

# **cmcd398-finance-honours**

*Release 0.0.1*

**Connor Robert McDowall**

Jan 18, 2022



<b>Python Module Index</b>	<b>11</b>
<b>Index</b>	<b>13</b>



**class** src.cmcd398\_finance\_honours.**CustomLossFunctionExample**

**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* )

**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 - 'higher\_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 in 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' )
```

```
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' )``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' )``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' )``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' )``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' )
```

```
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' )
```

```
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' )
```

```
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:**

[lamdba]: 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 differientable 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 )
```

Exammple 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 chronillogical 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 equi-  
ties. 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), 1

**C**

call() (src.cmcd398\_finance\_honours.custom\_hp method), 3  
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), 4  
call() (src.cmcd398\_finance\_honours.custom\_mse method), 4  
call() (src.cmcd398\_finance\_honours.custom\_sharpe method), 5  
call() (src.cmcd398\_finance\_honours.custom\_sharpe\_mse method), 5  
call() (src.cmcd398\_finance\_honours.custom\_treynor method), 5  
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), 2

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), 4  
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), 5  
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), 5

**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), 6

**I**

implement\_test\_data() (in module src.cmcd398\_finance\_honours), 6

**L**

loss\_function\_testing() (in module src.cmcd398\_finance\_honours), 6

**M**

make\_tensorflow\_predictions() (in module src.cmcd398\_finance\_honours), 6  
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), 7

**R**

ranking\_function() (in module src.cmcd398\_finance\_honours), 7  
reconfigure\_gpu() (in module src.cmcd398\_finance\_honours), 7  
reduce\_mem\_usage() (in module src.cmcd398\_finance\_honours), 7  
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), 8  
save\_df\_statistics() (in module src.cmcd398\_finance\_honours), 8  
set\_gpus() (in module src.cmcd398\_finance\_honours), 8  
shuffle\_columns() (in module src.cmcd398\_finance\_honours), 8  
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