OneDrive\Profile\Documents\dev\hadoop
JAVA_HOME	C:\Program Files\Java\jre1.8.0_281
OneDrive	D:\Users\robin\OneDrive - Georgia State University
OneDriveCommercial	D:\Users\robin\OneDrive - Georgia State University
OneDriveConsumer	D:\Users\robin\OneDrive
Path	C:\Users\robin\AppData\Local\Microsoft\WindowsApps;C:\Users\robin\AppData\Local\Googl...
PSModulePath	C:\Users\robin\Documents\WindowsPowerShell\Modules;C:\Users\robin\AppData\Local\Goog...
SPARK_HOME	D:\Users\robin\OneDrive\Profile\Documents\dev\spark\spark-2.4.6-bin-hadoop2.7
TEMP	C:\Users\robin\AppData\Local\Temp

New...

Edit...

Delete

System variables

# Add PySpark package to Python

- Simplified with conda environment
  - Create an environment (interpreter) for the spark packages
    - `conda create --name pyspark37 python==3.7`

```
D:\Users\robin\OneDrive\miniconda3>conda create --name pyspark37 python==3.7
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: D:\Users\robin\OneDrive\miniconda3\envs\pyspark37
```

```
added / updated specs:
- python==3.7
```

```
The following packages will be downloaded:
```

package	build	
python-3.7.0	hea74fb7_0	16.6 MB
Total:		16.6 MB



# Add PySpark package to Python

- Add spark package to Python interpreter
  - For conda, first activate the environment

```
D:\Users\robin\OneDrive\miniconda3>conda activate pyspark37  
(pyspark37) D:\Users\robin\OneDrive\miniconda3>
```

- Add PySpark package
  - First try conda install
  - If fails, the pip install

```
(pyspark37) D:\Users\robin\OneDrive\miniconda3>conda install pyspark  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
environment location: D:\Users\robin\OneDrive\miniconda3\envs\pyspark37  
  
added / updated specs:  
- pyspark
```

# Add additional packages to Python

- Common packages

- pandas, wget, setuptools, ...

```
(pyspark37) D:\Users\robin\OneDrive\miniconda3>conda install pandas setuptools
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

- Spark helpful packages

- findspark

```
(pyspark37) D:\Users\robin\OneDrive\miniconda3>pip install wget
Collecting wget
  Using cached wget-3.2-py3-none-any.whl
Installing collected packages: wget
Successfully installed wget-3.2
```

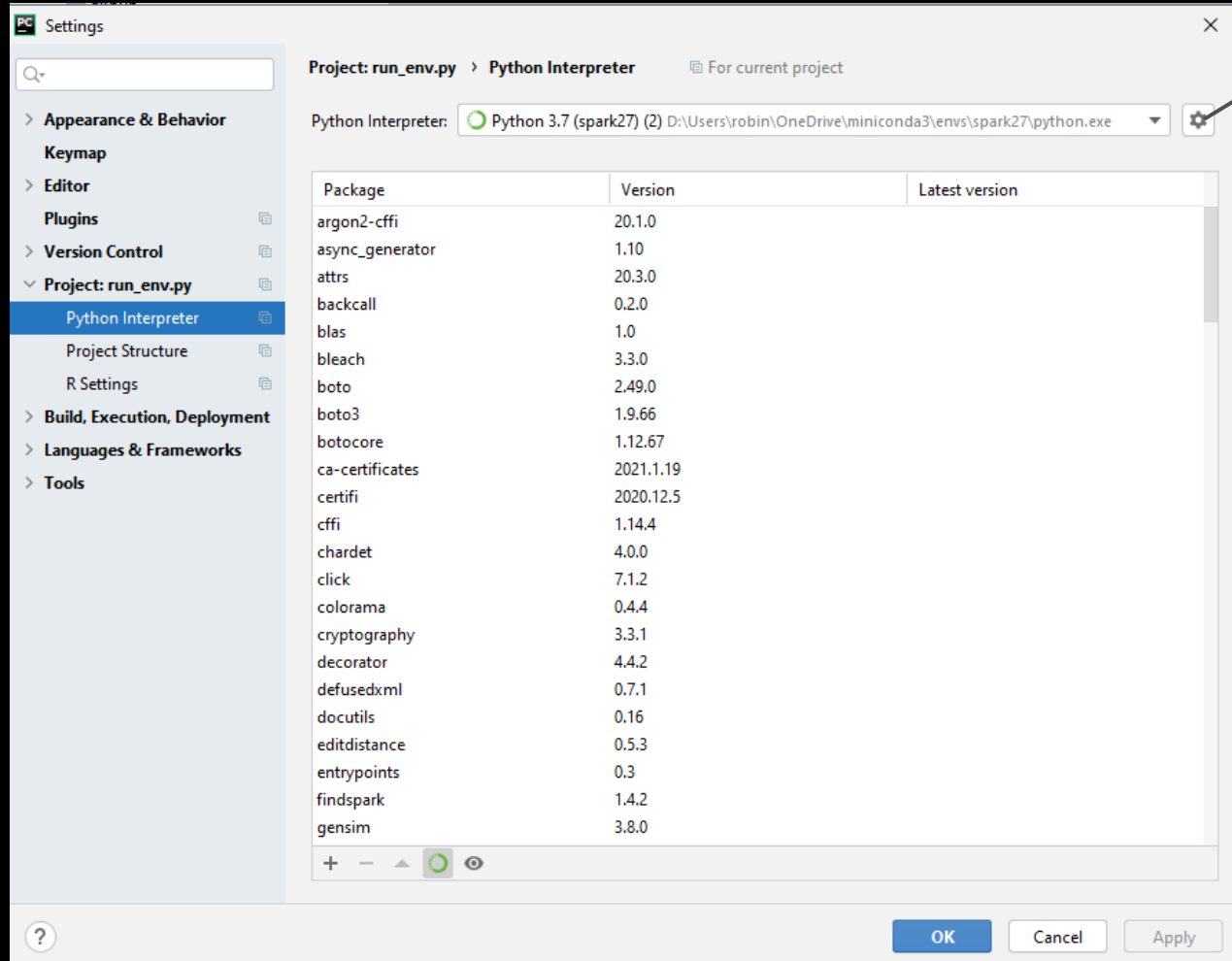
- pip

- May have to use pip to install wget, findspark

```
(pyspark37) D:\Users\robin\OneDrive\miniconda3>pip install findspark
Collecting findspark
  Using cached findspark-1.4.2-py2.py3-none-any.whl (4.2 kB)
Installing collected packages: findspark
Successfully installed findspark-1.4.2
```

# PySpark interpreter for PyCharm

## File | Settings | Project | Python Interpreter



Gear  
Add Interpreter

# Select interpreter with installed PySpark package

PC Add Python Interpreter

☐ New environment

Location:

Python version:

Conda executable:

☐ Make available to all projects

☒ Existing environment

Interpreter:

Conda executable:


☒ Make available to all projects

OK Cancel

Virtualenv Environment  
Conda Environment  
System Interpreter  
Pipenv Environment  
SSH Interpreter  
Vagrant  
WSL  
Docker  
Docker Compose

# Python Console in PyCharm

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("demo").getOrCreate()
```



The screenshot shows the PyCharm Python Console window. The title bar is 'Python Console x'. The console output includes the command prompt path, the execution of the code from the previous block, and various status messages from PyDev and Spark. The code being executed is:

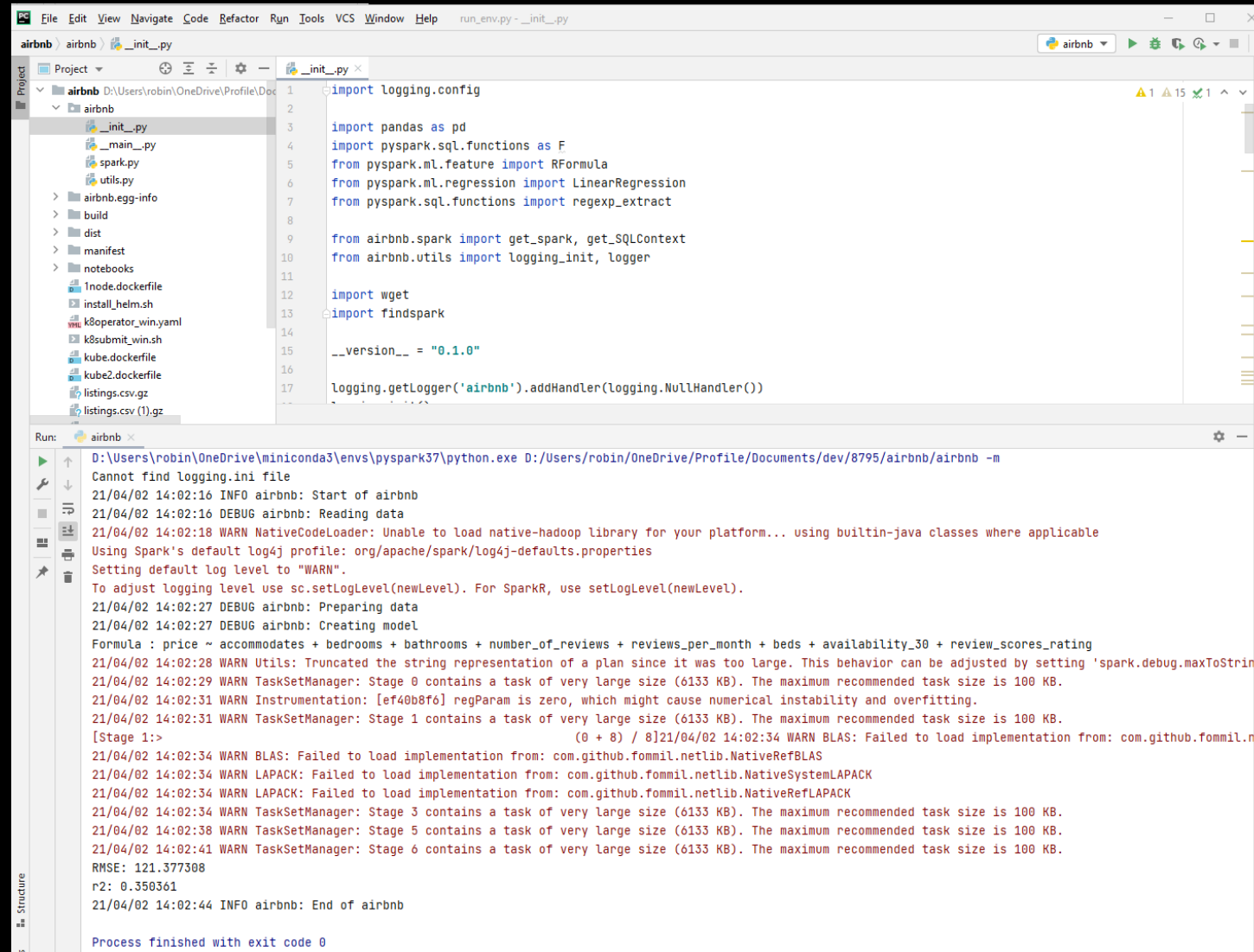
```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\Users\\robin\\OneDrive\\Profile\\Documents\\dev\\8795\\airbnb', 'D:/Users/robin/OneDrive/Profile/Documents/dev/8795/airbnb'])
```

The output shows the PyDev console starting, the Python version and platform (Python 3.7.0 on win32), and the successful execution of the SparkSession code. It also displays warning messages from Spark regarding the native-hadoop library and log settings.

```
D:\Users\robin\OneDrive\miniconda3\envs\pyspark37\python.exe "C:\Program Files\JetBrains\PyCharm 2020.3.2\plugins\python\helpers\pydev\pydevconsole.py"
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\Users\\robin\\OneDrive\\Profile\\Documents\\dev\\8795\\airbnb', 'D:/Users/robin/OneDrive/Profile/Documents/dev/8795/airbnb'])
PyDev console: starting.
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] on win32
>>> from pyspark.sql import SparkSession
... spark = SparkSession.builder.appName("demo").getOrCreate()
...
21/04/02 14:00:22 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
>>>
```

# Now, run projects in PySpark interpreter

## Play button upper right



The screenshot shows an IDE window with the following components:

- Project Explorer (Left):** Displays the project structure for 'airbnb'. The file `_init_.py` is selected.
- Code Editor (Center):** Shows the contents of `_init_.py`, which includes imports for `logging.config`, `pandas`, `pyspark.sql.functions`, `RFormula`, `LinearRegression`, `regexp_extract`, `get_spark`, `get_SQLContext`, `wget`, and `findspark`. It also defines `__version__ = "0.1.0"` and sets up logging.
- Run Console (Bottom):** Shows the output of the PySpark interpreter. The command executed is `D:\Users\robin\OneDrive\Profile\Documents\dev\8795\airbnb\airbnb -m`. The output includes logs for starting the interpreter, reading data, creating a model, and displaying the RMSE (121.377308) and R2 (0.350361) values. The process finished with exit code 0.

```
import logging.config
import pandas as pd
import pyspark.sql.functions as F
from pyspark.ml.feature import RFormula
from pyspark.ml.regression import LinearRegression
from pyspark.sql.functions import regexp_extract

from airbnb.spark import get_spark, get_SQLContext
from airbnb.utils import logging_init, logger

import wget
import findspark

__version__ = "0.1.0"

logging.getLogger('airbnb').addHandler(logging.NullHandler())
```

Run: airbnb

D:\Users\robin\OneDrive\miniconda3\envs\pyspark37\python.exe D:\Users\robin\OneDrive\Profile\Documents\dev\8795\airbnb\airbnb -m

Cannot find logging.ini file

21/04/02 14:02:16 INFO airbnb: Start of airbnb

21/04/02 14:02:16 DEBUG airbnb: Reading data

21/04/02 14:02:18 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

21/04/02 14:02:27 DEBUG airbnb: Preparing data

21/04/02 14:02:27 DEBUG airbnb: Creating model

Formula : price ~ accommodates + bedrooms + bathrooms + number\_of\_reviews + reviews\_per\_month + beds + availability\_30 + review\_scores\_rating

21/04/02 14:02:28 WARN Utils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.debug.maxToStringLength'.

21/04/02 14:02:29 WARN TaskSetManager: Stage 0 contains a task of very large size (6133 KB). The maximum recommended task size is 100 KB.

21/04/02 14:02:31 WARN Instrumentation: [ef40b8f6] regParam is zero, which might cause numerical instability and overfitting.

21/04/02 14:02:31 WARN TaskSetManager: Stage 1 contains a task of very large size (6133 KB). The maximum recommended task size is 100 KB.

[Stage 1: > (0 + 8) / 8] 21/04/02 14:02:34 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS

21/04/02 14:02:34 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeSystemLAPACK

21/04/02 14:02:34 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeRefLAPACK

21/04/02 14:02:34 WARN TaskSetManager: Stage 3 contains a task of very large size (6133 KB). The maximum recommended task size is 100 KB.

21/04/02 14:02:38 WARN TaskSetManager: Stage 5 contains a task of very large size (6133 KB). The maximum recommended task size is 100 KB.

21/04/02 14:02:41 WARN TaskSetManager: Stage 6 contains a task of very large size (6133 KB). The maximum recommended task size is 100 KB.

RMSE: 121.377308

r2: 0.350361

21/04/02 14:02:44 INFO airbnb: End of airbnb

Process finished with exit code 0