

cmcd398-finance-honours

Release 0.0.1

Connor Robert McDowall

Jan 18, 2022

Python Module Index	11
Index	13

class src.cmcd398_finance_honours.**CustomLossFunctionExample**

Example Custom Loss Function Class

Args:

tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)

Call for loss function

Args:

y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:

: Loss

class src.cmcd398_finance_honours.**CustomSharpeMetric** (**args*, ***kwargs*)

Custom Sharpe Metric

Args:

tf (env): Tensorflow keras metrics environment

result ()

Result return

Returns:

Metric:

update_state (*y_true*, *y_pred*, *sample_weight=None*)

Update state position

Args:

y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions *sample_weight* (int, optional): Sample weights. Defaults to None.

src.cmcd398_finance_honours.**analytical_analysis** ()

Tests symbolic math functionality

src.cmcd398_finance_honours.**autodiff_guide** (*example*)

Execute autodiff examples from Tensorflow resources.

Used to help gain an understanding of different functionalities (Demonstration Purposes Only)

Args:

example (int): Example to implement

: 1 - 'simple' : 2 - 'simple_tensor' : 3 - 'simple_model' : 4 - 'control_tape' : 5 - 'control_tensor_tape' : 6 - 'stop_recording' : 7 - 'watch_multiple_variables' : 8 - 'high-

er_order_derivatives' : 9 - 'jacobian' : 10- 'hessian_newton'

`src.cmcd398_finance_honours.build_tensor_flow_model (train_dataset, val_dataset, test_dataset, model_name, all_features, all_inputs, selected_optimizer, selected_losses, selected_metrics, finance_configuration=True)`

Builds tensorflow neural networks

Args:

train_dataset (ds): Training dataset val_dataset (ds): Validation dataset test_dataset (ds): Testing dataset model_name (str): Run name all_features ([type]): [description] all_inputs ([type]): [description] selected_optimizer (str): optimizer to use selected_losses (str): Loss function to use selected_metrics (list): List of selection metrics to use finance_configuration (bool, optional): Run the complex configuration options. Defaults to True.

Returns:

model (tf.model): Tensorflow model loss (float): Loss metric accuracy (float): Accuracy metric

`src.cmcd398_finance_honours.configure_training_ui (project, api_token)`

Configures Neptune.ai API, integrated with Github, to record and monitor dashboard performance

Args:

project (str): Name of Neptune.ai project api_token (str): API token to authenticate account

Returns:

var: Neptune callback configuration

`src.cmcd398_finance_honours.convert_datetime_to_int (dataframe, column_name)`

Convert datetime formats to int

Args:

dataframe (df): Dataframe column_name (str): column name to convert

Returns:

df: Updated dataframe

`src.cmcd398_finance_honours.convert_txt_to_tex (fp_in, fp_out, replace_text=False, replacement_text=None)`

Convert text files to latex format

Args:

fp_in (str): Path in fp_out (str): Path out replace_text (bool, optional): Replace text in file. Defaults to False. replacement_text ([type], optional): Text to replace. Defaults to None.

`src.cmcd398_finance_honours.create_dataframes (csv_location, multi_csv)`

Creates dataframes

Args:

csv_location (str): directory of csvs multi_csv (bool): True/False for loading multiple csvs

Returns:

dataframe: Returns dataframe after convert the csv file

`src.cmcd398_finance_honours.create_fama_factor_models (model_name, selected_losses, factor_location, prediction_location, dependant_column, regression_dictionary, realised_returns=False)`

Creates pricing models and regressions from predictions

Args:

model_name (Str): Name of run selected_losses (Str): Name of loss function factor_location (Str): Factors Directory location prediction_location (Str): Prediction Directory location dependant_column (Str): dependant variable regression_dictionary (Str): Checks for regressions realised_returns (bool, optional): use realised values. Defaults to False.

`src.cmcd398_finance_honours.create_feature_lists (list_of_columns, categorical_assignment)`

Creates required feature lists of normalisation and encoding

Args:

list_of_columns ([type]): [description] categorical_assignment ([type]): [description]

Returns:

numerical_features: categorical_features:

`src.cmcd398_finance_honours.create_learning_curves (model_name, selected_loss, model_history=None, from_load_file=True)`

Creates learning curves to model training losses

Args:

model_name (str): Run name selected_loss (str): Selected loss function model_history (str, optional): Load a model history. Defaults to None. from_load_file (bool, optional): Load from a file instead. Defaults to True.

`src.cmcd398_finance_honours.create_original_list_of_columns (dataframe)`

Gets the original dataframe list

Args:

dataframe (df): Pandas dataframe

`src.cmcd398_finance_honours.create_tensorflow_models (data_vm_directory, list_of_columns, categorical_assignment, target_column, chunk_size, resizing_options, batch_size, model_name, selected_optimizer, selected_losses, selected_metrics, split_data=False, trial=False, sample=False)`

Creates the tensorflow models combining all the analysis

Args:

data_vm_directory (str): Directory of source data list_of_columns (str): Directory to txt file with list of columns categorical_assignment (dict): Dictionary of features to be categorical target_column (str): Target column chunk_size (int): Chunk size resizing_options (list): List of boolean variables for resizing the dataset batch_size (int): Batch size for creating tensor slices model_name (str): Run name selected_optimizer (str): optimizer to use selected_losses (str): Loss function to use selected_metrics (list): List of selection metrics to use split_data (bool, optional): Boolean to split the original dataset. Defaults to False. trial (bool, optional): Boolean to take a smaller dataset. Defaults to False. sample (bool, optional): Boolean to take an even smaller dataset. Defaults to False.

`src.cmcd398_finance_honours.create_tf_dataset (dataframe, target_column, shuffle=True, batch_size=32)`

Set target variable and converts dataframe to tensorflow dataset

Args:

df (dataframe): dataframe target_column (str): Column used to predict for labels shuffle (bool, optional): [description]. Defaults to True. batch_size (int, optional): Sets batch size. Defaults to 32.

Returns:

[ds]: Tensorflow dataset

`src.cmcd398_finance_honours.custom_capm_metric (factors)`

Call for CAPM metric

Args:

factors (tensor): Tensor of factors

`class src.cmcd398_finance_honours.custom_hp (extra_tensor=None, reduction='auto', name='custom_hp')`

Custom Hedge Portfolio Function Class

Args:
 tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)
 Call for hp loss function

Args:
y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:
 : Loss

`src.cmcd398_finance_honours.custom_hp_metric (y_true, y_pred)`
 Call for hp metric

Args:
y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:
 : Loss

class `src.cmcd398_finance_honours.custom_hp_mse` (*extra_tensor=None*, *reduction='auto'*, *name='custom_hp_mse'*)
 Custom Hedge Portfolio + Mean Squared Error Loss Function Class

Args:
 tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)
 Call for hp mse loss function

Args:
y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:
 : Loss

class `src.cmcd398_finance_honours.custom_information` (*extra_tensor=None*, *reduction='auto'*, *name='custom_information'*)
 Custom Information Function Class

Args:
 tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)
 Call for information loss function

Args:
y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:
 : Loss

`src.cmcd398_finance_honours.custom_information_metric (y_true, y_pred)`
 Call for information metric

Args:
y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:
 : Loss

class `src.cmcd398_finance_honours.custom_information_mse` (*extra_tensor=None*, *reduction='auto'*, *name='custom_information_mse'*)
 Custom Information + Mean Squared Error Function Class

Args:
tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)
Call for information mse loss function

Args:
y_true (tf): Tensor of realisations y_pred (tf): Tensor of predictions

Returns:
: Loss

class src.cmcd398_finance_honours.**custom_mse** (*extra_tensor=None*, *reduction='auto'*, *name='custom_mse'*)

Custom Mean Squared Error Function Class

Args:
tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)
Call for mse loss function

Args:
y_true (tf): Tensor of realisations y_pred (tf): Tensor of predictions

Returns:
: Loss

src.cmcd398_finance_honours.**custom_mse_metric** (*y_pred*, *y_true*)

Call for mse metric

Args:
y_true (tf): Tensor of realisations y_pred (tf): Tensor of predictions

Returns:
: Loss

class src.cmcd398_finance_honours.**custom_sharpe** (*extra_tensor=None*, *reduction='auto'*, *name='custom_sharpe'*)

Custom Sharpe Ratio Function Class

Args:
tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)
Call for sharpe loss function

Args:
y_true (tf): Tensor of realisations y_pred (tf): Tensor of predictions

Returns:
: Loss

src.cmcd398_finance_honours.**custom_sharpe_metric** (*y_true*, *y_pred*)

Call for sharpe metric

Args:
y_true (tf): Tensor of realisations y_pred (tf): Tensor of predictions

Returns:
: Loss

class src.cmcd398_finance_honours.**custom_sharpe_mse** (*extra_tensor=None*, *reduction='auto'*, *name='custom_sharpe_mse'*)

Custom Sharpe + Mean Squared Error Function Class

Args:

tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)

Call for sharpe mse loss function

Args:

y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:

: Loss

class src.cmcd398_finance_honours.**custom_treynor** (*extra_tensor=None*, *reduction='auto'*, *name='custom_treynor'*)

Custom Treynor Ratio Function Class

Args:

tf (env): Tensorflow keras losses environment

call (*y_true*, *y_pred*)

Call for treynor loss function

Args:

y_true (tf): Tensor of realisations *y_pred* (tf): Tensor of predictions

Returns:

: Loss

src.cmcd398_finance_honours.**download_test_data** ()

Download test data

Returns:

[list]: List of different dataframes (Total, training, validation, testing)

src.cmcd398_finance_honours.**encode_tensor_flow_features** (*train_df*, *val_df*, *test_df*, *target_column*, *numerical_features*, *categorical_features*, *categorical_dictionary*, *size_of_batch=256*)

Encodes tensorflow features

Args:

train_df (df): Training dataframe *val_df* (df): Validation dataframe *test_df* (df): Testign dataframe *target_column* (str): Target column for prediction *numerical_features* (list): List of numerical features *categorical_features* (list): List of categorical features *categorical_dictionary* (dict): Dictionary of categorical features *size_of_batch* (int, optional): Batch size. Defaults to 256.

Returns:

all_features (tf.layer): Tensorflow layer of all features *all_inputs* (list): Tensorflow layer of all inputs *train_dataset* (ds): Training Dataset *val_dataset* (ds): Validation Dataset *test_dataset* (ds): Testing Dataset

src.cmcd398_finance_honours.**execute_conversion_options** (*model_name*, *selected_losses*, *hp_ols=False*, *pooled_ols=False*, *true_excess_returns=False*)

Do all the text to tex conversion.

Args:

model_name (Str): *selected_losses* ([type]): [description] *hp_ols* (bool, optional): Do HP OLS. Defaults to False. *pooled_ols* (bool, optional): Do pooled OLS. Defaults to False. *true_excess_returns* (bool, optional): Do realised OLS. Defaults to False.

src.cmcd398_finance_honours.**get_category_encoding_layer** (*name*, *dataset*, *dtype*, *max_tokens=None*)

Get encoding layer for categorical variables

Args:

name (str): [description] dataset (ds): Tensroflow dataset dtype (str): Datatype for encoded variable max_tokens (int, optional): Number of max tokens. Defaults to None.

Returns:

[lambda]: lambda function for the encoded feature

```
src.cmcd398_finance_honours.get_normalization_layer ( name, dataset )
```

Get normalisation error

Args:

name ([type]): [description] dataset ([type]): [description]

Returns:

[type]: [description]

```
src.cmcd398_finance_honours.implement_test_data ( dataframe, train, val, test, full_implementation=False )
```

[summary]

Args:

dataframe (df): Dataframe train (df): Training dataframe val (df): Validation dataframe test (df): Testing dataframe full_implementation (bool, optional): Implement a full implementation. Defaults to False.

```
src.cmcd398_finance_honours.loss_function_testing ( )
```

Uses tensorflow autodifferentiation functionality to confirm differentiable nature and feasibility of custom loss functions. Note: code verbatim from tensorflow guide. Merely for illustration purposes

```
src.cmcd398_finance_honours.make_tensorflow_predictions ( model_name, model_directory, selected_losses, dataframe_location, custom_objects )
```

Makes tensorflo predictions

Args:

model_name (str): Run name model_directory (str): Model directory selected_losses (str): Loss function dataframe_location (str): Directory to the government custom_objects (list): List of custom objects in the tensorflow model

```
src.cmcd398_finance_honours.monitor_memory_usage ( units, cpu=False, gpu=False )
```

Function to monitor both CPU & GPU memory consumption

Args:

units (int): Memory units (0 = Bytes, 1 = KB, 2 = MB, 3 = GB, 4 = TB, 5 = PB) cpu (bool, optional): CPU Information. Defaults to False. gpu (bool, optional): GPU Information. Defaults to False.

```
src.cmcd398_finance_honours.partition_data ( data_location, data_destination )
```

Converts dta format to a series of 100k line csvs

Args:

data_location (str): directory to source dta file data_destination (str): directory to store csvs

```
src.cmcd398_finance_honours.perform_tensorflow_model_inference ( model_name, sample )
```

Perform evaluations from model (must be configured)

Args:

model_name ([type]): [description] sample ([type]): [description]

Returns:

[type]: [description]

`src.cmcd398_finance_honours.process_vm_dataset (data_vm_dta, size_of_chunks, resizing_options, save_statistics=False, sample=False)`

This script processes the training and testing datasets for Tensorflow following the classify structured data with feature columns tutorial

Args:

data_vm_dta (str): Directory size_of_chunks (int): Size of chunks e.g., 10000
resizing_options ([type]): [description] save_statistics (bool, optional): Save Statistics. Defaults to False. sample (bool, optional): Process a smaller set of memory. Defaults to False.

Returns:

df: Complete dataset

`src.cmcd398_finance_honours.ranking_function ()`

Ranking function to produce charts for demonstration purposes

Args:

type ([type]): String for desired ranking functions

`src.cmcd398_finance_honours.reconfigure_gpu (restrict_tf, growth_memory)`

Reconfigures GPU to either restrict the numner of GPU

or enable allocated GPU to grow on use oppose to allocating all memory

Args:

restrict_tf (bool): True/False to restrict number of GPUs growth_memory (bool): True/False to enable contuous

`src.cmcd398_finance_honours.reduce_mem_usage (props)`

Function reducing the memory size of a dataframe from Kaggle

<https://www.kaggle.com/arjanso/reducing-dataframe-memory-size-by-65>

Args:

props (dataframe): Pandas Dataframe

Returns:

props (dataframe): Resized Pandas Dataframe

`src.cmcd398_finance_honours.reinforement_learning (model, env, target_vec)`

Exammples to reinforcement learning

Args:

model (tf.model): Configured model env (env): Reinforcement learning environment
target_vec ():

`src.cmcd398_finance_honours.replace_nan (df, replacement_method)`

Replace/Remove nan files in a dataframe

Args:

df (dataframe): Pandas Dataframe replacement_method (int): Specify replacement methods
{0: remove rows with nan values; {1: remove columns with nan values; {2: fill nan with column mean; {3: fill nan with column median}

Returns:

dataframe: Updated pandas dataframe

`src.cmcd398_finance_honours.resizing_dataframe (dataframe, resizing_options)`

Resizes the dataframe to control number of factors (fullset) or original ~178, remove mircro and nano size groups, and optimise variable type by reducing float64 types to float32.

Args:

dataframe (df): Data in dataframe format resizing_options (list): List of True/False statements to control sizing statements.

Returns:

df: Resized dataframe

```
src.cmcd398_finance_honours.sass_access ( dataframe )
```

Remote access to SAS functionalities

Args:

dataframe (dataframe): Data to convert to SAS datafile

```
src.cmcd398_finance_honours.save_df_statistics ( df, frame_set, statistics_location, data_location )
```

Save dataframe summary statistics

Args:

df (df): Dataframe frame_set (str): name of frame statistics_location (str): directory to store stats data_location (str): directory to store file

```
src.cmcd398_finance_honours.set_gpus ( manual_GPU_device_placement=False )
```

Set GPU configuration

Args:

manual_GPU_device_placement (bool, optional): Mnnual place CPU. Defaults to False.

Returns:

[sys]: GPU Device configuration

```
src.cmcd398_finance_honours.shuffle_columns ( df, column_name )
```

Shuffles columns to front of the dataframe

Args:

df (df): Dataframe column_name (Str): Column name

Returns:

[df]: Dataframe

```
src.cmcd398_finance_honours.sort_data_chronologically ( data_directory, size_of_chunks, set_top_500=False )
```

Processes the dataset to the following chronilogical order

Training: Before 1990 [1861,1989] Validation: Between 1990 and 1999 [1990,1999] Testing: After 2000 [2000,2021]

Args:

data_directory (str): Directory of stored data on the virtual instance size_of_chunks (int): Chunk_size for reading pandas dataframes set_top_500 (bool, optional): Select top 500 equities. Defaults to False.

```
src.cmcd398_finance_honours.split_vm_dataset ( data_vm_directory, create_statistics, split_new_data, create_validation_set )
```

Splits the dta dataset into training, testing, and validation sets

Args:

data_vm_directory (str): Directory locating dta file (combined factors) create_statistics (bool): True/False to create summary statistics split_new_data (bool): True/False to split the data into training/testing create_validation_set (bool): Treu/False (nested) to create validation set

- genindex
- modindex
- search

S

src

src.cmd398_finance_honours, [XX](#)

A

analytical_analysis() (in module src.cmcd398_finance_honours), 1
autodiff_guide() (in module src.cmcd398_finance_honours), 1

B

build_tensor_flow_model() (in module src.cmcd398_finance_honours), 2

C

call() (src.cmcd398_finance_honours.custom_hp method), 4
call() (src.cmcd398_finance_honours.custom_hp_mse method), 4
call() (src.cmcd398_finance_honours.custom_information method), 4
call() (src.cmcd398_finance_honours.custom_information_mse method), 5
call() (src.cmcd398_finance_honours.custom_mse method), 5
call() (src.cmcd398_finance_honours.custom_sharpe method), 5
call() (src.cmcd398_finance_honours.custom_sharpe_mse method), 6
call() (src.cmcd398_finance_honours.custom_treynor method), 6
call() (src.cmcd398_finance_honours.CustomLossFunctionExample method), 1
configure_training_ui() (in module src.cmcd398_finance_honours), 2
convert_datetime_to_int() (in module src.cmcd398_finance_honours), 2
convert_txt_to_tex() (in module src.cmcd398_finance_honours), 2
create_dataframes() (in module src.cmcd398_finance_honours), 2
create_fama_factor_models() (in module src.cmcd398_finance_honours), 2
create_feature_lists() (in module src.cmcd398_finance_honours), 3

create_learning_curves() (in module src.cmcd398_finance_honours), 3
create_original_list_of_columns() (in module src.cmcd398_finance_honours), 3
create_tensorflow_models() (in module src.cmcd398_finance_honours), 3
create_tf_dataset() (in module src.cmcd398_finance_honours), 3
custom_capm_metric() (in module src.cmcd398_finance_honours), 3
custom_hp (class in src.cmcd398_finance_honours), 3
custom_hp_metric() (in module src.cmcd398_finance_honours), 4
custom_hp_mse (class in src.cmcd398_finance_honours), 4
custom_information (class in src.cmcd398_finance_honours), 4
custom_information_metric() (in module src.cmcd398_finance_honours), 4
custom_information_mse (class in src.cmcd398_finance_honours), 4
custom_mse (class in src.cmcd398_finance_honours), 5
custom_mse_metric() (in module src.cmcd398_finance_honours), 5
custom_sharpe (class in src.cmcd398_finance_honours), 5
custom_sharpe_metric() (in module src.cmcd398_finance_honours), 5
custom_sharpe_mse (class in src.cmcd398_finance_honours), 5
custom_treynor (class in src.cmcd398_finance_honours), 6
CustomLossFunctionExample (class in src.cmcd398_finance_honours), 1
CustomSharpeMetric (class in src.cmcd398_finance_honours), 1

D

download_test_data() (in module src.cmcd398_finance_honours), 6

E

encode_tensor_flow_features() (in module src.cmcd398_finance_honours), 6
execute_conversion_options() (in module src.cmcd398_finance_honours), 6

G

get_category_encoding_layer() (in module src.cmcd398_finance_honours), 6
get_normalization_layer() (in module src.cmcd398_finance_honours), 7

I

implement_test_data() (in module src.cmcd398_finance_honours), 7

L

loss_function_testing() (in module src.cmcd398_finance_honours), 7

M

make_tensorflow_predictions() (in module src.cmcd398_finance_honours), 7
module
src.cmcd398_finance_honours, 1
monitor_memory_usage() (in module src.cmcd398_finance_honours), 7

P

partition_data() (in module src.cmcd398_finance_honours), 7
perform_tensorflow_model_inference() (in module src.cmcd398_finance_honours), 7
process_vm_dataset() (in module src.cmcd398_finance_honours), 8

R

ranking_function() (in module src.cmcd398_finance_honours), 8
reconfigure_gpu() (in module src.cmcd398_finance_honours), 8
reduce_mem_usage() (in module src.cmcd398_finance_honours), 8
reinforcement_learning() (in module src.cmcd398_finance_honours), 8
replace_nan() (in module src.cmcd398_finance_honours), 8
resizing_dataframe() (in module src.cmcd398_finance_honours), 8
result() (src.cmcd398_finance_honours.CustomSharpeMetric method), 1

S

sass_access() (in module src.cmcd398_finance_honours), 9
save_df_statistics() (in module src.cmcd398_finance_honours), 9
set_gpus() (in module src.cmcd398_finance_honours), 9
shuffle_columns() (in module src.cmcd398_finance_honours), 9
sort_data_chronologically() (in module src.cmcd398_finance_honours), 9
split_vm_dataset() (in module src.cmcd398_finance_honours), 9
src.cmcd398_finance_honours
module, 1

U

update_state() (src.cmcd398_finance_honours.CustomSharpeMetric method), 1