# checkstyle: noqa

from twml.feature\_config import FeatureConfigBuilder

def get\_feature\_config(data\_spec\_path, label):

return (

FeatureConfigBuilder(data\_spec\_path=data\_spec\_path, debug=True)

.batch\_add\_features(

[

("ebd.author\_specific\_score", "A"),

("ebd.has\_diff\_lang", "A"),

("ebd.has\_english\_tweet\_diff\_ui\_lang", "A"),

("ebd.has\_english\_ui\_diff\_tweet\_lang", "A"),

("ebd.is\_self\_tweet", "A"),

("ebd.tweet\_age\_in\_secs", "A"),

("encoded\_tweet\_features.favorite\_count", "A"),

("encoded\_tweet\_features.from\_verified\_account\_flag", "A"),

("encoded\_tweet\_features.has\_card\_flag", "A"),

# ("encoded\_tweet\_features.has\_consumer\_video\_flag", "A"),

("encoded\_tweet\_features.has\_image\_url\_flag", "A"),

("encoded\_tweet\_features.has\_link\_flag", "A"),

("encoded\_tweet\_features.has\_multiple\_hashtags\_or\_trends\_flag", "A"),

# ("encoded\_tweet\_features.has\_multiple\_media\_flag", "A"),

("encoded\_tweet\_features.has\_native\_image\_flag", "A"),

("encoded\_tweet\_features.has\_news\_url\_flag", "A"),

("encoded\_tweet\_features.has\_periscope\_flag", "A"),

("encoded\_tweet\_features.has\_pro\_video\_flag", "A"),

("encoded\_tweet\_features.has\_quote\_flag", "A"),

("encoded\_tweet\_features.has\_trend\_flag", "A"),

("encoded\_tweet\_features.has\_video\_url\_flag", "A"),

("encoded\_tweet\_features.has\_vine\_flag", "A"),

("encoded\_tweet\_features.has\_visible\_link\_flag", "A"),

("encoded\_tweet\_features.is\_offensive\_flag", "A"),

("encoded\_tweet\_features.is\_reply\_flag", "A"),

("encoded\_tweet\_features.is\_retweet\_flag", "A"),

("encoded\_tweet\_features.is\_sensitive\_content", "A"),

# ("encoded\_tweet\_features.is\_user\_new\_flag", "A"),

("encoded\_tweet\_features.language", "A"),

("encoded\_tweet\_features.link\_language", "A"),

("encoded\_tweet\_features.num\_hashtags", "A"),

("encoded\_tweet\_features.num\_mentions", "A"),

# ("encoded\_tweet\_features.profile\_is\_egg\_flag", "A"),

("encoded\_tweet\_features.reply\_count", "A"),

("encoded\_tweet\_features.retweet\_count", "A"),

("encoded\_tweet\_features.text\_score", "A"),

("encoded\_tweet\_features.user\_reputation", "A"),

("extended\_encoded\_tweet\_features.embeds\_impression\_count", "A"),

("extended\_encoded\_tweet\_features.embeds\_impression\_count\_v2", "A"),

("extended\_encoded\_tweet\_features.embeds\_url\_count", "A"),

("extended\_encoded\_tweet\_features.embeds\_url\_count\_v2", "A"),

("extended\_encoded\_tweet\_features.favorite\_count\_v2", "A"),

("extended\_encoded\_tweet\_features.label\_abusive\_hi\_rcl\_flag", "A"),

("extended\_encoded\_tweet\_features.label\_dup\_content\_flag", "A"),

("extended\_encoded\_tweet\_features.label\_nsfw\_hi\_prc\_flag", "A"),

("extended\_encoded\_tweet\_features.label\_nsfw\_hi\_rcl\_flag", "A"),

("extended\_encoded\_tweet\_features.label\_spam\_flag", "A"),

("extended\_encoded\_tweet\_features.label\_spam\_hi\_rcl\_flag", "A"),

("extended\_encoded\_tweet\_features.quote\_count", "A"),

("extended\_encoded\_tweet\_features.reply\_count\_v2", "A"),

("extended\_encoded\_tweet\_features.retweet\_count\_v2", "A"),

("extended\_encoded\_tweet\_features.weighted\_favorite\_count", "A"),

("extended\_encoded\_tweet\_features.weighted\_quote\_count", "A"),

("extended\_encoded\_tweet\_features.weighted\_reply\_count", "A"),

("extended\_encoded\_tweet\_features.weighted\_retweet\_count", "A"),

]

)

.add\_labels(

[

label, # Tensor index: 0

"recap.engagement.is\_clicked", # Tensor index: 1

"recap.engagement.is\_favorited", # Tensor index: 2

"recap.engagement.is\_open\_linked", # Tensor index: 3

"recap.engagement.is\_photo\_expanded", # Tensor index: 4

"recap.engagement.is\_profile\_clicked", # Tensor index: 5

"recap.engagement.is\_replied", # Tensor index: 6

"recap.engagement.is\_retweeted", # Tensor index: 7

"recap.engagement.is\_video\_playback\_50", # Tensor index: 8

"timelines.earlybird\_score", # Tensor index: 9

]

)

.define\_weight("meta.record\_weight/type=earlybird")

.build()

)