In [ ]:
```
import boto3
import numpy as np
import argparse
import ast
import json
import ember
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import RobustScaler
```

In [ ]:
```
def prot_buffer(directory):
    shutil.rmtree(export_dir)
    build = builder.SavedModelBuilder(directory)

def load_save_model(model,weight,path):
    json_file = open(model, 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json,custom_objects={"GlorotUniform": t
    loaded_model.load_weights(weight)
    build = prot_buffer(directory)
    signature = predict_signature_def(inputs={"inputs": loaded_model.input}, outputs={"
    return build,signature

build,signature = load_save_model('model1.json','weights1.h5','cloud/')
```

In [ ]:
```
from keras import backend as K

with K.get_session() as sess:
    # Save the meta graph and variables
    build.add_meta_graph_and_variables(
        sess=sess, tags=[tag_constants.SERVING], signature_def_map={"serving_default":
    build.save()


with tarfile.open('model.tar.gz', mode='w:gz') as archive:
    archive.add('cloud', recursive=True)


sagemaker_session = sagemaker.Session()
inputs = sagemaker_session.upload_data(path='model.tar.gz', key_prefix='model')


sagemaker_model = TensorFlowModel(model_data = 's3://' + sagemaker_session.default_buck
                                  role = role,
                                  framework_version = '1.12',
                                  entry_point = 'train.py')
```

In [ ]:
```
%%time
predictor = sagemaker_model.deploy(initial_instance_count=1,
                                   instance_type='ml.m5.xlarge')
predictor.endpoint
```