"""

Feature configuration for DeepBird jobs:

- Which features to keep

- Which features to blacklist

- Which features are labels

- Which feature is the weight

"""

from twitter.deepbird.io.legacy import feature\_config

class FeatureConfig(feature\_config.FeatureConfig):

def get\_feature\_spec(self):

"""

Generates a serialization-friendly dict representing this FeatureConfig.

"""

doc = super(FeatureConfig, self).get\_feature\_spec()

# Override the class in the spec.

doc["class"] = "twml.FeatureConfig"

return doc

class FeatureConfigBuilder(feature\_config.FeatureConfigBuilder):

def build(self):

# Overwrite self.build() to return twml.FeatureConfig instead

"""

Builds and returns FeatureConfig object.

"""

(

features,

tensor\_types,

sparse\_tensor\_types,

feature\_map,

feature\_name\_to\_feature\_parser,

feature\_in\_bq\_name,

) = self.\_build()

return FeatureConfig(

features=features,

labels=self.\_labels,

weight=self.\_weight,

filters=self.\_filter\_features,

tensor\_types=tensor\_types,

sparse\_tensor\_types=sparse\_tensor\_types,

feature\_types=feature\_map,

decode\_mode=self.\_decode\_mode,

legacy\_sparse=self.\_legacy\_sparse,

feature\_name\_to\_feature\_parser=self.\_feature\_name\_to\_feature\_parser,

feature\_in\_bq\_name=self.\_feature\_in\_bq\_name,

)

\_name\_to\_id = feature\_config.\_name\_to\_id