afs Documentation

Benchuang, Jacobchen

May 03, 2022

Contents

1 Documents **3**

2 Installation **5**
2.1 pip install on AFS notebook 5

3 Develop **7**
3.1 (For SDK developer) From sources 7
3.2 (For SDK developer) Build from source 7

4 Examples **9**
4.1 models 9

5 API Reference **15**
5.1 afs.models module 15

6 Indices and tables **19**

Python Module Index **21**

Index **23**

Documentation Status Build Status

CHAPTER 1

Documents

Reference documents Readthedocs

CHAPTER 2

Installation

Support python version 3.6 or later

2.1 pip install on AFS notebook

AFS provides the release version SDK on private pypi server. Run the following command on notebook cell to install SDK:

```
!pip install afs2-model
```

List the installed packages.

CHAPTER 3

Develop

3.1 (For SDK developer) From sources

1. Clone the repository to local.
2. To build the library run:

```
$ python setup.py install
```

3.2 (For SDK developer) Build from source

1. Clone the repository to local.
2. To build the wheel package:

```
$ python setup.py bdist_wheel
```

1. .whl will be in dist/

CHAPTER 4

Examples

4.1 models

4.1.1 upload_model

How to upload a model file(<5GB) on notebook.

Code

```
from afs import models

# Write a file as model file.
with open('model.h5', 'w') as f:
    f.write('dummy model')

# User-define evaluation result. Type:dict
extra_evaluation = {
    'confusion_matrix_TP': 0.9,
    'confusion_matrix_FP': 0.8,
    'confusion_matrix_TN': 0.7,
    'confusion_matrix_FN': 0.6,
    'AUC': 1.0
}

# User-define Tags. Type:dict
tags = {'machine': 'machine01'}

# User-define Feature Importance Type:list(dict)
feature_importance = [
        {'feature': 'petal_length', 'importance': 0.9473576807512394},
        {'feature': 'petal_width',  'importance': 0.038191635936882906},
        {'feature': 'sepal_length', 'importance': 0.011053241240641932},
        {'feature': 'sepal_width',  'importance': 0.0033974420712357825}
]
```

(continues on next page)

(continued from previous page)

```
coefficient = [
        {'feature': 'B-0070-0068-1-FN66F_strength', 'coefficient': -4.730741400252476}
↪,
        {'feature': 'B-0070-0068-1-FN66F_vendor', 'coefficient': -0.9335123601234512},
        {'feature': 'B-0070-0068-1-FN66F_tensile','coefficient': 0.16411707246054036},
        {'feature': 'B-0070-0068-1-FN66F_lot','coefficient': -0.08745686004816221},
        {'feature': 'Machine','coefficient': 0.015048547152059243},
        {'feature': 'Lot','coefficient': -0.010971975766858174},
        {'feature': 'RPM','coefficient': 0.0003730247816832932},
        {'feature': 'record_purpose','coefficient': 0.0}
]

# User-define RSA Public key Type:string
encrypt_key = """-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBAJrX4NIpPBdsOoFDo6oTUiRVM2CCYF/wdCy/KZ54mHyPRRFwtxcuRYt5omODY8uh
zTWjkb0jv0JphsHJmjeYkggHUxTrWeJ2gPfReTKPfmIGP0BQhHtwb92gxPYiJVHjSVgLLOe2
75iRyb7a5N20eiw5bEB4IFsuy+QXbDvUUsNPAgMBAAE=
-----END RSA PUBLIC KEY-----
"""

# Model object
afs_models = models()

# Upload the model to repository and the repository name is the same as file name,␣
↪the the following is optional parameters:
#   1. (optional) accuracy is a evaluation of the model by the result of testing.
#   2. (optional) loss is a evaluation of the model by the result of testing.
#   3. (optional) extra_evaluation is for other evaluations for the model, you can␣
↪put them to this parameter.
#   4. (optional) tags is the label for the model, like the time of data or the type␣
↪of the algorithm.
#   5. (optional) feature_importance is the record how the features important in the␣
↪model.
#        6. (optional) coefficient indicates the direction of the relationship␣
↪between a predictor variable and the response variable.
#   7. (optional) If there is a encrypt_key, use the encrypt_key to encrypt the model
afs_models.upload_model(
    model_path='model.h5',
        model_repository_name='model.h5',
        accuracy=0.4,
        loss=0.3,
        extra_evaluation=extra_evaluation,
        tags=tags,
        feature_importance=feature_importance,
        coefficient=coefficient,
        encrypt_key=encrypt_key
)

# Get the latest model info
model_info = afs_models.get_latest_model_info(model_repository_name='model.h5')

# See the model info
print(model_info)
```

results

```
{
        'uuid': '3369315c-d652-4c4d-b481-405be2ad5b33',
        'name': '3369315c-d652-4c4d-b481-405be2ad5b33',
        'model_repository': 'ef388859-64fb-4718-b90f-34defc8a3aae',
        'owner': '12345338-62b6-11ea-b1de-d20dfb084846',
        'evaluation_result': {
                'accuracy': 0.4,
                'loss': 0.3,
                'confusion_matrix_TP': 0.9,
                'confusion_matrix_FP': 0.8,
                'confusion_matrix_TN': 0.7,
                'confusion_matrix_FN': 0.6,
                'AUC': 1.0
        },
        'tags': {
                'machine': 'machine01',
                'is_encrypted': True
        },
        'feature_importance': [{
                'feature': 'petal_length',
                'importance': 0.9473576808
        }, {
                'feature': 'petal_width',
                'importance': 0.0381916359
        }, {
                'feature': 'sepal_length',
                'importance': 0.0110532412
        }, {
                'feature': 'sepal_width',
                'importance': 0.0033974421
        }],
        'coefficient': [
                {'feature': 'B-0070-0068-1-FN66F_strength', 'coefficient': -4.
↪730741400252476},
                {'feature': 'B-0070-0068-1-FN66F_vendor', 'coefficient': -0.
↪9335123601234512},
                {'feature': 'B-0070-0068-1-FN66F_tensile','coefficient': 0.
↪16411707246054036},
                {'feature': 'B-0070-0068-1-FN66F_lot','coefficient': -0.
↪08745686004816221},
                {'feature': 'Machine','coefficient': 0.015048547152059243},
                {'feature': 'Lot','coefficient': -0.010971975766858174},
                {'feature': 'RPM','coefficient': 0.0003730247816832932},
                {'feature': 'record_purpose','coefficient': 0.0}
        ],
        'size': 11,
        'created_at': '2020-04-06T10:23:56.228000+00:00'
}
```


4.1.2 get_latest_model_info

Code

```
from afs import models
afs_models = models()
afs_models.get_latest_model_info(model_repository_name='model.h5')
```

Output

```
{
        'evaluation_result': {
                'accuracy': 0.123,
                'loss': 0.123
        },
        'tags': {},
        'created_at': '2018-09-11 10:15:54'
}
```

4.1.3 download_model

Code

```
from afs import models

# Model object
afs_models = models()

# Download model from model repository, and get the last one model.
afs_models.download_model(
        save_path='dl_model.h5',
        model_repository_name='model.h5',
        last_one=True)

# Or get the specific model name in the model repository.
afs_models.download_model(
        save_path='dl_model.h5',
        model_repository_name='model.h5',
        model_name='3369315c-d652-4c4d-b481-405be2ad5b33')

# List the directory
!ls
```

Output

```
dl_model.h5
```

4.1.4 decrypt_model

Code

```
from afs import models

decrypt_key = """-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQCCCmzSJsR2kb5Tol7RyNM19XIbXsYYjzNSA0maJ3lPDNnJE0Kb6bP2Au+Y
QOT6y9Mse/hqKkeXjpMaOBbx+5fGM5ivEJiqepRBdD3mbkJMhvnrNTR4hzYpUgD69d+LzZkp
Q4GBHR93LRYvdUaBrSDXH0G6BLFe3AGF44LoWEa5fQIDAQABAoGBAICWFmzncLVeADlrAR+n
2VIt1htCZ9e5IiIiphEMn2OPbXrq1J6fRRgqZwjCgqmMCtCd9VHlZM10afkvJWE6SySBHR2T
qsMTuvJfGAEIRKo19p6BSqxpSRitP/Ow3liaND0i8MTJ5ixaiOhmRUX2jlWM1XfhuUjF6YGV
ProlojMBAkEA0s6eyxTUjazAEVCQ9BeCorwF/FM9Nf+4ZShOlLJZ+eRwFa5Epeziu/ecVbzQ
jzDI1KWlj+Gxl35zl03ooA5JrQJBAJ3rPG8CJFafxi2WcORIhGr2R/tcJRGjJcutMyGpi+zH
k7+eNRt9ZBowwCkjeJh+MJmKwjZ0NToS8jGox2LHyRECQFQ20L7mSmdynKQOIGoyvjBOlsGP
a0OYLczThljmywUGWjR/EtOKR6W5rE2gCV06qvAwYGyTSAPyMzE9oXHXY10CQQAO5g2aj4Is
```

(continues on next page)

(continued from previous page)

```
JgDFdkcKUokjqj6aSVQ5+MFtGNcVGvDXkvCuiFeMU2UpT2Yhu3X6NRWSttOh3Y7T/suYwcql
2CFxAkEAu+c3M1YSCpUZLdne9EW18/+4wWYNuFSOuGVK0FUyjjNhWzfDFARYNFgR+5mUMaJ0
4wNCyD5hvUVkBOZINHWtiw==
-----END RSA PRIVATE KEY-----
"""

# Model object
afs_models = models()

# Download model from model repository.
afs_models.download_model(
        save_path='dl_model.h5',
        model_repository_name='model.h5',
        last_one=True)

# Open the model and decrypt.
with open('dl_model.h5', 'rb') as f:
        data = f.read()
        model = afs_models.decrypt_model(data, decrypt_key
```

4.1.5 [Advanced] Token download_model

Code

```
from afs import models

# AFS connect info.
# Example format, CANNOT COPY AND PASTE.
# AFS API target endpoint
target_endpoint="https://api.afs.wise-paas.com"

# AFS service instance id
instance_id="123e4567-e89b-12d3-a456-426655440000"

# WISE-PaaS SSO token be gotten from SSO authentication
token="bearer␣
↪eyJhbGciOiJSUzI1NiIsImprdSI6Imh0dHBzOi8vdWFhLmFyZmEud2lzZS1wYWFzLmNvbS90b2tlbl9rZXlzIiwia2lkIjoia2V
↪eyJqdGkiOiJjZWExYTMwMGNjMmY0YzczYmMyNmY3Y2FiNTIwYjI4YSIsInN1YiI6IjhiNTJjODk0LTkyNmEtNDA4Mi1iNTdlLTc
↪R1SHUv8CIIoEN1pL5aGjxTn3OMB1rgjumD0hFFFrqNVIwcctN4QvNH1kK6G6SZyrlXvjU_
↪TXNDAbsAXiWLUkG7L60GZR2ZpJyPGNemZITjffuCKi0paQOrmAW5S0Nvn505G955DbuGDMGxQPOaorAcOkJYzFfAujSoZk3KMZn
↪Z96Fx5yhYnfXjT9aZDsASsx9I5UHYpunHRbzINJFx2PIxrYwCzfX2vFJqgeqeyeE1rjRsoS6GRj7eM3ud4YKQaC-
↪MK0TFttkTeRtPwggUJV51QhDmH03EYQ5qVFqsixE_zPGKFQb4wnTkdWGUOyBjoYSTuzk_dUZNbHGG"

# Model object
my_models = models(
    target_endpoint=target_endpoint,
    instance_id=instance_id,
    token=token,
)

# Download model from model repository, and get the last one model.
my_models.download_model(save_path='dl_model.h5', model_repository_name='model.h5',␣
↪last_one=True)

# Or get the specific model name in the model repository.
```

(continues on next page)

(continued from previous page)

```
my_models.download_model(save_path='dl_model.h5', model_repository_name='model.h5',␣
↪model_name='2019-07-10 02:59:11.610828')

# List the directory
!ls
```

Output

```
dl_model.h5
```

4.1.6 [Advanced] Download model from blob

Code

```
from afs import models

# Model object
afs_models = models()

# Example format, CANNOT COPY AND PASTE.
# Download model from blob
afs_models.download_model_from_blob(
        instance_id="123e4567-e89b-12d3-a456-426655440000",
        model_repository_id="7a54b92d-96d7-413c-968e-6c2c75251233",
        model_id="b2ea5261-743f-40bb-8e8f-643b2b3eecaf",
        save_path="dl_model.h5",
        blob_endpoint="https://minio.aifs.wise-paas.com",
        blob_accessKey="xxx",
        blob_secretKey="xxx",
        bucket_name="xxx",
    )
# List the directory
!ls
```

Output

```
dl_model.h5
```

CHAPTER 5

API Reference

5.1 afs.models module

class afs.models.**models**(*target_endpoint=None*, *instance_id=None*, *auth_code=None*, *token=None*)

Bases: afs.get_env.AfsEnv

create_model_repo(*model_repository_name*)

Create a new model repository. (Support v2 API)

Parameters **model_repository_name** (*str*) – (optional)The name of model repository.

Returns the new uuid of the repository

decrypt_model(*model*, *decrypt_key*)

Decrypt model.

Parameters

- **model** (*object*) – the object of model
- **decrypt_key** (*str*) – use decrypt_key to decrypt the model

Returns object

delete_model(*model_name*, *model_repository_name=None*)

Delete model.

Parameters

- **model_name** – model name.
- **model_repository_name** – model repository name.

Returns bool

delete_model_repository(*model_repository_name*)

Delete model repository.

Parameters **model_repository_name** – model repository name.

Returns bool

download_model (*save_path*, *model_repository_name=None*, *model_name=None*, *last_one=False*)
Download model from model repository to a file.

Parameters

- **model_repository_name** (*str*) – The model name exists in the model repository
- **save_path** (*str*) – The path exist in the file system
- **model_name** (*str*) – Get the specific model file from the model reposiotry, if getting last one value for None.
- **last_one** (*str*) – Get the last uploading model from the model repository.

download_model_from_blob (*instance_id*, *model_repository_id*, *model_id*, *save_path*, *blob_endpoint*, *blob_accessKey*, *blob_secretKey*, *bucket_name*)
API dowload model : instance_id: AIFS instance id : model_repository_id: model repository id in instance : model_id: model id in model repository : save_path: download filepath : blob_endpoint: blob : blob_accessKey: : blob_secretKey: : bucket_name:

get_latest_model_info (*model_repository_name=None*)
Get the latest model info, including created_at, tags, evaluation_result. (Support v2 API)

Parameters **model_repository_name** – (optional)The name of model repository.

Returns dict. the latest of model info in model repository.

get_model_id (*model_name=None*, *model_repository_name=None*, *last_one=True*)
Get model id by model name.

Parameters

- **model_name** (*str*) – model name. No need if last_one is true.
- **model_repository_name** (*str*) – model respository name where the model is.
- **last_one** (*bool*) – auto get the model_repository last one model

Returns str model id

get_model_info (*model_name*, *model_repository_name=None*)
Get model info, including created_at, tags, evaluation_result. (V2 API)

Parameters

- **model_name** – model name
- **model_repository_name** – The name of model repository.

Returns dict model info

get_model_repo_id (*model_repository_name=None*)
Get model repository by name.

Parameters **model_repository_name** (*str*) –

Returns str model repository id

set_blob_credential (*blob_endpoint*, *encode_blob_accessKey*, *encode_blob_secretKey*, *blob_record_id*, *bucket_name*)
Set blob credential when upload the big model.

Parameters

- **blob_endpoint** (*str*) – blob endpoint

- **encode_blob_accessKey** (*str*) – blob accessKey encode with base64
- **encode_blob_secretKey** (*str*) – blob secretKey encode with base64
- **blob_record_id** (*str*) – MD5 with instance_id + '_' + accessKey
- **bucket_name** (*str*) – blob bucket name

upload_model(*model_path*, *accuracy=None*, *loss=None*, *tags={}*, *extra_evaluation={}*, *feature_importance=None*, *coefficient=None*, *model_repository_name=None*, *model_name=None*, *encrypt_key=''*)

Upload model to model repository. (Support v2 API)

Parameters

- **model_path** (*str*) – (required) model filepath
- **accuracy** (*float*) – (optional) model accuracy value, between 0-1
- **loss** (*float*) – (optional) model loss value
- **tags** (*dict*) – (optional) tag from model
- **extra_evaluation** (*dict*) – (optional) other evaluation from model
- **model_name** (*str*) – (optional) Give model a name or a default name
- **model_repository_name** (*str*) – (optional) model_repository_name
- **feature_importance** (*list*) – (optional) feature_importance is the record how the features important in the model
- **coefficient** (*list*) – (optional) coefficient indicates the direction of the relationship between a predictor variable and the response
- **encrypt_key** (*str*) – (optional) If there is a encrypt_key, use the encrypt_key to encrypt the model

Returns dict. the information of the upload model.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

afs.models, 15

Index

A

afs.models (*module*), 15

C

create_model_repo() (*afs.models.models method*), 15

D

decrypt_model() (*afs.models.models method*), 15
delete_model() (*afs.models.models method*), 15
delete_model_repository() (*afs.models.models method*), 15
download_model() (*afs.models.models method*), 16
download_model_from_blob() (*afs.models.models method*), 16

G

get_latest_model_info() (*afs.models.models method*), 16
get_model_id() (*afs.models.models method*), 16
get_model_info() (*afs.models.models method*), 16
get_model_repo_id() (*afs.models.models method*), 16

M

models (*class in afs.models*), 15

S

set_blob_credential() (*afs.models.models method*), 16

U

upload_model() (*afs.models.models method*), 17