About the Machine Leaerning Program

This Simple Machine learning Program will be used to predict protein function using Gene Ontology, and this program can be used by bilogist and drugmakers to make life saving drugs ang vaccines

Importing all neceassary files

```
import os
import gc
import plotly.express as px
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, Dropout,LSTM
from tensorflow.keras.models import Model, Sequential
from sklearn.model_selection import
train_test_split,RandomizedSearchCV
from sklearn.metrics import roc_auc_score

import xgboost as xgb
import numpy as np
import pandas as pd

from tqdm import tqdm
tqdm.pandas()

# annoy for approximate nearest neighbors
from annoy import AnnoyIndex

import gc
```

```
/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/
__init__.py:98: UserWarning: unable to load
libtensorflow_io_plugins.so: unable to open file:
libtensorflow_io_plugins.so, from paths:
['/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/
libtensorflow_io_plugins.so']
caused by:
['/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/
libtensorflow_io_plugins.so: undefined symbol:
_ZN3tsl6StatusC1EN10tensorflow5error4CodeESt17basic_string_viewIcSt11c
har_traitsIcEENS_14SourceLocationE']
  warnings.warn(f"unable to load libtensorflow_io_plugins.so: {e}")
/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/__ini
```

```
t__.py:104: UserWarning: file system plugins are not loaded: unable to
open file: libtensorflow_io.so, from paths:
['/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/
libtensorflow_io.so']
caused by:
['/opt/conda/lib/python3.10/site-packages/tensorflow_io/python/ops/
libtensorflow_io.so: undefined symbol:
_ZTVN10tensorflow13GcsFileSystemE']
  warnings.warn(f"file system plugins are not loaded: {e}")
```

```
Data_dir  = '/kaggle/input/cafa-5-protein-function-prediction'
Max_labels= 1500

train_terms = pd.read_csv(os.path.join(Data_dir, 'Train',
'train_terms.tsv'), sep='\t')

terms = train_terms.groupby(['aspect', 'term'])
['term'].count().reset_index(name='frequency')
print(terms.groupby('aspect')['term'].nunique())
```

```
aspect
BPO    21285
CCO     2957
MFO     7224
Name: term, dtype: int64
```

Modifying the Dataset to reduce Memory

```
def reduce_mem_usage(df):
    """ iterate through all the columns of a dataframe and modify the
data type
        to reduce memory usage.
    """
    start_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage of dataframe is {:.2f} MB'.format(start_mem))

    for col in df.columns:
        col_type = df[col].dtype

        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max <
np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max <
np.iinfo(np.int16).max:
```

```
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max <
np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max <
np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max <
np.finfo(np.float16).max:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max <
np.finfo(np.float32).max:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
        else:
            df[col] = df[col].astype('category')

    end_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage after optimization is: {:.2f}
MB'.format(end_mem))
    print('Decreased by {:.1f}%'.format(100 * (start_mem - end_mem) /
start_mem))

    return df

px.histogram(reduce_mem_usage(train_terms), x="aspect")
```

```
Memory usage of dataframe is 122.77 MB
Memory usage after optimization is: 42.17 MB
Decreased by 65.6%
```

```
fractions = (terms.groupby('aspect')['term'].nunique() /
terms['term'].nunique() * Max_labels).apply(round)
print(fractions)

selected_terms = set()
for aspect, number in fractions.items():
    selection = terms.loc[(terms.aspect == aspect)]
    selection = selection.nlargest(number, columns='frequency',
keep='first')
    selected_terms.update(selection.term.to_list())
```

```
aspect
BPO    1015
CCO     141
MFO     344
Name: term, dtype: int64
```

```
print(selected_terms)
```

{'GO:0046915', 'GO:0098916', 'GO:0005615', 'GO:0048585', 'GO:0016236',
'GO:0008170', 'GO:0034660', 'GO:0048736', 'GO:0050660', 'GO:0051536',
'GO:0030425', 'GO:0098660', 'GO:0019829', 'GO:0007626', 'GO:0042326',
'GO:0098852', 'GO:0030198', 'GO:0035821', 'GO:0009267', 'GO:0031032',
'GO:0030163', 'GO:0021953', 'GO:0006816', 'GO:0050803', 'GO:0033218',
'GO:0022853', 'GO:0004930', 'GO:0014706', 'GO:0030155', 'GO:0035270',
'GO:0034762', 'GO:0043069', 'GO:0051049', 'GO:0007560', 'GO:0071478',
'GO:0010605', 'GO:0055080', 'GO:0044057', 'GO:0009117', 'GO:0008092',
'GO:0007051', 'GO:0007517', 'GO:0030312', 'GO:0000785', 'GO:0097190',
'GO:1903522', 'GO:0046777', 'GO:0099537', 'GO:1901607', 'GO:0015980',
'GO:0004672', 'GO:0044242', 'GO:0009755', 'GO:0016757', 'GO:0016567',
'GO:1903532', 'GO:0003700', 'GO:0015629', 'GO:0070851', 'GO:0043621',
'GO:0044087', 'GO:0101005', 'GO:0002697', 'GO:0009890', 'GO:0051050',
'GO:0006139', 'GO:0000793', 'GO:0044419', 'GO:1901617', 'GO:0022411',
'GO:0048545', 'GO:0050954', 'GO:0008654', 'GO:0051146', 'GO:0030427',
'GO:0098609', 'GO:1903829', 'GO:0016747', 'GO:0042592', 'GO:0120036',
'GO:0016758', 'GO:0006468', 'GO:0005886', 'GO:0008150', 'GO:0098813',
'GO:0035239', 'GO:0016209', 'GO:0005911', 'GO:0048827', 'GO:0006486',
'GO:0030855', 'GO:0071214', 'GO:0016922', 'GO:0016773', 'GO:0070925',
'GO:0034654', 'GO:0008234', 'GO:0009790', 'GO:0098960', 'GO:0010628',
'GO:0036477', 'GO:0032880', 'GO:0090092', 'GO:0005267', 'GO:0051604',
'GO:0120039', 'GO:0051336', 'GO:0015078', 'GO:0034976', 'GO:0043066',
'GO:0007017', 'GO:0050770', 'GO:0031668', 'GO:0032879', 'GO:0005253',
'GO:0099402', 'GO:0034329', 'GO:0061061', 'GO:0035770', 'GO:0007276',
'GO:0019787', 'GO:0048812', 'GO:0030659', 'GO:0010498', 'GO:0015075',
'GO:0005819', 'GO:0000977', 'GO:0046483', 'GO:0042325', 'GO:0071466',
'GO:0030534', 'GO:0150034', 'GO:0051213', 'GO:0006082', 'GO:0005777',
'GO:0030545', 'GO:0071705', 'GO:0007163', 'GO:0016811', 'GO:0009991',
'GO:0031175', 'GO:0048029', 'GO:0016070', 'GO:0042063', 'GO:1901698',
'GO:0016835', 'GO:0031327', 'GO:0140096', 'GO:0033674', 'GO:0006914',
'GO:0098796', 'GO:0004659', 'GO:1901699', 'GO:0008173', 'GO:0060538',
'GO:1903037', 'GO:0005634', 'GO:0030234', 'GO:0009056', 'GO:0034220',
'GO:0018958', 'GO:0006721', 'GO:0006401', 'GO:0120025', 'GO:0015179',
'GO:0045859', 'GO:0048522', 'GO:0015630', 'GO:0032501', 'GO:0140657',
'GO:0032868', 'GO:0048229', 'GO:0009651', 'GO:0045664', 'GO:0090257',
'GO:0048701', 'GO:1901615', 'GO:0008168', 'GO:0003713', 'GO:0016763',
'GO:0090304', 'GO:1901575', 'GO:0042887', 'GO:0044782', 'GO:0019900',
'GO:0010256', 'GO:0019904', 'GO:0005543', 'GO:0040011', 'GO:0032559',
'GO:0051015', 'GO:0005215', 'GO:0006605', 'GO:0016042', 'GO:0048477',
'GO:0044264', 'GO:0099513', 'GO:0051046', 'GO:0048592', 'GO:0048639',
'GO:0140297', 'GO:0001667', 'GO:0000278', 'GO:0048706', 'GO:0005342',
'GO:0071554', 'GO:0051249', 'GO:0042802', 'GO:0009058', 'GO:0008380',
'GO:0009165', 'GO:0071248', 'GO:0000323', 'GO:0002253', 'GO:0048018',
'GO:0002682', 'GO:0005667', 'GO:0048737', 'GO:0017076', 'GO:0140030',
'GO:0040013', 'GO:0050769', 'GO:0004812', 'GO:0051701', 'GO:0001934',
'GO:0019213', 'GO:0071241', 'GO:0060627', 'GO:0005230', 'GO:0000228',
'GO:0010562', 'GO:0098657', 'GO:0009798', 'GO:0000287', 'GO:0071944',
'GO:0030100', 'GO:0043933', 'GO:0097305', 'GO:0051171', 'GO:0000976',
'GO:0050804', 'GO:0016324', 'GO:0022603', 'GO:0048471', 'GO:0045165',
'GO:0051347', 'GO:0005575', 'GO:0035639', 'GO:0010646', 'GO:0050678',

'GO:0006091', 'GO:2000113', 'GO:0009266', 'GO:0048880', 'GO:0016874',
'GO:0019903', 'GO:0042110', 'GO:0016020', 'GO:1901361', 'GO:0033643',
'GO:0045892', 'GO:0019887', 'GO:0007519', 'GO:0009628', 'GO:0005622',
'GO:0031072', 'GO:1901990', 'GO:0008509', 'GO:0065009', 'GO:0060485',
'GO:0071216', 'GO:0004519', 'GO:0030239', 'GO:0051668', 'GO:0015291',
'GO:0044248', 'GO:0043086', 'GO:0043436', 'GO:0007610', 'GO:0016878',
'GO:0033554', 'GO:0015631', 'GO:0010720', 'GO:0034655', 'GO:0051241',
'GO:0009410', 'GO:0005789', 'GO:0001525', 'GO:0043087', 'GO:0005815',
'GO:0048609', 'GO:0016772', 'GO:1901215', 'GO:0051130', 'GO:0032990',
'GO:0009887', 'GO:0001503', 'GO:0007015', 'GO:0009893', 'GO:0050832',
'GO:0000302', 'GO:0008047', 'GO:0004725', 'GO:0009308', 'GO:0018205',
'GO:0009952', 'GO:0023051', 'GO:0032555', 'GO:0005085', 'GO:0003924',
'GO:0043412', 'GO:0030447', 'GO:0004540', 'GO:0070848', 'GO:0030414',
'GO:0051179', 'GO:0016054', 'GO:0005976', 'GO:0001216', 'GO:0005539',
'GO:0031099', 'GO:0010558', 'GO:0003712', 'GO:0061053', 'GO:0004722',
'GO:0005813', 'GO:0045137', 'GO:1901981', 'GO:0017022', 'GO:0045814',
'GO:0031967', 'GO:0000981', 'GO:0030182', 'GO:0016491', 'GO:0062012',
'GO:0006644', 'GO:0097447', 'GO:0010631', 'GO:0051703', 'GO:0006631',
'GO:0016043', 'GO:0050878', 'GO:0031406', 'GO:0000325', 'GO:0022835',
'GO:0006997', 'GO:0042742', 'GO:0048646', 'GO:0001817', 'GO:1902531',
'GO:0015267', 'GO:0015031', 'GO:1903706', 'GO:0006109', 'GO:0007018',
'GO:0021700', 'GO:0045177', 'GO:0019866', 'GO:0032496', 'GO:0052547',
'GO:0042995', 'GO:0080134', 'GO:0065003', 'GO:0009607', 'GO:0050877',
'GO:0030036', 'GO:0060828', 'GO:0060537', 'GO:0006955', 'GO:0019220',
'GO:0031401', 'GO:0051540', 'GO:0009987', 'GO:0045595', 'GO:0060341',
'GO:0008080', 'GO:0003727', 'GO:0006897', 'GO:0048598', 'GO:0001217',
'GO:0045930', 'GO:0075136', 'GO:0009737', 'GO:0043021', 'GO:0005179',
'GO:0006403', 'GO:0048731', 'GO:0009055', 'GO:0043269', 'GO:0045786',
'GO:0051056', 'GO:0061629', 'GO:0031324', 'GO:0140513', 'GO:0001653',
'GO:0043413', 'GO:0043231', 'GO:0045596', 'GO:0033043', 'GO:0036293',
'GO:0032991', 'GO:0043228', 'GO:0048364', 'GO:0043233', 'GO:0015103',
'GO:0007476', 'GO:0006633', 'GO:0016831', 'GO:0006753', 'GO:0012505',
'GO:0045597', 'GO:0043230', 'GO:0008285', 'GO:0050778', 'GO:0043170',
'GO:0008237', 'GO:0008094', 'GO:0034645', 'GO:0016838', 'GO:0035091',
'GO:0004536', 'GO:0032838', 'GO:1901653', 'GO:0002252', 'GO:0140677',
'GO:0010942', 'GO:0010721', 'GO:0031667', 'GO:0044550', 'GO:0030246',
'GO:0090276', 'GO:0015036', 'GO:0016651', 'GO:0008324', 'GO:0012506',
'GO:0044089', 'GO:0009986', 'GO:0060589', 'GO:0031328', 'GO:0009855',
'GO:0015294', 'GO:0046890', 'GO:1901988', 'GO:0048523', 'GO:0031968',
'GO:0050863', 'GO:0099120', 'GO:0043025', 'GO:0005102', 'GO:0045787',
'GO:0009896', 'GO:0044270', 'GO:0061458', 'GO:0007162', 'GO:1990234',
'GO:0016860', 'GO:0005249', 'GO:0031347', 'GO:0003008', 'GO:0009725',
'GO:0040012', 'GO:0007165', 'GO:0035220', 'GO:0051235', 'GO:0046907',
'GO:0042578', 'GO:0002165', 'GO:0048588', 'GO:0048872', 'GO:0005764',
'GO:0009259', 'GO:0048514', 'GO:0030705', 'GO:0019222', 'GO:0001568',
'GO:0090132', 'GO:0010556', 'GO:0009605', 'GO:0005496', 'GO:0035556',
'GO:0007605', 'GO:0007283', 'GO:0051129', 'GO:0070647', 'GO:0030295',
'GO:0051094', 'GO:0030900', 'GO:0015318', 'GO:0009416', 'GO:0031323',
'GO:1903131', 'GO:0051246', 'GO:0010647', 'GO:0071495', 'GO:0007286',
'GO:0044003', 'GO:0048513', 'GO:0030695', 'GO:0048863', 'GO:0035114',

'GO:0048705', 'GO:0080135', 'GO:0052548', 'GO:0060429', 'GO:0016830',
'GO:0045471', 'GO:0004175', 'GO:0048568', 'GO:0046474', 'GO:0009953',
'GO:0016311', 'GO:0009617', 'GO:0010639', 'GO:0006954', 'GO:0046394',
'GO:0016853', 'GO:1901342', 'GO:0016310', 'GO:0007033', 'GO:0045182',
'GO:0003682', 'GO:0006720', 'GO:0070062', 'GO:0046395', 'GO:0031330',
'GO:0007548', 'GO:0016798', 'GO:0006950', 'GO:0006357', 'GO:0030003',
'GO:0051961', 'GO:0004252', 'GO:0008284', 'GO:0030139', 'GO:0051641',
'GO:0000904', 'GO:0048565', 'GO:0015081', 'GO:0016788', 'GO:0007267',
'GO:0005507', 'GO:0016620', 'GO:0044281', 'GO:0008194', 'GO:0060255',
'GO:0042054', 'GO:0002164', 'GO:0003006', 'GO:0016192', 'GO:0016740',
'GO:0099094', 'GO:1903507', 'GO:0051656', 'GO:0046530', 'GO:0004386',
'GO:0007268', 'GO:0031012', 'GO:0051234', 'GO:0061919', 'GO:0019439',
'GO:0031974', 'GO:1990778', 'GO:0033993', 'GO:0016462', 'GO:0044389',
'GO:0005794', 'GO:0019058', 'GO:0071704', 'GO:0001894', 'GO:0022834',
'GO:0006952', 'GO:0005576', 'GO:0051321', 'GO:0031625', 'GO:0022839',
'GO:0018193', 'GO:0000226', 'GO:0042594', 'GO:0009895', 'GO:0000377',
'GO:0007059', 'GO:0007444', 'GO:0030424', 'GO:0009611', 'GO:0016410',
'GO:0006629', 'GO:1901987', 'GO:0048749', 'GO:0030334', 'GO:0006355',
'GO:0001664', 'GO:0051174', 'GO:0072657', 'GO:0019783', 'GO:0046872',
'GO:0035251', 'GO:0006417', 'GO:0050790', 'GO:0016836', 'GO:0090596',
'GO:1990904', 'GO:0052689', 'GO:0050776', 'GO:2000147', 'GO:0016887',
'GO:0010035', 'GO:0048193', 'GO:0009791', 'GO:0006281', 'GO:0022836',
'GO:0042626', 'GO:0006796', 'GO:0005635', 'GO:0010941', 'GO:0090087',
'GO:0023056', 'GO:0005694', 'GO:0051493', 'GO:0090130', 'GO:0008299',
'GO:0042221', 'GO:0050793', 'GO:0003729', 'GO:0019752', 'GO:1901576',
'GO:0009615', 'GO:1901362', 'GO:0031966', 'GO:0007584', 'GO:0040017',
'GO:0065007', 'GO:0048584', 'GO:2000146', 'GO:0019216', 'GO:0045132',
'GO:0009582', 'GO:1904888', 'GO:0019725', 'GO:0051259', 'GO:0015293',
'GO:0046486', 'GO:0031647', 'GO:0045017', 'GO:1901652', 'GO:0051020',
'GO:0022898', 'GO:0006793', 'GO:0009966', 'GO:0009526', 'GO:0051239',
'GO:0051607', 'GO:1901135', 'GO:0043161', 'GO:0140694', 'GO:0090558',
'GO:0043408', 'GO:0099536', 'GO:0042579', 'GO:0046700', 'GO:0008289',
'GO:0031349', 'GO:0006364', 'GO:0051216', 'GO:0007155', 'GO:0010970',
'GO:0022607', 'GO:0050839', 'GO:0062197', 'GO:0048519', 'GO:0097159',
'GO:0099568', 'GO:0098687', 'GO:0007399', 'GO:0016791', 'GO:0001501',
'GO:0010154', 'GO:0010033', 'GO:0030029', 'GO:0098662', 'GO:0098590',
'GO:0010975', 'GO:0044877', 'GO:0010563', 'GO:0048569', 'GO:0002521',
'GO:0051345', 'GO:0042175', 'GO:0016877', 'GO:0044272', 'GO:0043067',
'GO:0061024', 'GO:0001933', 'GO:0009314', 'GO:0099106', 'GO:0005996',
'GO:0042176', 'GO:0051222', 'GO:0022857', 'GO:0007552', 'GO:0007600',
'GO:1902495', 'GO:0098798', 'GO:0016903', 'GO:0031984', 'GO:0004674',
'GO:0055085', 'GO:0031325', 'GO:0000375', 'GO:0032956', 'GO:0001228',
'GO:0048732', 'GO:0060562', 'GO:0004713', 'GO:0009994', 'GO:0052173',
'GO:0048367', 'GO:0044265', 'GO:0007346', 'GO:0042803', 'GO:0034599',
'GO:0048707', 'GO:0006974', 'GO:0008038', 'GO:0005773', 'GO:1901214',
'GO:0061695', 'GO:0048583', 'GO:0031326', 'GO:0032553', 'GO:0014033',
'GO:0008406', 'GO:0031090', 'GO:0000910', 'GO:0045944', 'GO:0010629',
'GO:0016331', 'GO:0019899', 'GO:0008544', 'GO:0007507', 'GO:0009414',
'GO:0001221', 'GO:0009415', 'GO:0099111', 'GO:0009408', 'GO:0009581',
'GO:0070201', 'GO:0008233', 'GO:0061008', 'GO:0006915', 'GO:0016301',

'GO:0005938', 'GO:0043168', 'GO:0009612', 'GO:0065008', 'GO:0051606',
'GO:0008238', 'GO:0051093', 'GO:0070161', 'GO:0005975', 'GO:0032103',
'GO:0061013', 'GO:0007389', 'GO:0030141', 'GO:0006518', 'GO:0009968',
'GO:0051054', 'GO:1901565', 'GO:0009101', 'GO:0035107', 'GO:0003735',
'GO:0032412', 'GO:0034248', 'GO:0008757', 'GO:0042692', 'GO:2000241',
'GO:0001822', 'GO:0042562', 'GO:0045937', 'GO:0044237', 'GO:0003676',
'GO:0015276', 'GO:0006811', 'GO:0007127', 'GO:0032182', 'GO:0008152',
'GO:0002009', 'GO:0016616', 'GO:0009411', 'GO:0007368', 'GO:0051603',
'GO:0019199', 'GO:0034097', 'GO:0001101', 'GO:0055123', 'GO:0032409',
'GO:0055044', 'GO:0043254', 'GO:0008219', 'GO:0071407', 'GO:0051253',
'GO:0002696', 'GO:0043085', 'GO:0004521', 'GO:0042327', 'GO:0071496',
'GO:0050865', 'GO:0051338', 'GO:0008081', 'GO:0031331', 'GO:1903509',
'GO:0062023', 'GO:0008374', 'GO:0017111', 'GO:0035295', 'GO:0007049',
'GO:0032870', 'GO:0099080', 'GO:0043167', 'GO:0019207', 'GO:0031490',
'GO:0006399', 'GO:0005178', 'GO:0000398', 'GO:1903530', 'GO:0110165',
'GO:0008361', 'GO:0005768', 'GO:0052200', 'GO:0010333', 'GO:0009636',
'GO:0004866', 'GO:0050727', 'GO:0032446', 'GO:0050795', 'GO:1901566',
'GO:0050867', 'GO:0050808', 'GO:0007292', 'GO:0000271', 'GO:0051047',
'GO:0044238', 'GO:0006650', 'GO:0043414', 'GO:0005829', 'GO:0046983',
'GO:0006066', 'GO:1905114', 'GO:1901265', 'GO:0048871', 'GO:0043009',
'GO:0043207', 'GO:0008340', 'GO:0048839', 'GO:0034470', 'GO:0008270',
'GO:0140546', 'GO:0044403', 'GO:0007166', 'GO:0022626', 'GO:0032535',
'GO:0048285', 'GO:0090079', 'GO:0032970', 'GO:0090287', 'GO:0007420',
'GO:0051098', 'GO:0042981', 'GO:0000987', 'GO:0045893', 'GO:1901681',
'GO:0046914', 'GO:0007417', 'GO:0048563', 'GO:0070482', 'GO:0009536',
'GO:0000070', 'GO:1902903', 'GO:0016765', 'GO:0045184', 'GO:0055001',
'GO:0009886', 'GO:0003674', 'GO:0003690', 'GO:0065010', 'GO:0003774',
'GO:0040029', 'GO:0031975', 'GO:0051252', 'GO:0051247', 'GO:0019902',
'GO:0048469', 'GO:0048468', 'GO:0007409', 'GO:0010001', 'GO:0032561',
'GO:0035282', 'GO:0071840', 'GO:0050890', 'GO:0044282', 'GO:0050789',
'GO:0044262', 'GO:0048699', 'GO:1901654', 'GO:0051052', 'GO:0043169',
'GO:0016298', 'GO:0017148', 'GO:0036294', 'GO:0016050', 'GO:0048593',
'GO:1903047', 'GO:0140535', 'GO:0120031', 'GO:0014070', 'GO:0007154',
'GO:0015849', 'GO:0010467', 'GO:0009891', 'GO:0045785', 'GO:0015171',
'GO:0042393', 'GO:0006996', 'GO:0051082', 'GO:0051173', 'GO:0006338',
'GO:0097014', 'GO:0004867', 'GO:0016667', 'GO:0009743', 'GO:0005515',
'GO:0007423', 'GO:0000902', 'GO:0007568', 'GO:0006909', 'GO:0098772',
'GO:0031400', 'GO:0070085', 'GO:0071345', 'GO:0050794', 'GO:0022412',
'GO:0030010', 'GO:0016879', 'GO:0060271', 'GO:0045860', 'GO:1901605',
'GO:0015144', 'GO:0005254', 'GO:0042391', 'GO:0010604', 'GO:0019838',
'GO:0001701', 'GO:0038023', 'GO:0005506', 'GO:0010959', 'GO:0051223',
'GO:0016705', 'GO:0046906', 'GO:0045862', 'GO:0061640', 'GO:0048608',
'GO:0019538', 'GO:0009719', 'GO:0072001', 'GO:0048729', 'GO:0016051',
'GO:0015079', 'GO:0048515', 'GO:1902074', 'GO:0045229', 'GO:0022008',
'GO:0048738', 'GO:0006351', 'GO:0043565', 'GO:0003697', 'GO:0097435',
'GO:0003007', 'GO:0044182', 'GO:0002684', 'GO:0043043', 'GO:0051716',
'GO:0051276', 'GO:0051172', 'GO:0003823', 'GO:0016829', 'GO:0003824',
'GO:0140101', 'GO:0034641', 'GO:0050801', 'GO:0002699', 'GO:0051254',
'GO:0080090', 'GO:0006325', 'GO:0007186', 'GO:0098552', 'GO:0120035',
'GO:0022804', 'GO:0006873', 'GO:0030546', 'GO:0002237', 'GO:0022890',

'GO:0050896', 'GO:0016818', 'GO:0009894', 'GO:0016645', 'GO:0006396',
'GO:0006979', 'GO:0042127', 'GO:0097367', 'GO:0031267', 'GO:0019219',
'GO:0140678', 'GO:0034330', 'GO:0009826', 'GO:0000122', 'GO:0005740',
'GO:0016614', 'GO:0005516', 'GO:0060284', 'GO:0006511', 'GO:0031981',
'GO:0044283', 'GO:2000026', 'GO:0019637', 'GO:0002020', 'GO:0015108',
'GO:0019867', 'GO:0008016', 'GO:0001666', 'GO:0006260', 'GO:0001754',
'GO:0051962', 'GO:0016866', 'GO:0051090', 'GO:0043005', 'GO:0009451',
'GO:0055082', 'GO:0032886', 'GO:0019941', 'GO:0005126', 'GO:0006790',
'GO:0007623', 'GO:0004518', 'GO:0016570', 'GO:0033293', 'GO:0051640',
'GO:0005783', 'GO:0048511', 'GO:0045765', 'GO:1901360', 'GO:0016477',
'GO:0046903', 'GO:0006913', 'GO:0010927', 'GO:0016358', 'GO:0030594',
'GO:0140640', 'GO:0061982', 'GO:0005524', 'GO:1990837', 'GO:0006302',
'GO:0044092', 'GO:0048666', 'GO:0022824', 'GO:0060541', 'GO:0010243',
'GO:0051248', 'GO:0060089', 'GO:0051726', 'GO:0006310', 'GO:0032259',
'GO:0035120', 'GO:0016049', 'GO:0022843', 'GO:0045321', 'GO:0008134',
'GO:0042330', 'GO:0035148', 'GO:0022803', 'GO:1990351', 'GO:0022622',
'GO:0140375', 'GO:0005509', 'GO:0043227', 'GO:0048562', 'GO:0006869',
'GO:0009967', 'GO:0016741', 'GO:0002376', 'GO:0071702', 'GO:0034249',
'GO:0007169', 'GO:0099512', 'GO:0050807', 'GO:0009799', 'GO:0031016',
'GO:0048316', 'GO:0022407', 'GO:0002831', 'GO:0040007', 'GO:0009880',
'GO:0033500', 'GO:0008610', 'GO:0033365', 'GO:0010817', 'GO:0016746',
'GO:0070727', 'GO:0051240', 'GO:2000243', 'GO:0006886', 'GO:0016072',
'GO:0110053', 'GO:0046943', 'GO:0002833', 'GO:1903311', 'GO:0000166',
'GO:0016604', 'GO:0048232', 'GO:0016709', 'GO:0016197', 'GO:0005125',
'GO:0023057', 'GO:0010648', 'GO:0008283', 'GO:0003013', 'GO:0048667',
'GO:2000145', 'GO:0010564', 'GO:0048589', 'GO:0015931', 'GO:0032386',
'GO:0015085', 'GO:0072359', 'GO:0001775', 'GO:0097485', 'GO:0019901',
'GO:0010638', 'GO:0008104', 'GO:0048518', 'GO:0016627', 'GO:1902494',
'GO:0022402', 'GO:0005319', 'GO:0044255', 'GO:0140352', 'GO:0050767',
'GO:0003723', 'GO:0030335', 'GO:0045927', 'GO:0061135', 'GO:0032502',
'GO:0031982', 'GO:0031329', 'GO:0003677', 'GO:0060560', 'GO:0001227',
'GO:0042440', 'GO:0000819', 'GO:0001745', 'GO:0019827', 'GO:0043657',
'GO:1903506', 'GO:0015297', 'GO:0006164', 'GO:0016485', 'GO:0140098',
'GO:0018130', 'GO:0005743', 'GO:0009150', 'GO:0030054', 'GO:1903046',
'GO:0005737', 'GO:0006643', 'GO:0098794', 'GO:1901363', 'GO:0061630',
'GO:0098655', 'GO:0071453', 'GO:0030674', 'GO:0098542', 'GO:0019209',
'GO:0034637', 'GO:0016247', 'GO:0050829', 'GO:0140097', 'GO:1904062',
'GO:0051169', 'GO:0044325', 'GO:0051128', 'GO:0030099', 'GO:0009620',
'GO:0008236', 'GO:0006508', 'GO:0007167', 'GO:0016032', 'GO:0002694',
'GO:0007281', 'GO:0001889', 'GO:0104004', 'GO:0005198', 'GO:0046942',
'GO:0019438', 'GO:0031344', 'GO:0019953', 'GO:0098771', 'GO:0030162',
'GO:0044706', 'GO:0003714', 'GO:0045333', 'GO:2000112', 'GO:0008015',
'GO:0042254', 'GO:0043062', 'GO:0042445', 'GO:0060090', 'GO:0070013',
'GO:1901293', 'GO:0001819', 'GO:0030145', 'GO:0031399', 'GO:0043434',
'GO:0023052', 'GO:0016053', 'GO:0007498', 'GO:0005618', 'GO:1901137',
'GO:0048858', 'GO:0071310', 'GO:1901700', 'GO:0044391', 'GO:0009892',
'GO:0008528', 'GO:0016071', 'GO:0098791', 'GO:0005261', 'GO:0045087',
'GO:0030554', 'GO:0021537', 'GO:0004842', 'GO:0006810', 'GO:0009792',
'GO:0071363', 'GO:0061448', 'GO:0005654', 'GO:0006812', 'GO:0043232',
'GO:0071417', 'GO:0046883', 'GO:0019001', 'GO:0046661', 'GO:0007275',

'GO:0030902', 'GO:0150063', 'GO:0060249', 'GO:0048878', 'GO:0018995',
'GO:0010038', 'GO:0016810', 'GO:0061134', 'GO:0000280', 'GO:0006366',
'GO:0098797', 'GO:0017171', 'GO:1902533', 'GO:0140013', 'GO:0046165',
'GO:0071396', 'GO:0007010', 'GO:0140104', 'GO:0001763', 'GO:0009653',
'GO:0006807', 'GO:0046527', 'GO:0072330', 'GO:0090066', 'GO:0016787',
'GO:0042060', 'GO:0140110', 'GO:2001233', 'GO:0051051', 'GO:0008017',
'GO:0098793', 'GO:0099503', 'GO:0005262', 'GO:0001505', 'GO:0007613',
'GO:0004857', 'GO:0030016', 'GO:0001654', 'GO:0048870', 'GO:0045926',
'GO:1902075', 'GO:0009506', 'GO:0090407', 'GO:0032102', 'GO:1904951',
'GO:1902850', 'GO:0043604', 'GO:0007411', 'GO:0060041', 'GO:1902679',
'GO:0140014', 'GO:0098978', 'GO:0043068', 'GO:0043583', 'GO:0060548',
'GO:0009059', 'GO:0001932', 'GO:0001944', 'GO:0043473', 'GO:0007005',
'GO:0004721', 'GO:0006935', 'GO:0006970', 'GO:0061564', 'GO:0034765',
'GO:0016684', 'GO:1903561', 'GO:0042546', 'GO:0043549', 'GO:0032101',
'GO:0030001', 'GO:0005856', 'GO:0006725', 'GO:0019842', 'GO:0051963',
'GO:0004197', 'GO:0004553', 'GO:0010608', 'GO:0052572', 'GO:0042593',
'GO:0055002', 'GO:0045934', 'GO:0045089', 'GO:0033044', 'GO:0005759',
'GO:0015370', 'GO:0030587', 'GO:0006412', 'GO:0008652', 'GO:0005525',
'GO:0009409', 'GO:0008202', 'GO:0019748', 'GO:0005231', 'GO:0008276',
'GO:0008135', 'GO:0043410', 'GO:0004497', 'GO:0045296', 'GO:0071900',
'GO:0004527', 'GO:0050708', 'GO:0043010', 'GO:0009057', 'GO:0031346',
'GO:0016817', 'GO:0048638', 'GO:0046467', 'GO:0046982', 'GO:0016779',
'GO:0008514', 'GO:0005488', 'GO:0009100', 'GO:0043632', 'GO:0005244',
'GO:0005840', 'GO:0022832', 'GO:0051301', 'GO:0042277', 'GO:0007472',
'GO:0046873', 'GO:0002683', 'GO:0043603', 'GO:0016052', 'GO:0003779',
'GO:0035050', 'GO:0009507', 'GO:0020037', 'GO:1901505', 'GO:0010876',
'GO:0043177', 'GO:0043292', 'GO:0030154', 'GO:0004888', 'GO:0010468',
'GO:0051960', 'GO:0071375', 'GO:0044260', 'GO:1901701', 'GO:0005216',
'GO:0019693', 'GO:0044085', 'GO:0036211', 'GO:0044093', 'GO:0033036',
'GO:1901564', 'GO:0055086', 'GO:0045936', 'GO:1901888', 'GO:0071695',
'GO:0000978', 'GO:0030097', 'GO:0032989', 'GO:0045202', 'GO:0008201',
'GO:0030031', 'GO:0019955', 'GO:0098588', 'GO:0097708', 'GO:0043226',
'GO:0004620', 'GO:0048580', 'GO:0055074', 'GO:0098727', 'GO:0048762',
'GO:0044297', 'GO:0001708', 'GO:0050920', 'GO:0007369', 'GO:0072522',
'GO:0015711', 'GO:0007424', 'GO:0007611', 'GO:0043523', 'GO:0048869',
'GO:0022414', 'GO:0031348', 'GO:0001067', 'GO:0060322', 'GO:0072521',
'GO:0006163', 'GO:0099081', 'GO:0097659', 'GO:0004601', 'GO:0036094',
'GO:0060972', 'GO:0016782', 'GO:0012501', 'GO:0072594', 'GO:0040008',
'GO:2001141', 'GO:0061659', 'GO:0010557', 'GO:0032787', 'GO:0030111',
'GO:0002791', 'GO:0005096', 'GO:0045088', 'GO:0045935', 'GO:0031669',
'GO:0006397', 'GO:0006520', 'GO:0003012', 'GO:0005774', 'GO:0022613',
'GO:0098827', 'GO:0016407', 'GO:0048856', 'GO:0099177', 'GO:0005929',
'GO:0032940', 'GO:0031410', 'GO:0030030', 'GO:0044271', 'GO:0010506',
'GO:0000003', 'GO:0009889', 'GO:0044249', 'GO:0006259', 'GO:0016706',
'GO:0048704', 'GO:0032504', 'GO:0051707', 'GO:1902532', 'GO:0005730',
'GO:0001558', 'GO:0010948', 'GO:0005739', 'GO:0015399', 'GO:0003002',
'GO:1902936', 'GO:0008033', 'GO:0071456', 'GO:0090068', 'GO:0022604',
'GO:1903508', 'GO:0008037', 'GO:1901702', 'GO:0070887', 'GO:0046649',
'GO:0016875', 'GO:0009888', 'GO:0043229', 'GO:0032774', 'GO:0043065',
'GO:0003730', 'GO:1902680', 'GO:0002064', 'GO:0051649', 'GO:0006261'}

The Function labels all selected Annotations

```
def assign_labels(annotations, selected_terms=selected_terms):

    intersection = selected_terms.intersection(annotations)
    labels = np.isin(np.array(list(selected_terms)),
np.array(list(intersection)))

    return list(labels.astype('int'))

annotations = train_terms.groupby('EntryID')['term'].apply(set)
labels = annotations.progress_apply(assign_labels)

labels.head()
```

```
100%|██████████| 142246/142246 [03:11<00:00, 740.89it/s]

EntryID
A0A009IHW8    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
A0A021WW32    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
A0A021WZA4    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
A0A023FBW4    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
A0A023FBW7    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
Name: term, dtype: object
```

Spliting embeds and ids into x and y training sets

```
train_ids = np.load('/kaggle/input/t5embeds/train_ids.npy')

x_train = np.load('/kaggle/input/t5embeds/train_embeds.npy')
y_train = np.array(labels[train_ids].to_list())

x_train, x_valid, y_train, y_valid = train_test_split(x_train,
y_train, shuffle=True, random_state=42)

nfeats=x_train.shape[1]
nlabels=y_train.shape[1]
```

A Neural Network with 2 hidden layers

```
model=Sequential()
model.add(Dense(256,activation='relu',input_dim=nfeats))
model.add(Dense(128,activation='swish'))
model.add(Dense(128,activation='swish'))
model.add(Dense(nlabels,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics='acc
uracy')
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
```

```
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_6 (Dense)              (None, 256)               262400

 dense_7 (Dense)              (None, 128)               32896

 dense_8 (Dense)              (None, 128)               16512

 dense_9 (Dense)              (None, 1500)              193500

=================================================================
Total params: 505,308
Trainable params: 505,308
Non-trainable params: 0
_________________________________________________________________

model.fit(x_train, y_train, epochs=15, batch_size=128,
validation_data=(x_valid, y_valid))

Epoch 1/15
834/834 [==============================] - 21s 18ms/step - loss:
0.0863 - accuracy: 0.1598 - val_loss: 0.0678 - val_accuracy: 0.1223
Epoch 2/15
834/834 [==============================] - 14s 17ms/step - loss:
0.0656 - accuracy: 0.1475 - val_loss: 0.0641 - val_accuracy: 0.1547
Epoch 3/15
834/834 [==============================] - 14s 17ms/step - loss:
0.0629 - accuracy: 0.1490 - val_loss: 0.0622 - val_accuracy: 0.1699
Epoch 4/15
834/834 [==============================] - 14s 17ms/step - loss:
0.0613 - accuracy: 0.1621 - val_loss: 0.0612 - val_accuracy: 0.1846
Epoch 5/15
834/834 [==============================] - 14s 17ms/step - loss:
0.0602 - accuracy: 0.1657 - val_loss: 0.0604 - val_accuracy: 0.1582
Epoch 6/15
834/834 [==============================] - 15s 18ms/step - loss:
0.0593 - accuracy: 0.1700 - val_loss: 0.0599 - val_accuracy: 0.1795
Epoch 7/15
834/834 [==============================] - 14s 17ms/step - loss:
0.0585 - accuracy: 0.1701 - val_loss: 0.0591 - val_accuracy: 0.1732
Epoch 8/15
834/834 [==============================] - 14s 17ms/step - loss:
0.0577 - accuracy: 0.1698 - val_loss: 0.0588 - val_accuracy: 0.1511
Epoch 9/15
834/834 [==============================] - 14s 17ms/step - loss:
0.0571 - accuracy: 0.1691 - val_loss: 0.0584 - val_accuracy: 0.1527
Epoch 10/15
834/834 [==============================] - 15s 18ms/step - loss:
0.0565 - accuracy: 0.1675 - val_loss: 0.0582 - val_accuracy: 0.1706
Epoch 11/15
834/834 [==============================] - 15s 17ms/step - loss:
```

```
0.0559 - accuracy: 0.1662 - val_loss: 0.0581 - val_accuracy: 0.1637
Epoch 12/15
834/834 [==============================] - 15s 18ms/step - loss:
0.0554 - accuracy: 0.1653 - val_loss: 0.0579 - val_accuracy: 0.1633
Epoch 13/15
834/834 [==============================] - 15s 17ms/step - loss:
0.0550 - accuracy: 0.1655 - val_loss: 0.0578 - val_accuracy: 0.1378
Epoch 14/15
834/834 [==============================] - 16s 19ms/step - loss:
0.0545 - accuracy: 0.1646 - val_loss: 0.0577 - val_accuracy: 0.1489
Epoch 15/15
834/834 [==============================] - 16s 20ms/step - loss:
0.0540 - accuracy: 0.1643 - val_loss: 0.0576 - val_accuracy: 0.1603

<keras.callbacks.History at 0x7f69d2149960>
```

```
y_hat = model.predict(x_valid)

scores = pd.DataFrame(columns=list(selected_terms), index=['roc_auc'])

for i, term in enumerate(selected_terms):
    score = roc_auc_score(y_valid[:, i], y_hat[:, i])
    scores[term] = score

scores.mean(axis=1)
```

```
1112/1112 [==============================] - 4s 3ms/step

roc_auc    0.893654
dtype: float64
```

```
test_ids = np.load('/kaggle/input/t5embeds/test_ids.npy')
x_test = np.load('/kaggle/input/t5embeds/test_embeds.npy')

del x_train, y_train, x_valid, y_valid, labels
gc.collect()
```

```
---------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[26], line 1
----> 1 del x_train, y_train, x_valid, y_valid, labels
      2 gc.collect()

NameError: name 'x_train' is not defined
```

The Final Prediction Dataset

```
predictions = model.predict(x_test)
del x_test
```

```
gc.collect()

chunk_size = 5_000
chunks = [range(i, min(i + chunk_size, len(predictions))) for i in
range(0, len(predictions), chunk_size)]

final_sub = pd.DataFrame()  # Create an empty DataFrame to hold the
final result

print(f"processing {len(chunks)} chunks of {chunk_size} predictions
each")

for chunk in chunks:
    print(f"processing chunk {chunk}")
    sub = pd.DataFrame(data=predictions[chunk],
columns=list(selected_terms), index=test_ids[chunk])
    sub = sub.T.unstack().reset_index(name='prediction')
    sub = sub.loc[sub['prediction'] > 0]
    final_sub = pd.concat([final_sub, sub])  # Concatenate current
chunk DataFrame to the final DataFrame

final_sub.head()
```

```
4434/4434 [==============================] - 15s 3ms/step
processing 29 chunks of 5000 predictions each
processing chunk range(0, 5000)
processing chunk range(5000, 10000)
processing chunk range(10000, 15000)
processing chunk range(15000, 20000)
processing chunk range(20000, 25000)
processing chunk range(25000, 30000)
processing chunk range(30000, 35000)
processing chunk range(35000, 40000)
processing chunk range(40000, 45000)
processing chunk range(45000, 50000)
processing chunk range(50000, 55000)
processing chunk range(55000, 60000)
processing chunk range(60000, 65000)
processing chunk range(65000, 70000)
processing chunk range(70000, 75000)
processing chunk range(75000, 80000)
processing chunk range(80000, 85000)
processing chunk range(85000, 90000)
processing chunk range(90000, 95000)
processing chunk range(95000, 100000)
processing chunk range(100000, 105000)
processing chunk range(105000, 110000)
processing chunk range(110000, 115000)
processing chunk range(115000, 120000)
processing chunk range(120000, 125000)
```

```
processing chunk range(125000, 130000)
processing chunk range(130000, 135000)
processing chunk range(135000, 140000)
processing chunk range(140000, 141865)
```

	level_0	level_1	prediction
0	Q9CQV8	GO:0046915	0.000025
1	Q9CQV8	GO:0098916	0.005075
2	Q9CQV8	GO:0005615	0.104366
3	Q9CQV8	GO:0048585	0.045323
4	Q9CQV8	GO:0016236	0.008446