Get the data

In [ ]:
```
!gdown --id 1HVSazFk8m553VWPjFnZZ-YfJA_KecPea
!unzip translated_data_updated.zip
```

```
Downloading...
From: https://drive.google.com/uc?id=1HVSazFk8m553VWPjFnZZ-YfJA_KecPea
To: /content/translated_data_updated.zip
100% 122M/122M [00:00<00:00, 198MB/s]
Archive:  translated_data_updated.zip
replace data_translated/coupon_visit_train.csv? [y]es, [n]o, [A]ll, [N]one, [r]e
name: N
```

In [ ]:
```
# imports
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

import tensorflow as tf
from tensorflow import keras
import sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

sns.set_theme(context='notebook', style='darkgrid')
mpl.rcParams['figure.figsize'] = (12, 10)
colors = plt.rcParams['axes.prop_cycle'].by_key()['color']
```

In [ ]:
```
# Important Note:
# Visits = browsing history in the training period. No test set available.
# Purchases = purchase history in the training period. No test set available.

df_users        = pd.read_csv('data_translated/user_list.csv')
df_c_list_train = pd.read_csv('data_translated/coupon_list_train.csv')
df_c_list_test  = pd.read_csv('data_translated/coupon_list_test.csv')
df_area_train   = pd.read_csv('data_translated/coupon_area_train.csv')
df_area_test    = pd.read_csv('data_translated/coupon_area_test.csv')
df_visit_train  = pd.read_csv('data_translated/coupon_visit_train.csv')
df_purch_train  = pd.read_csv('data_translated/coupon_detail_train.csv')
df_locations    = pd.read_csv('data_translated/prefecture_locations.csv')
```

Feature Engineering

User List

In [ ]:
```
# rename SEX_ID column, change to categorical value (0 Male, 1 Female)
df_users['SEX'] = df_users['SEX_ID'].replace('f', 1)
df_users['SEX'] = df_users['SEX'].replace('m', 0)
```

In [ ]:
```
# create a categorical variable for age group:
```

```
# 14-21, 22-35, 36-49, 50-65, 66-75, 76-90
def age_cat(age):
  if age <= 21:
    return 0
  elif age <= 35:
    return 1
  elif age <= 49:
    return 2
  elif age <= 65:
    return 3
  elif age <= 75:
    return 4
  elif age <= 90:
    return 5
  else:
    return 6

lbl_age_ranges = ['14-21', '22-35', '36-49', '50-65', '66-75', '76-90']
```

In [ ]:

```
# Data prep the columns
df_users['AGE_GROUP'] = [age_cat(a) for a in df_users['AGE']]
df_users.columns = ['un', 'reg_date', 'sex_id', 'age', 'withdraw_date', 'user_id
df_users = df_users[['user_id', 'age_group', 'sex', 'pref_name']].fillna(0)

df_purch_train.columns = ['un', 'count', 'date', 'purchase_id', 'user_id', 'coup
df_purch_train = df_purch_train[['purchase_id', 'count', 'user_id', 'coupon_id',

df_coupons = df_c_list_train
df_coupons.columns = ['un', 'discount_rate', 'cat_price', 'discount_price', 'dis
df_coupons = df_coupons[['coupon_id', 'discount_rate', 'discount_price', 'capsul
```

In [ ]:

```
merged_df = df_purch_train.set_index('purchase_id').join(df_coupons.set_index('c
merged_df = merged_df.join(df_users.set_index('user_id'), on='user_id').reset_in
```

In [ ]:

```
test_set_df = df_c_list_test
test_set_df.columns = ['un', 'discount_rate', 'cat_price', 'discount_price', 'di
test_set_df = test_set_df[['coupon_id', 'discount_rate', 'discount_price', 'caps
```

In [ ]:

```
merged_df = merged_df.drop(columns=['small_area_purchase'])
```

In [ ]:

```
# rename columns one final time
merged_df.columns = ['purchase_id', 'count', 'user_id', 'coupon_id', 'discount_r
merged_df.head()
```

Out[ ]:

	purchase_id	count	user_id	
0	c820a8882374a4e472f0984a8825893f	1	d9dca3cb44bab12ba313eaa681f663eb	34c48f84026
1	1b4eb2435421ede98c8931c42e8220ec	1	560574a339f1b25e57b0221e486907ed	767673b7a77

	purchase_id	count	user_id	
2	36b5f9ba46c44b65587d0b16f2e4c77f	1	560574a339f1b25e57b0221e486907ed	4f3b5b91d9
3	2f30f46937cc9004774e576914b2aa1a	1	560574a339f1b25e57b0221e486907ed	4f3b5b91d9
4	4d000c64a55ac573d0ae1a8f03677f50	1	560574a339f1b25e57b0221e486907ed	4f3b5b91d9

Get User's Purchased Coupons

In [ ]:
```
# get a reference table for one-hot encoding
df_merge_ohe = pd.get_dummies(merged_df, columns=['capsule_text', 'genre', 'larg
df_merge_ohe = df_merge_ohe.drop(columns=['purchase_id', 'count'])
df_merge_ohe.shape
```

Out[ ]:
```
(168996, 208)
```

In [ ]:
```
# also filter and one-hot the test set concatenated with user data
def get_test_set(user):
  df_test_ohe = df_c_list_test.fillna(0)
  df_test_ohe = df_test_ohe[['coupon_id', 'discount_rate', 'discount_price', 'ca
  df_test_ohe.columns = ['coupon_id', 'discount_rate', 'discount_price', 'capsul
  df_test_ohe = df_test_ohe.set_index('coupon_id')

  ## add user data
  df_test_ohe['age_group'] = user['age_group']
  df_test_ohe['sex'] = user['sex']
  df_test_ohe['user_prefecture'] = user['pref_name']

  df_test_ohe = pd.get_dummies(df_test_ohe, columns=['capsule_text', 'genre', 'l
  df_test_ohe = df_test_ohe.reset_index('coupon_id').reindex(columns=df_merge_oh
  return df_test_ohe
```

In [ ]:
```
from sklearn.metrics.pairwise import cosine_similarity

def cos_sim(user, user_coupon, test_coupons):
  user_s = user_coupon.drop(index=['coupon_id'])
  test_df = test_coupons.drop(columns=['coupon_id', 'user_id'])

  cs = cosine_similarity([user_s], test_df, dense_output=True)

  coupon_id_list = []
  cosine_list = []
  for i, c in test_coupons.iterrows():
    coupon_id_list.append(c['coupon_id'])
    cosine_list.append(cs[0][i])

  return coupon_id_list, cosine_list
```

In [ ]:
```
from tqdm import tqdm
```

```
predictions = []

for i, u in tqdm(df_users.iterrows(), total=len(df_users)):
  bought_coupons_df = df_merge_ohe[df_merge_ohe['user_id'] == u.user_id]
  bought_coupons_df = bought_coupons_df.drop(columns=['user_id'])
  test_coupons_df = get_test_set(u).fillna(0)

  coupon_list = []
  score_list = []
  for j, bought_coupon in bought_coupons_df.iterrows(): # for each users' purcha
    coupons, scores = cos_sim(u, bought_coupon, test_coupons_df)
    coupon_list.append(coupons)
    score_list.append(scores)

  results_df = pd.DataFrame(columns=['coupon_id', 'score'])
  results_df['coupon_id'] = np.ravel(coupon_list)
  results_df['score'] = np.ravel(score_list)
  results_df = results_df.drop_duplicates().sort_values(by='score', ascending=Fa

  coupons_string = ' '.join(results_df['coupon_id']).strip()

  # Add it to the user's file
  # Get top 10 similarity coupons
  predictions.append({'USER_ID_hash': u.user_id, 'PURCHASED_COUPONS': coupons_st

predictions_df = pd.DataFrame.from_dict(predictions)
predictions_df.to_csv('submission_cosine.csv', header=True, index=False)
predictions_df
```

```
100%|██████████| 22873/22873 [2:53:18<00:00,  2.20it/s]
```

Out[ ]:

	USER_ID_hash	PURCHASED_COUPONS
0	d9dca3cb44bab12ba313eaa681f663eb	c0d22b2252fa23eb3c44d8edce1804fb ffe734ef0b1d8...
1	560574a339f1b25e57b0221e486907ed	3905228fb8cac640b673f71d5f315df5 784c1314b9f64...
2	e66ae91b978b3229f8fd858c80615b73	db7c52cbb13947dd532fcd4253d794f2 e4db7645ae556...
3	43fc18f32eafb05713ec02935e2c2825	c0d22b2252fa23eb3c44d8edce1804fb 0e917a0e87224...
4	dc6df8aa860f8db0d710ce9d4839840f	4470e4b7e6f9f7bee5c8a6738d63b757 cb4c67c749dc5...
...	...	...
22868	2f0a2f36a9f63b6ba2fa3a7e53bef906	128ad3628350e513914a2cd7d9c1e17b 4c973e37ebd1c...
22869	6ae7811a9c7c58546d6a1567ab098c21	70987622f5824a3b209e97b32021e50b fe3dfe6334edd...
22870	a417308c6a79ae0d86976401ec2e3b04	ca8ea3d52ca939d6ab1b9c792baa6169 ffe734ef0b1d8...
22871	4937ec1c86e71d901c4ccc0357cff0b1	64b92e53b6e56f7f7bd158ec31887f3d 4c0aa767668e1...
22872	280f0cedda5c4b171ee6245889659571	db7c52cbb13947dd532fcd4253d794f2 09ac6e78e77fa...

22873 rows × 2 columns

In [ ]:
```
predictions_df
```

Out[ ]:

	USER_ID_hash	PURCHASED_COUPONS
0	d9dca3cb44bab12ba313eaa681f663eb	c0d22b2252fa23eb3c44d8edce1804fb ffe734ef0b1d8...
1	560574a339f1b25e57b0221e486907ed	3905228fb8cac640b673f71d5f315df5 784c1314b9f64...
2	e66ae91b978b3229f8fd858c80615b73	db7c52cbb13947dd532fcd4253d794f2 e4db7645ae556...
3	43fc18f32eafb05713ec02935e2c2825	c0d22b2252fa23eb3c44d8edce1804fb 0e917a0e87224...
4	dc6df8aa860f8db0d710ce9d4839840f	4470e4b7e6f9f7bee5c8a6738d63b757 cb4c67c749dc5...
...	...	...
22868	2f0a2f36a9f63b6ba2fa3a7e53bef906	128ad3628350e513914a2cd7d9c1e17b 4c973e37ebd1c...
22869	6ae7811a9c7c58546d6a1567ab098c21	70987622f5824a3b209e97b32021e50b fe3dfe6334edd...
22870	a417308c6a79ae0d86976401ec2e3b04	ca8ea3d52ca939d6ab1b9c792baa6169 ffe734ef0b1d8...
22871	4937ec1c86e71d901c4ccc0357cff0b1	64b92e53b6e56f7f7bd158ec31887f3d 4c0aa767668e1...
22872	280f0cedda5c4b171ee6245889659571	db7c52cbb13947dd532fcd4253d794f2 09ac6e78e77fa...

22873 rows × 2 columns

In [ ]:
```
predictions_df.to_csv('submissions_cosine_pred.csv', header=True, index=False)
```

In [ ]:
```

```