STRIDE CLASSIFICATION DATASET, FEATURES, AND MODEL GENERATION

Installing SensiML

```
!pip install sensiml-dev -U
```

Login into the Project

```
import pandas as pd
from sensiml import SensiML
dsk = SensiML()
```

```
    /usr/local/lib/python3.7/dist-packages/sensiml/client.py:112: UserWarning: Confi
      mgc("%config Completer.use_jedi = False")
```

Sensor Data

```
dsk.project = 'Stride Classification'
dsk.project.columns()
```

```
    dict_keys(['GyroscopeX', 'GyroscopeY', 'GyroscopeZ', 'AccelerometerX', 'Accelero
```

Metadata

```
dsk.project.metadata_columns()
```

```
    ['Cont or Event', 'Type', 'Side', 'Subject', 'capture_uuid', 'segment_uuid']
```

Data Samples

```
dsk.project.get_project_summary().T
```

	0	1	2	3	4	
Capture Name	Rafael_Normal 2021-07-31 16_00_17.csv	Martin_Heel Striking 2021-07-30 08_06_42.csv	Rafael_Pronation 2021-07-31 16_13_34.csv	Rafael_Pronation 2021-07-31 16_14_27.csv	Rafael_Supination 2021-07-30 21_12_49.csv	l 16
Capture UUID	01267146-efa1-4bfb-ad0f-f27d64db23ab	062dddc8-7474-48e0-8b4c-fdece0a6a453	0779527e-a0c8-4c51-8c99-68f837196a70	1629d4c8-b571-4eb9-bb85-f1106668e56a	1b7a563d-311d-4b41-9bdb-e1d28aeda6d5	95
Total Event Count	0	0	0	0	0	
Size (MB)	0.01	0.02	0.02	0.02	0.08	
Cont or Event	Continuous	None	Continuous	Continuous	Discrete Event	
Side	Right	Right	Right	Right	Right	

Project Pipeline

```
dsk.pipeline = 'Pipeline Final'
```

Feature Generator

```
pd.set_option("display.max_rows",150)
dsk.list_functions(qgrid=False).head(100)
```

	NAME	TYPE	SUBTYPE	DESCRIPTION	KP FUNCTION
0	Add Convolve	Augmentation	Supervised	Add Convolve:\n Convolve (smoothing...	False
1	Add Quantize	Augmentation	Supervised	Add Quantize:\n Quantize time serie...	False
2	Add Noise	Augmentation	Supervised	Add Noise:\n Add random noise to ti...	False
3	Add Drift	Augmentation	Supervised	Add Drift:\n The augmenter drifts t...	False
4	Add Dropout	Augmentation	Supervised	Add Dropout:\n Dropout values of so...	False
5	Add Pool	Augmentation	Supervised	Add Pool:\n Reduce the temporal res...	False
6	Add Reverse	Augmentation	Supervised	Add Reverse:\n Reverse the time lin...	False
7	Add TimeWarp	Augmentation	Supervised	Add Timewarp:\n Random time warping...	False
8	PME	Classifier	Clustering	PME or pattern matching engine is a distance b...	False
9	Decision Tree Ensemble	Classifier	Ensemble	The decision tree ensemble classifier is an en...	False
10	Boosted Tree Ensemble	Classifier	Ensemble	The boosted tree ensemble classifier is an ens...	False
11	Bonsai	Classifier	Ensemble	Bonsai is a tree model for supervised learning...	False
				The Tensorflow Micro	

```
dsk.pipeline.add_feature_generator?
```

Pipeline Datasets

```
dsk.list_queries()
```

Name	Created	UUID

```
dsk.pipeline.reset()
dsk.pipeline.set_input_query("Continuous Final")
dsk.pipeline.describe()
```

```
------------------------------------------------------------------------
 0.     Name: Continuous Final                                Type: query
------------------------------------------------------------------------
------------------------------------------------------------------------
```

Absolute area of low

```
print(dsk.snippets.Segmenter.Windowing())
```

```
    dsk.pipeline.add_transform("Windowing", params={"window_size": 250,
                                    "delta": 250,
                                    "train_delta": 0,
                                    "return_segment_index": False,
                                    })
```

23 Absolute Area of High Frequency Feature Generator Area frequency components True

```
dsk.function_description("Windowing")
```

```
        This function transfer the `input_data` and `group_column` from the previou
        It groups 'input_data' by using group_column. It divides each group into wi
        The argument `delta` represents the extent of overlap.

        Args:
            window_size: Size of each window
            delta: The number of samples to increment. It is similar to overlap.
              If delta is equal to window size, this means no overlap.
            train_delta: Train delta will be used only during training. Can be usec
              Only used if train_delta is set to > 0.
            return_segment_index (False): Set to true to see the segment indexes
              for start and end. Note: This should only be used for visualization r
              pipeline building.
        Returns:
            DataFrame: Returns dataframe with `SegmentID` column added to the origi

        Example:
            >>> dsk.pipeline.reset()
            >>> df = dsk.datasets.load_activity_raw_toy()
            >>> df
                out:
                       Subject     Class  Rep  accelx  accely  accelz
                    0      s01  Crawling    1     377     569    4019
                    1      s01  Crawling    1     357     594    4051
                    2      s01  Crawling    1     333     638    4049
                    3      s01  Crawling    1     340     678    4053
                    4      s01  Crawling    1     372     708    4051
                    5      s01  Crawling    1     410     733    4028
                    6      s01  Crawling    1     450     733    3988
                7      s01  Crawling    1     492     696    3947
                8      s01  Crawling    1     518     677    3943
                9      s01  Crawling    1     528     695    3988
                10     s01  Crawling    1      -1    2558    4609
                11     s01   Running    1     -44   -3971     843
                12     s01   Running    1     -47   -3982     836
                13     s01   Running    1     -43   -3973     832
                14     s01   Running    1     -40   -3973     834
                15     s01   Running    1     -48   -3978     844
                16     s01   Running    1     -52   -3993     842
                17     s01   Running    1     -64   -3984     821
                18     s01   Running    1     -64   -3966     813
                19     s01   Running    1     -66   -3971     826
                20     s01   Running    1     -62   -3988     827
                21     s01   Running    1     -57   -3984     843

        >>> dsk.pipeline.set_input_data('test_data', df, force=True,
                            data_columns=['accelx', 'accely', 'accelz'],
                            group_columns=['Subject', 'Class', 'Rep'],
                            label_column='Class')

        >>> dsk.pipeline.add_transform('Windowing',
                                       params={'window_size' : 5,
                                               'delta': 5})

        >>> results, stats = dsk.pipeline.execute()
        >>> print results
            out:
                     Class  Rep  SegmentID  Subject  accelx  accely  accelz
```

Feature Calculate the peak

```
dsk.pipeline.add_transform("Windowing", params={"window_size":250, "delta":100,"train_
dsk.pipeline.describe()
```

```
    ------------------------------------------------------------------------
     0.     Name: Continuous Final                          Type: query
    ------------------------------------------------------------------------
    ------------------------------------------------------------------------
     1.     Name: Windowing                                 Type: segmenter
    ------------------------------------------------------------------------
            group_columns: ['Cont or Event', 'Side', 'Stride', 'Subject', 'Type', 's
            window_size: 250
            delta: 100
            train_delta: 50
            return_segment_index: False
    ------------------------------------------------------------------------
```

47 Area Generator Physical magnitude area.\n \n .. True

Adding Feature Mannually to the Dataset

48 Mean Difference Generator Change difference of each True

```
dsk.pipeline.add_transform("Strip",params={"input_columns":["AccelerometerX","Accelero
                                           "type":"mean",},)
dsk.pipeline.describe()
dsk.pipeline.add_transform("Strip",params={"input_columns":["GyroscopeX","GyroscopeY",
dsk.pipeline.add_transform("Strip",params={"input_columns":["GyroscopeX","GyroscopeY",
                                                 "type":"mean",},)
dsk.pipeline.describe()
```

```
    ------------------------------------------------------------------------
     0.     Name: Continuous Final                              Type: query
    ------------------------------------------------------------------------
    ------------------------------------------------------------------------
     1.     Name: Windowing                                     Type: segmenter
    ------------------------------------------------------------------------
            group_columns: ['Cont or Event', 'Side', 'Stride', 'Subject', 'Type', 's
            window_size: 250
            delta: 100
            train_delta: 50
            return_segment_index: False
    ------------------------------------------------------------------------
     2.     Name: Strip                                         Type: transform
    ------------------------------------------------------------------------
            group_columns: ['Cont or Event', 'SegmentID', 'Side', 'Stride', 'Subject
            input_columns: ['AccelerometerX', 'AccelerometerY', 'AccelerometerZ']
            type: mean
    ------------------------------------------------------------------------


    ------------------------------------------------------------------------
     0.     Name: Continuous Final                              Type: query
    ------------------------------------------------------------------------
    ------------------------------------------------------------------------
     1.     Name: Windowing                                     Type: segmenter
    ------------------------------------------------------------------------
            group_columns: ['Cont or Event', 'Side', 'Stride', 'Subject', 'Type', 's
            window_size: 250
            delta: 100
            train_delta: 50
            return_segment_index: False
    ------------------------------------------------------------------------
     2.     Name: Strip                                         Type: transform
    ------------------------------------------------------------------------
            group_columns: ['Cont or Event', 'SegmentID', 'Side', 'Stride', 'Subject
            input_columns: ['GyroscopeX', 'GyroscopeY', 'GyroscopeZ']
            type: mean
    ------------------------------------------------------------------------
```

Adding Features with Feature Generator (Rate of Change & Statistical)

63 Maximum Feature Generator Statistical of each column in True

```
dsk.pipeline.add_feature_generator?
```

Feature Computes the arithmetic

```
sensor_columns = ['AccelerometerX', 'AccelerometerY', 'AccelerometerZ','GyroscopeX', '
dsk.pipeline.add_feature_generator([
                                    {'subtype_call':'Rate of Change'},
                                    {'subtype_call':'Statistical'},
                                    {"name":"MFCC",
 "params": {
    "columns":sensor_columns,
    "sample_rate":100,
    "cepstra_count":10,
}},
 ],
 function_defaults={'columns':sensor_columns},
)
```

Feature Computes the number of

```
fv, s = dsk.pipeline.execute()
```

```
Executing Pipeline with Steps:

------------------------------------------------------------------------
 0.     Name: Continuous Final                          Type: query
------------------------------------------------------------------------
------------------------------------------------------------------------
 1.     Name: Windowing                                 Type: segmenter
------------------------------------------------------------------------
------------------------------------------------------------------------
 2.     Name: Strip                                     Type: transform
------------------------------------------------------------------------
------------------------------------------------------------------------
 3.     Name: generator_set                             Type: generatorset
------------------------------------------------------------------------
------------------------------------------------------------------------



Results Retrieved... Execution Time: 0 min. 2 sec.
```

Generator column in

Features Added

Generator column i...

```
dsk.pipeline.describe()
```

```
------------------------------------------------------------------------
 0.     Name: Continuous Final                          Type: query
------------------------------------------------------------------------
------------------------------------------------------------------------
 1.     Name: Windowing                                 Type: segmenter
------------------------------------------------------------------------
        group_columns: ['Cont or Event', 'Side', 'Stride', 'Subject', 'Type', '
        window_size: 250
        delta: 100
        train_delta: 50
        return_segment_index: False
------------------------------------------------------------------------
 2.     Name: Strip                                     Type: transform
------------------------------------------------------------------------
        group_columns: ['Cont or Event', 'SegmentID', 'Side', 'Stride', 'Subjec
        input_columns: ['GyroscopeX', 'GyroscopeY', 'GyroscopeZ']
        type: mean
------------------------------------------------------------------------
 3.     Name: generator_set                               Type: generatorset
------------------------------------------------------------------------
         0. Name: MFCC
         1. Name: MFCC
         2. Name: MFCC
         3. Name: MFCC
         4. Name: MFCC
         5. Name: MFCC
         6. Name: Mean Difference
         7. Name: Threshold Crossing Rate
         8. Name: Mean Crossing Rate
         9. Name: Zero Crossing Rate
        10. Name: Sigma Crossing Rate
        11. Name: Second Sigma Crossing Rate
        12. Name: Threshold With Offset Crossing Rate
        13. Name: Kurtosis
        14. Name: Maximum
        15. Name: Absolute Mean
        16. Name: Mean
        17. Name: Variance
        18. Name: Zero Crossings
        19. Name: Positive Zero Crossings
        20. Name: Negative Zero Crossings
        21. Name: Median
        22. Name: Linear Regression Stats
        23. Name: Linear Regression Stats
        24. Name: Linear Regression Stats
        25. Name: Linear Regression Stats
        26. Name: Linear Regression Stats
        27. Name: Linear Regression Stats
        28. Name: Standard Deviation
        29. Name: Skewness
        30. Name: Interquartile Range
        31. Name: 25th Percentile
        32. Name: 75th Percentile
        33. Name: 100th Percentile
        34. Name: Minimum
        35. Name: Sum
        36. Name: Absolute Sum
        group_columns: ['Cont_or_Event', 'SegmentID', 'Side', 'Stride', 'Subje
```

New Shape With The Added Features

```
fv.T.shape
```

```
(241, 67)
```

```
fv.T.head(100)
```

	0	1	2
Cont or Event	Continuous	Continuous	Continuous
SegmentID	0	0	0
Side	Right	Right	Right
Stride	Normal	Normal	Normal
Subject	Rafael	Rafael	Rafael
Type	Train	Train	Train
segment_uuid	0dbde19a-d028-47ae-9642-ef056e4dcdf0	2a28cd32-0193-4b52-a0ed-5e25da32832c	2d4326a4-61cd-415c-9af5-dc27933bbb0a
gen_0001_AccelerometerXmfcc_000000	330571	330408	322229
gen_0001_AccelerometerXmfcc_000001	-96066	-72169	-91373
gen_0001_AccelerometerXmfcc_000002	-90505	-105532	-73317
gen_0001_AccelerometerXmfcc_000003	-34643	-39155	-51241
gen_0001_AccelerometerXmfcc_000004	-9615	-14631	-26385
gen_0001_AccelerometerXmfcc_000005	-56522	-17195	-42642
gen_0001_AccelerometerXmfcc_000006	-32336	-9984	-71405
gen_0001_AccelerometerXmfcc_000007	-28055	10705	-49373
gen_0001_AccelerometerXmfcc_000008	-7462	-8046	7528
gen_0001_AccelerometerXmfcc_000009	-21134	33507	-9233
gen_0002_AccelerometerYmfcc_000000	362054	359008	345784
gen_0002_AccelerometerYmfcc_000001	-60355	-57397	-47997
gen_0002_AccelerometerYmfcc_000002	-38405	-20586	-12613
gen_0002_AccelerometerYmfcc_000003	-23059	-28082	-38464
gen_0002_AccelerometerYmfcc_000004	-7128	-57807	-27268
gen_0002_AccelerometerYmfcc_000005	5684	-13782	-11805
gen_0002_AccelerometerYmfcc_000006	1322	34924	19483
gen_0002_AccelerometerYmfcc_000007	-2923	14919	12632
gen_0002_AccelerometerYmfcc_000008	-23784	-7330	9773
gen_0002_AccelerometerYmfcc_000009	-2254	28701	7625
gen_0003_AccelerometerZmfcc_000000	358888	364146	349770

```
fv.T.tail(100)
```

```
rv.r.tail(100)
```

	0	1	2

Feature Selection (using Variance Threshol, Correlation Threshold & t-Test Feature Selector) and Scaling the output data before training the model.

gen_0084_GyroscopeZZeroCrossings	60	57	52

```
dsk.pipeline.add_feature_selector([{'name':'Variance Threshold','params':{"threshold":
                                   {'name':'Correlation Threshold','params':{"threshol
                                   {'name':'t-Test Feature Selector','params':{"Featur
                                   ])
dsk.pipeline.add_transform(
    "Min Max Scale",)
dsk.pipeline.describe()
```

```
    ------------------------------------------------------------------------
     0.     Name: Continuous Final                              Type: query
    ------------------------------------------------------------------------
    ------------------------------------------------------------------------
     1.     Name: Windowing                                     Type: segmenter
    ------------------------------------------------------------------------
            group_columns: ['Cont or Event', 'Side', 'Stride', 'Subject', 'Type', '
            window_size: 250
            delta: 100
            train_delta: 50
            return_segment_index: False
    ------------------------------------------------------------------------
     2.     Name: Strip                                         Type: transform
    ------------------------------------------------------------------------
            group_columns: ['Cont or Event', 'SegmentID', 'Side', 'Stride', 'Subjec
            input_columns: ['GyroscopeX', 'GyroscopeY', 'GyroscopeZ']
            type: mean
    ------------------------------------------------------------------------
     3.     Name: generator_set                                 Type: generatorset
    ------------------------------------------------------------------------
             0. Name: MFCC
             1. Name: MFCC
             2. Name: MFCC
             3. Name: MFCC
             4. Name: MFCC
             5. Name: MFCC
             6. Name: Mean Difference
             7. Name: Threshold Crossing Rate
             8. Name: Mean Crossing Rate
             9. Name: Zero Crossing Rate
            10. Name: Sigma Crossing Rate
            11. Name: Second Sigma Crossing Rate
            12. Name: Threshold With Offset Crossing Rate
            13. Name: Kurtosis
            14. Name: Maximum
            15. Name: Absolute Mean
            16. Name: Mean
            17. Name: Variance
            18. Name: Zero Crossings
```

```
        19. Name: Positive Zero Crossings
        20. Name: Negative Zero Crossings
        21. Name: Median
        22. Name: Linear Regression Stats
        23. Name: Linear Regression Stats
        24. Name: Linear Regression Stats
        25. Name: Linear Regression Stats
        26. Name: Linear Regression Stats
        27. Name: Linear Regression Stats
        28. Name: Standard Deviation
        29. Name: Skewness
        30. Name: Interquartile Range
        31. Name: 25th Percentile
        32. Name: 75th Percentile
        33. Name: 100th Percentile
        34. Name: Minimum
        35. Name: Sum
        36. Name: Absolute Sum
        group_columns: ['Cont or Event', 'SegmentID', 'Side', 'Stride', 'Subjec
```

gen_0107_GyroscopeYLinearRegressionStdErr_0003 0.376796 0.490591 0.490307

Executing the Pipeline

gen_0108_GyroscopeZLinearRegressionIntercept_0001 115.38 75.7575 14.666

```
fv_t, s_t = dsk.pipeline.execute()
```

```
Executing Pipeline with Steps:

------------------------------------------------------------------------
 0.     Name: Continuous Final                          Type: query
------------------------------------------------------------------------
------------------------------------------------------------------------
 1.     Name: Windowing                                 Type: segmenter
------------------------------------------------------------------------
------------------------------------------------------------------------
 2.     Name: Strip                                     Type: transform
------------------------------------------------------------------------
------------------------------------------------------------------------
 3.     Name: generator_set                             Type: generatorset
------------------------------------------------------------------------
------------------------------------------------------------------------
 4.     Name: selector_set                              Type: selectorset
------------------------------------------------------------------------
------------------------------------------------------------------------
 5.     Name: Min Max Scale                             Type: transform
------------------------------------------------------------------------
------------------------------------------------------------------------



Results Retrieved... Execution Time: 0 min. 0 sec.
```

Significant Features Selected (reduced to a few)

gen_0125_GyroscopeYIQR 414.25 486.25 374.75

```
fv_t.T
```

	0	1	2	
gen_0099_AccelerometerZMedian	199	193	166	19
gen_0102_GyroscopeZMedian	125	156	148	9
gen_0133_AccelerometerX75Percentile	133	104	135	12
gen_0150_GyroscopeZminimum	187	198	243	23
Cont or Event	Continuous	Continuous	Continuous	Continuou
SegmentID	0	0	0	
Side	Right	Right	Right	Rigl
Stride	Normal	Normal	Normal	Norma
Subject	Rafael	Rafael	Rafael	Rafa
Type	Train	Train	Train	Trai
segment_uuid	0dbde19a-d028-47ae-9642-ef056e4dcdf0	2a28cd32-0193-4b52-a0ed-5e25da32832c	2d4326a4-61cd-415c-9af5-dc27933bbb0a	bed1d3fa 096a-4bb2 859 f45c391b311

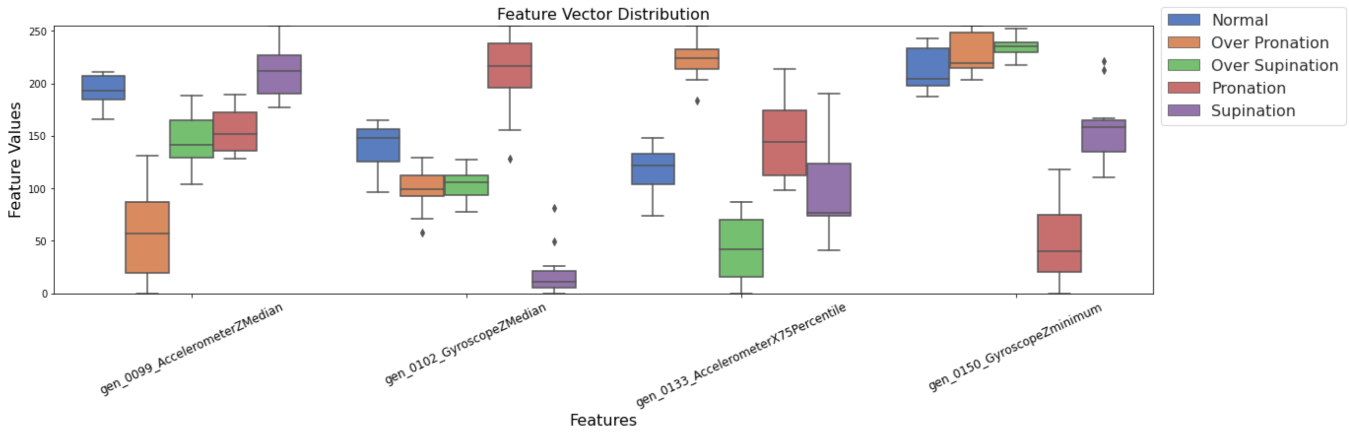
gen_0142_GyroscopeX100Percentile 5600 5032 4806

```
fv_t.T.shape
```

```
(11, 67)
```

gen_0145_AccelerometerXminimum -1165 -802 -620

```
dsk.pipeline.visualize_features(fv_t)
```



Creating Train and Test Datasets

```
x_train, x_test, x_validate, y_train, y_test, y_validate, class_map = dsk.pipeline.fea
```

```
    -----  Summary  -----
    Class Map: {'Normal': 0, 'Over Pronation': 1, 'Over Supination': 2, 'Pronation':
    Train:
     total:  53
     by class: [ 8. 12. 13. 12.  8.]
    Validate:
     total:  14
     by class: [1. 4. 3. 3. 3.]
    Train:
     total:  0
     by class: [0. 0. 0. 0. 0.]
```

```
x_train.shape
```

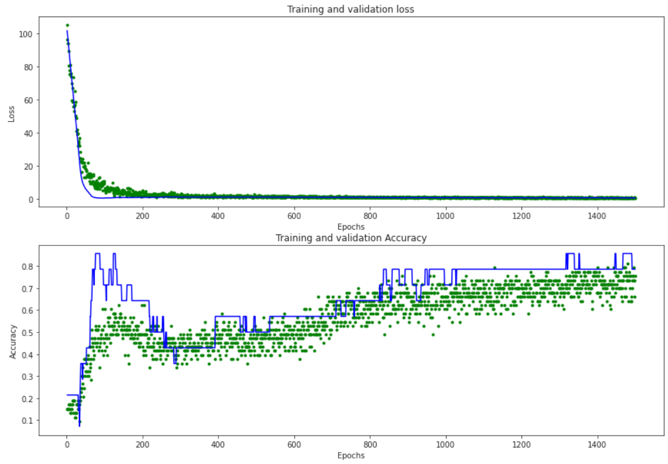
```
    (53, 4)
```

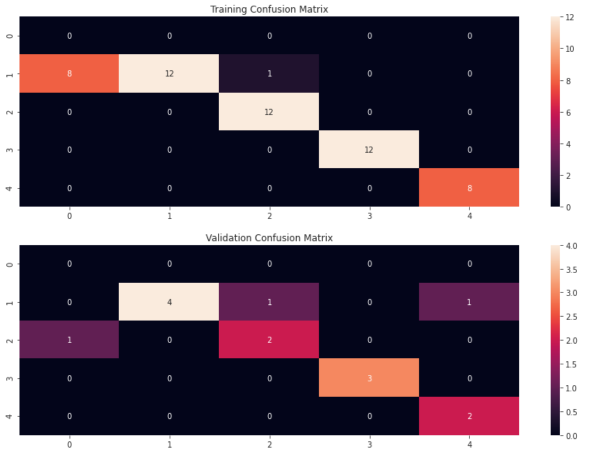
Creating the NN Aerchitecture Model in Tensorflow

```
from tensorflow.keras import layers
import tensorflow as tf

tf_model = tf.keras.Sequential()

tf_model.add(layers.Dense(11, activation='relu',kernel_regularizer='l1',input_shape=(x
tf_model.add(layers.Dropout(0.1))
tf_model.add(layers.Dense(8, activation='relu',input_shape=(x_train.shape[1],)))
tf_model.add(layers.Dropout(0.1))
tf_model.add(layers.Dense(y_train.shape[1], activation='softmax'))

# Fitting the Model
tf_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy

tf_model.summary()
train_history = {'loss':[],'val_loss':[],'accuracy':[],'val_accuracy':[]}
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense (Dense)                (None, 11)                55        
_________________________________________________________________
dropout (Dropout)            (None, 11)                0         
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 96        
_________________________________________________________________
dropout_1 (Dropout)          (None, 8)                 0         
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 45        
=================================================================
Total params: 196
Trainable params: 196
Non-trainable params: 0
_________________________________________________________________
```

```
from IPython.display import clear_output
import sensiml.tensorflow.utils as sml_tf

num_iterations=10
epochs=100
batch_size=32

data = tf.data.Dataset.from_tensor_slices((x_train, y_train))
shuffle_ds = data.shuffle(buffer_size=x_train.shape[0], reshuffle_each_iteration=True)

for i in range(num_iterations):
  history = tf_model.fit( shuffle_ds, epochs=epochs, batch_size=batch_size, validatio

  for key in train_history:
    train_history[key].extend(history.history[key])

  clear_output()
  sml_tf.plot_training_results(tf_model, train_history, x_train, y_train, x_validate,
```





Qualtizing the Model for TFLite

```
import numpy as np
def representative_dataset_generator():
  for value in x_validate:
    yield[np.array(value, dtype=np.float32, ndmin=2)]


# Unquantized Model
converter = tf.lite.TFLiteConverter.from_keras_model(tf_model)
tflite_model_full = converter.convert()
print("Full Model Size", len(tflite_model_full))

# Quantized Model
converter = tf.lite.TFLiteConverter.from_keras_model(tf_model)
converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]
converter.representative_dataset = representative_dataset_generator
tflite_model_quant = converter.convert()

print("Quantized Model Size", len(tflite_model_quant))
```

```
    INFO:tensorflow:Assets written to: /tmp/tmpy8kneu2f/assets
    INFO:tensorflow:Assets written to: /tmp/tmpy8kneu2f/assets
    Full Model Size 2560
    INFO:tensorflow:Assets written to: /tmp/tmpv12ki8l1/assets
    INFO:tensorflow:Assets written to: /tmp/tmpv12ki8l1/assets
    Quantized Model Size 2720
```

Uploading the Model Back to SensiML Project

```
class_map_tmp = {k:v+1 for k,v in class_map.items()} #+1 because 0 is unknown

dsk.pipeline.set_training_algorithm("Load Model TF Micro",
                                    params={"model_parameters":{
                                            "tflite":sml_tf.convert_tf_lite(tflite_mo
                                    "class_map":class_map_tmp,
                                    "estimator_type":"classification",
                                    "threshold":0.0,
                                    "train_history":train_history,
                                    "model_json":tf_model.to_json()})

dsk.pipeline.set_validation_method("Recall",params={})
dsk.pipeline.set_classifier("TF Micro", params={})
dsk.pipeline.set_tvo()
results, stats = dsk.pipeline.execute()
```

```
Executing Pipeline with Steps:

------------------------------------------------------------------------
 0.     Name: Continuous Final                          Type: query
------------------------------------------------------------------------
------------------------------------------------------------------------
 1.     Name: Windowing                                 Type: segmenter
------------------------------------------------------------------------
------------------------------------------------------------------------
 2.     Name: Strip                                     Type: transform
------------------------------------------------------------------------
------------------------------------------------------------------------
 3.     Name: generator_set                             Type: generatorset
------------------------------------------------------------------------
------------------------------------------------------------------------
 4.     Name: selector_set                              Type: selectorset
------------------------------------------------------------------------
------------------------------------------------------------------------
 5.     Name: Min Max Scale                             Type: transform
------------------------------------------------------------------------
------------------------------------------------------------------------
 6.     Name: tvo                                       Type: tvo
------------------------------------------------------------------------
        Classifier: TF Micro


        Training Algo: Load Model TF Micro
                class_map: {'Normal': 1, 'Over Pronation': 2, 'Over Supination':
                estimator_type: classification
                model_json: {"class_name": "Sequential", "config": {"name": "seq
                model_parameters: {'tflite': '1c00000054464c3314002000040008000c
                threshold: 0.0
                train_history: {'loss': [46.95075607299805, 44.74125671386719, 3

        Validation Method: Recall



------------------------------------------------------------------------




Results Retrieved... Execution Time: 0 min. 0 sec.
```

```
results.summarize()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-30-468118724af2> in <module>()
----> 1 results.summarize()

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

SEARCH STACK OVERFLOW

Confusion Matrix

```
model = results.configurations[0].models[0]
model.confusion_matrix_stats['validation']
```

```
CONFUSION MATRIX:
              NormalOver PronationOver Supination PronationSupination       UNK
    Normal       9.0       0.0       0.0       0.0       0.0       0.0       0.0
Over Pronation       0.0      16.0       0.0       0.0       0.0       0.0
Over Supination       0.0       0.0      16.0       0.0       0.0       0.0
 Pronation       0.0       0.0       0.0      15.0       0.0       0.0       0.0
Supination       0.0       0.0       0.0       0.0      11.0       0.0       0.0

     Total         9        16        16        15        11         0         0

PosPred(%)     100.0     100.0     100.0     100.0     100.0
```

Double-click (or enter) to edit

```
model.knowledgepack.save("TFu_With_SensiML_Features")
```

```
Knowledgepack 'TFu_With_SensiML_Features' updated.
{'class_map': {'1': 'Normal',
  '2': 'Over Pronation',
  '3': 'Over Supination',
  '4': 'Pronation',
  '5': 'Supination'},
 'configuration_index': '0',
 'cost_summary': {'framework': {'flash': 0,
   'latency': 0,
   'sram': 0,
   'stack': 0},
  'model_size': 2593,
  'neurons': 2593,
  'pipeline': [{'flash': 0,
    'latency': 0.0,
    'name': 'Windowing',
    'sram': 0,
    'stack': 0,
    'type': 'segmenter'},
   {'flash': 0,
    'latency': 0.0,
    'name': 'Strip',
    'sram': 0,
    'stack': 0,
    'type': 'transform'},
   {'flash': 0,
    'latency': 0.0,
    'name': 'generator_set',
    'per_generator_costs': {'75th Percentile': {'flash': 0,
      'latency': 0.0,
      'name': '75th Percentile',
      'num_features': 1,
      'num_iterations': 1,
      'sram': 0,
      'stack': 0,
      'type': 'generator'},
     'Median': {'flash': 0,
      'latency': 0.0,
      'name': 'Median',
      'num_features': 2,
      'num_iterations': 2,
      'sram': 0,
      'stack': 0,
      'type': 'generator'},
     'Minimum': {'flash': 0,
      'latency': 0.0,
      'name': 'Minimum',
      'num_features': 1,
      'num_iterations': 1,
      'sram': 0,
      'stack': 0,
      'type': 'generator'}},
    'sram': 0,
    'stack': 0,
    'type': 'generatorset'},
   {'flash': 0,
    'latency': 0.0,
    'name': 'Min Max Scale',
```

Flashing

```
!pip install qgrid
```

```
!pip install bqplot
```

```
from sensiml import SensiML
from sensiml.widgets import *

dsk = SensiML()
FlashWidget(dsk, folder='pack').create_widget()
```

```
# Replace <Your Folder> with the directory folder path of your Knowledge Pack
# Note that the folder path needs double backslashes. See example:
# C:\\Users\\YourName\\Documents\\notebooks\\knowledgepacks
```

```
    /usr/local/lib/python3.7/dist-packages/sensiml/client.py:112: UserWarning: Confi
      mgc("%config Completer.use_jedi = False")
```

Platform Nordic Thingy

Binary

Flash Method J-Link   Flash