model = Sequential()

model.add(Dense(128, input\_dim=num\_features, activation='relu'))

model.add(Dense(32, activation = 'relu'))

model.add(Dense(64, activation = 'relu'))

model.add(Dense(1, activation = 'relu'))

model.compile(optimizer='adam', loss='mean\_squared\_error', metrics=['mse', 'mae', 'mape', 'cosine'])

model.summary()

25 epoche

LOSS: 0.128387

ROOT MEAN SQUARED LOGARITHMIC ERROR: 0.357

SUBMISSION: 0.53970

CLUSTER: 40

**100 epoche**

**LOSS: 0.1172**

**ROOT MEAN SQUARED LOGARITHMIC ERROR: 0.3430**

**SUBMISSION: 0.43055**

**CLUSTER: 50**

100 epoche con CLUSTER : 100 non va molto bene

model = Sequential()

model.add(Dense(128, input\_dim=num\_features, activation='relu'))

model.add(Dense(32, activation = 'relu'))

model.add(Dense(64, activation = 'relu'))

model.add(Dense(32, activation = 'relu'))

model.add(Dense(1, activation = 'relu'))

model.compile(optimizer='adam', loss='mean\_squared\_error', metrics=['mse', 'mae', 'mape', 'cosine'])

model.summary()

LOSS: 0.1268

ROOT MEAN SQUARED LOGARITHMIC ERROR: 0.354

SUBMISSION:

CLUSTER: 40

model = Sequential()

model.add(Dense(256, input\_dim=num\_features, activation='relu'))

model.add(Dense(64, activation = 'relu'))

model.add(Dense(64, activation = 'relu'))

model.add(Dense(128, activation = 'relu'))

model.add(Dense(32, activation = 'relu'))

model.add(Dense(1, activation = 'relu'))

model.compile(optimizer='adam', loss='mean\_squared\_error', metrics=['mse', 'mae', 'mape', 'cosine'])

model.summary()

LOSS: 0.1226

ROOT MEAN SQUARED LOGARITHMIC ERROR: 0.351

SUBMISSION: 0.52

CLUSTER: 40

model = Sequential()

model.add(Dense(256, input\_dim=num\_features, activation='relu'))

model.add(Dense(128, activation = 'relu'))

model.add(Dense(64, activation = 'relu'))

model.add(Dense(128, activation = 'relu'))

model.add(Dense(64, activation = 'relu'))

model.add(Dense(32, activation = 'relu'))

model.add(Dense(1, activation = 'relu'))

model.compile(optimizer='adam', loss='mean\_squared\_error', metrics=['mse', 'mae', 'mape', 'cosine'])

model.summary()

con 25 epoche

LOSS: 0.1213

ROOT MEAN SQUARED LOGARITHMIC ERROR: 0.3488

SUBMISSION: 0.5268

CLUSTER: 40

con 100 epoche

LOSS: 0.1130

ROOT MEAN SQUARED LOGARITHMIC ERROR: 0.3417

SUBMISSION: 0.563

CLUSTER: 40