



STORE ITEM DEMAND FORECASTING USING DEEP LEARNING



-
- Demand Forecasting
 - Why is it needed?

DEMAND





Motivation

- Inventory Management
 - Resource Management
-



Related Work [1]

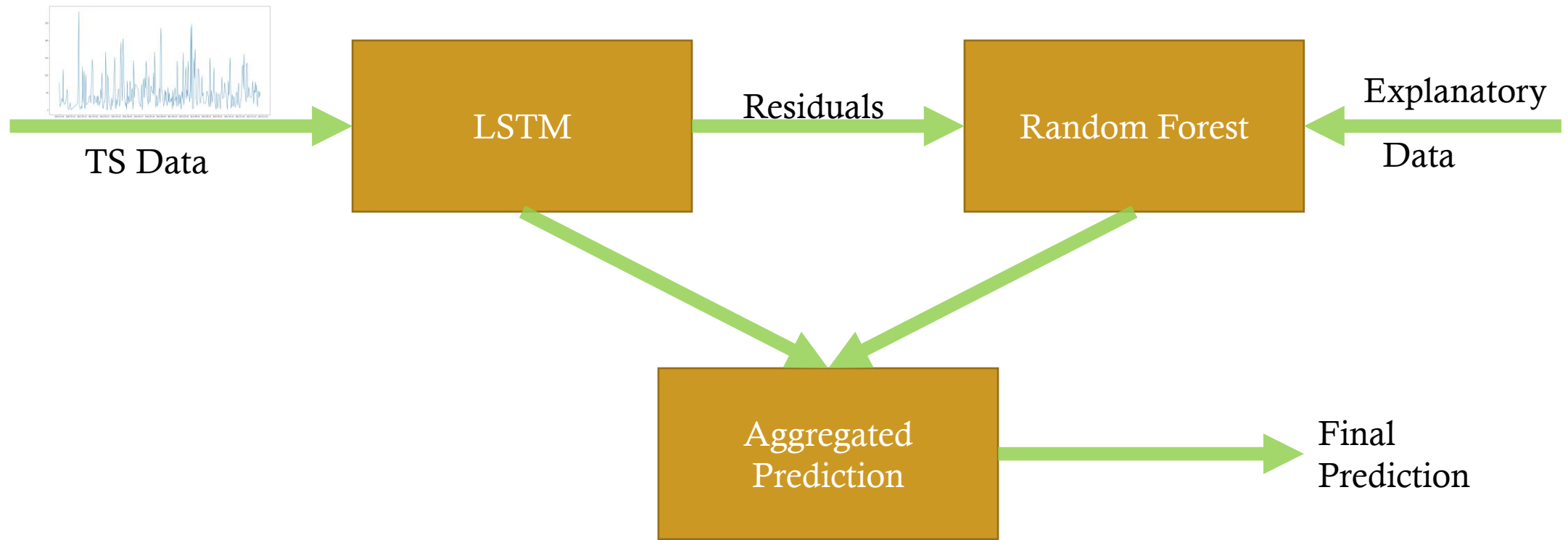
- ANN works efficiently over traditional methods.
-

Related Work [2]

- LSTM is best among all ANN for Time series data forecasting.



Method



```
In [9]: def createmodel(n_layers = 2, activation_func = "linear", opt = "sgd"):
        model = keras.models.Sequential()

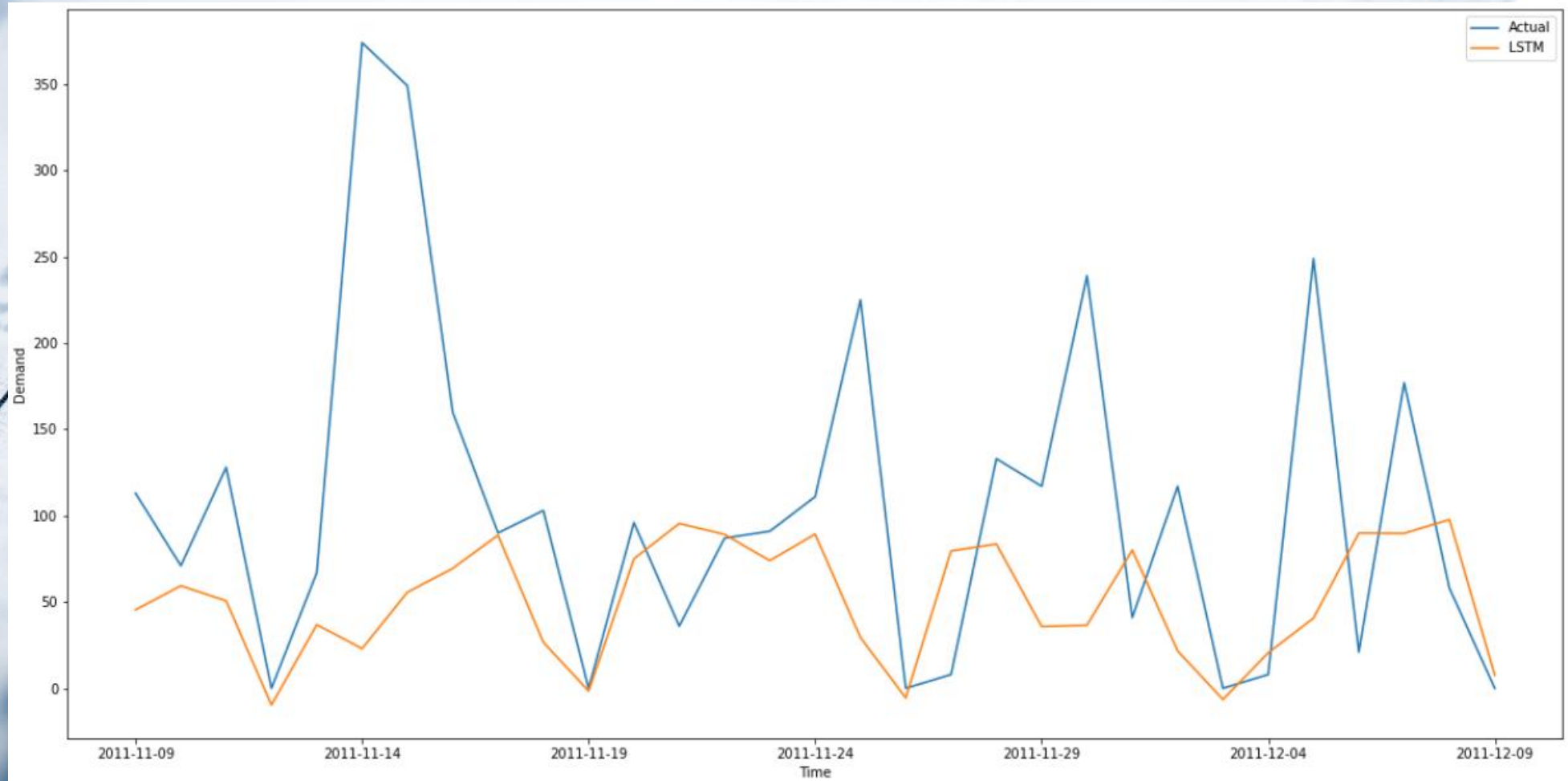
        model.add(keras.layers.Lambda(lambda x: tf.expand_dims(x, axis=-1),
                                       input_shape=[None]))
        for i in range(n_layers):
            if i != n_layers - 1:
                model.add(keras.layers.Bidirectional(tf.keras.layers.LSTM(50, return_sequences=True)))
            else:
                model.add(keras.layers.Bidirectional(tf.keras.layers.LSTM(50)))

        model.add(keras.layers.Dense(1, activation=activation_func))
        model.add(keras.layers.Lambda(lambda x: x * 100.0))
        model.compile(optimizer = opt, loss='huber_loss', metrics = ["mae"])

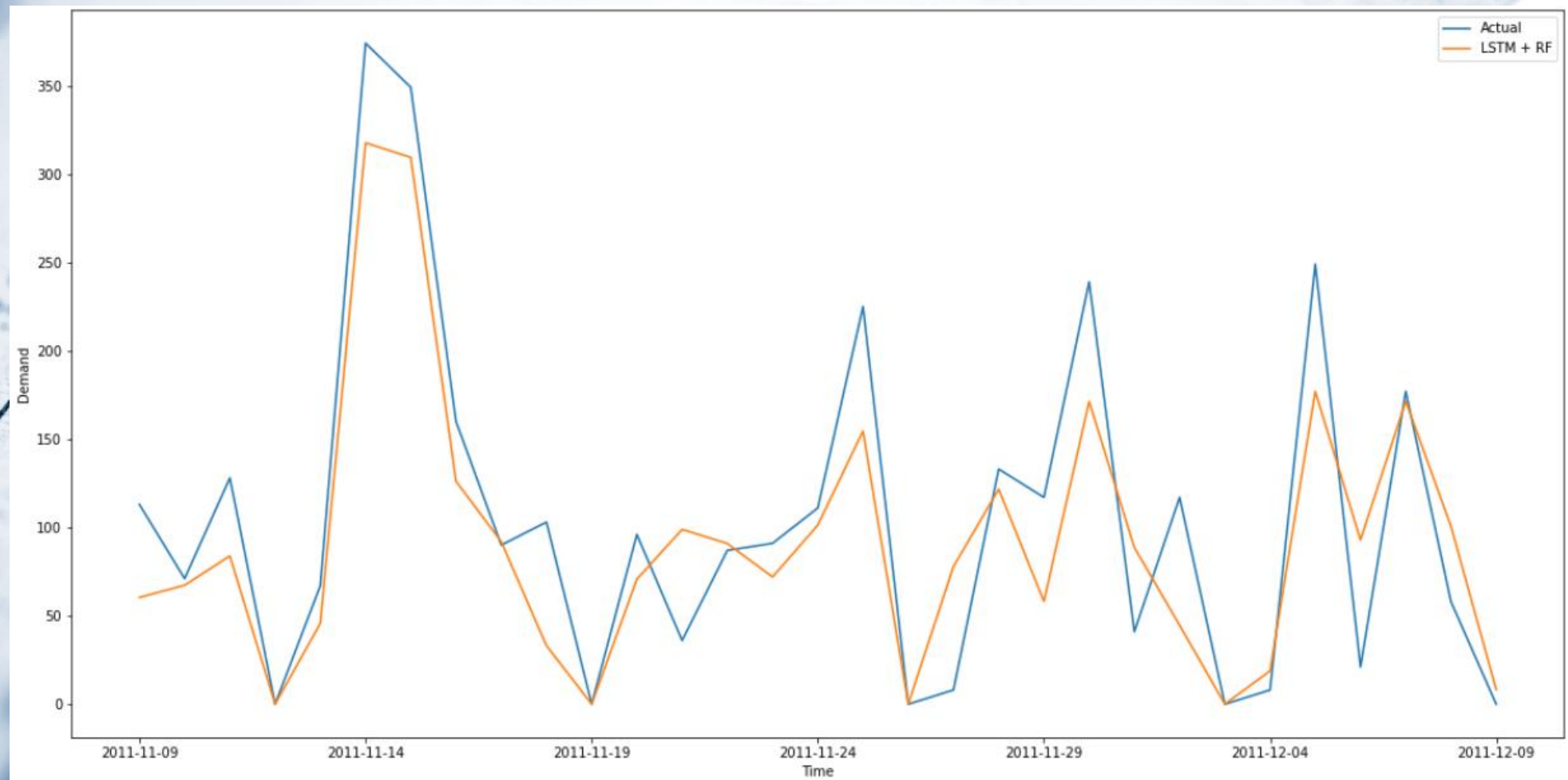
        return model

model = KerasRegressor(model=createmodel,n_layers = 2, activation_func = "linear", opt = "sgd",verbose=False,random_state=0)
```

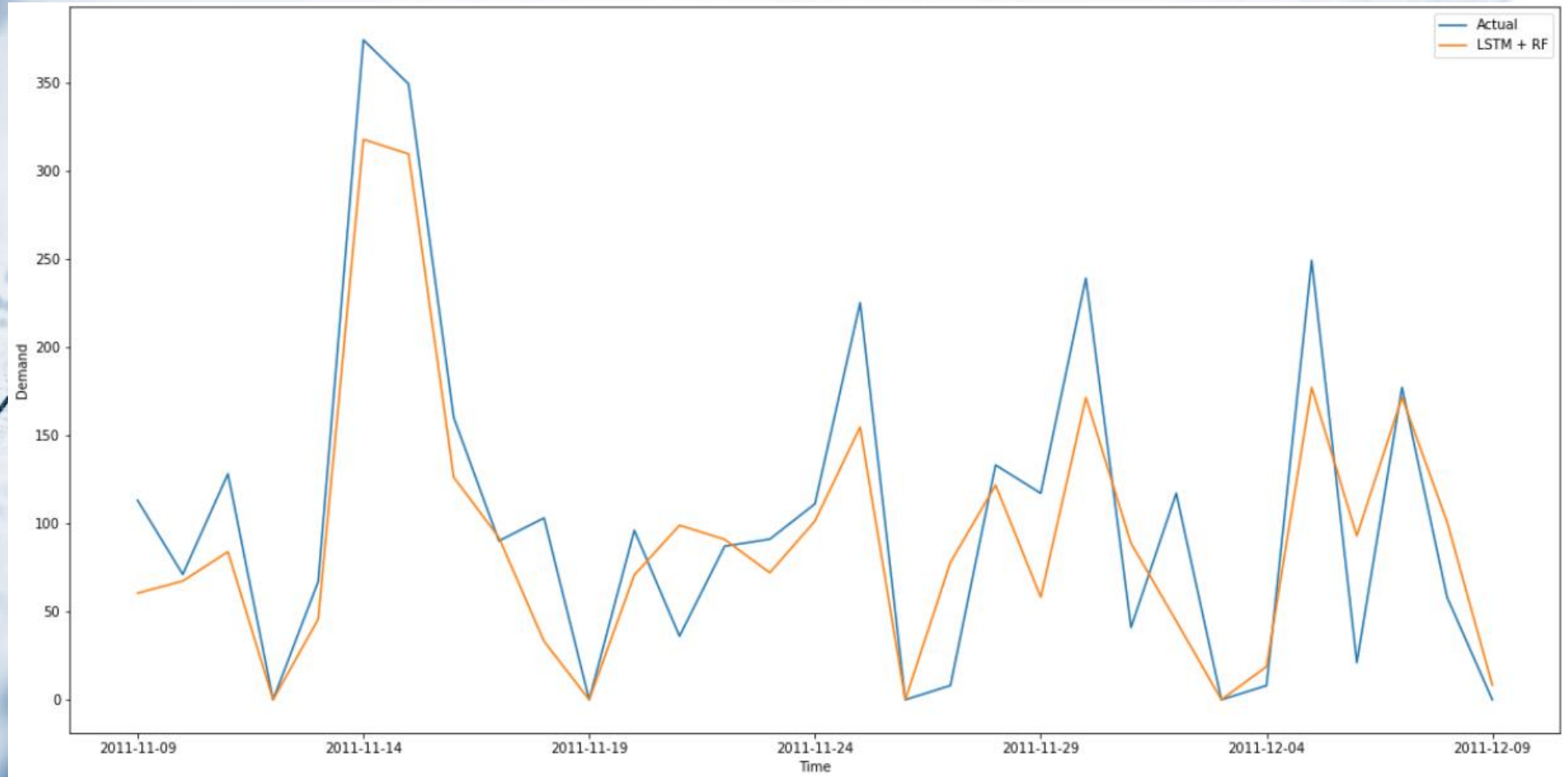
```
In [10]: activation_funcs = ['linear', 'relu', 'tanh']
        optimizers = ["rmsprop", "adam", "sgd"]
        param_grid = {"n_layers": [2,3], "activation_func" : activation_funcs, "opt" : optimizers, "batch_size" : [5], "epochs" : [100]}
        cv = ShuffleSplit(n_splits=1, test_size=0.09, random_state=0)
        grid = GridSearchCV(estimator = model, param_grid = param_grid, cv=cv)
```



RESULTS ACTUAL AND LSTM – MAE 74.28



RESULTS ACTUAL AND LSTM + RF – MAE 34



RESULTS ACTUAL AND LSTM + RF – MAE 34

Thank You!!!
