**CAPM Model – Documentation**

**Path to model assessment:** D:\Utilitários\Facul\Mestrado\2o Ano\Research\Thesis\Tables\Assessment

**Suggestions:**

1. Merge Target 0 folder with Target 0 – Raw
2. Merge Target 15 folder with Target 15 - Raw
3. Improve importCleanedData.py
   1. More specifically, the importCleanedData which is the most used function to import data throughout the project
4. Improve comments and possibly simplify ModelAssess

**Note:** folders are marked with font 14, subfolders with font 13 and ipynb files with font 12. Folders and subfolders are both in **bold**.

**CreateSample - Folder**

* CalculateBeta.ipynb
* CalculateTechnicalIndicators.ipynb
* Dataset\_MergeFundamentals.ipynb

**SampleData – Sample - Folder**

1. fundamentalData.csv – input of Data\_Preparatiion\_Pipeline.ipynb
2. LongTerm-DataPreparation\_Skew – output of Data\_Preparatiion\_Pipeline.ipynb with Data Preparation steps including variable transformation. In this dataset, the Target is named ‘CAPM’. There’s information regarding the adjustedCAPM but is not used.
3. LongTerm-DataPreparation - output of Data\_Preparatiion\_Pipeline.ipynb with Data Preparation steps excluding variable transformation. In this dataset, the Target is named ‘futAlpha’. There’s information regarding the adjustedCAPM but is not used, in this dataset this column is named ‘newAlpha’
4. df\_clean.csv – similar to LongTerm-DataPreparation, however, it’s more organized. In this dataset, the Target is named ‘CAPM’.
5. df\_clean\_skew.csv - similar to LongTerm-DataPreparation\_skew, however, it’s more organized. In this dataset, the Target is named ‘CAPM’. There’s a columns referring to the ‘newAlpha’ which is called ‘adjCAPM’.
6. df\_target.csv – contains data related to Target creation for all observations used in the study
7. Target\_CAPM – contains Target data for 150k observations and includes the newAlpha which adjusts the CAPM in moments of bad market performance – for the context of this project is not relevant
8. Data\_TechnicalIndicators

**CalculateBeta.ipynb**

For each ticker:

1. Imports data from *mergedPrices* – contains daily closing price data from the ticker and the S&P
2. Calculates daily returns – daily returns are used to calculate the correlation and volatility
3. Calculates rolling correlations with a window of n – in the case of the project n = 63
4. Calculates volatility of the last n trading days
5. Calculates returns of the past quarter
6. Calculates beta of the asset – correlation \* (std(asset)/std(market))
7. Calculates alpha – stockReturns – (beta \* marketReturns)
8. Inserts data into QuarterlyAlpha

**CalculateTechnicalIndicators.ipynb**

For each ticker:

1. Imports data from *mergedPrices* – contains daily closing price data from the ticker and the S&P
2. Calculates daily returns – daily returns are used to calculate the correlation and volatility
3. Calculates rolling correlations with a window of n – in the case of the project n = 63
4. Calculates volatility of the last n trading days
5. Calculates Technical Indicators – the following n are used, 21,42,63,84
   1. MA – average daily returns over the past n trading days
   2. Volatility – std of returns over the past n trading days
   3. Excess Returns
   4. Beta
6. Inserts data into TechnicalIndicator

**Dataset\_MergeFundamentals.ipynb**

1. Imports data from the following tables – balanceSheet, financialRatios, incomeStatement, cashFlows, companyInfo, historicalDividends, stock\_shares
2. Merge incomeStatement, balanceSheet and financialRatios and cashFlows - incomeStatement, balanceSheet and cashFlows are merged considering fillingDate, while financialRatios is merged considering date (financialratios doesn’t have fillingDate)
3. Merge shares, dividends and company info with the dataset created in step 2
4. Change numeric columns to float
5. Insert data into QuarterlyFundamentals

**Data\_Preparation – Folder:**

* Data\_Preparatiion\_Pipeline\_Cloud.ipynb
* Data\_Preparatiion\_Pipeline.ipynb
* Dimensional\_Reduction\_PCA.ipynb
* Outliers\_and\_Skew\_Analysis.ipynb
* Outliers\_Z-Score.ipynb

**Data\_Preparatiion\_Pipeline.ipynb**

* 1. If Cloud = False - Imports data from fundamentalData.csv
  2. If Cloud = True – quarterlyAlphaDataset - BigQuery

1. Filter sample dates – 2003/01/01 – 2023/01/01 – in this case, 2003 is chosen because we want to calculate YoY and TTM features, which will make 1 year of data unusable
2. Calculate nr of days between futDate and fillingDate – this is to guarantee consistency, because it’s possible that in some cases we are not predicting 3 months ahead, and we want to avoid that. An incoherence is created where observations that have more than 100 days of dateDiff are removed.
3. Calculate 1st Feature Engineer – YoY, QoQ and TTM features
4. Impute missing values
5. Perform incoherence checking
6. Calculate 2nd Feature Engineer – Categorical Features (marketCap); OHE (Industry and Sector), Binary Features; Financial Ratios
7. Variable Transformation – Log + Cubic Root Transformations
8. Export data
   1. LongTerm-DataPreparation.csv
   2. LongTerm-DataPreparation\_Skew.csv

**Data Exploration - Folder**

Long-Term-Strategy-Data\_Exploration.ipynb

1. Imports data from LongTerm-DataPreparation.csv
2. Performs basic EDA

Long-Term-Strategy-Data\_Exploration - 2.ipynb

1. Imports data from LongTerm-DataPreparation.csv and LongTerm-DataPreparation\_Skew.csv
2. Performs EDA against the Target Feature on the training set

**Modelling - Folder:**

Long-Term-Strategy-Feature\_Selection.ipynb

1. Imports data from LongTerm-DataPreparation\_Skew.csv and Target\_CAPM.csv
2. Performs feature selection for both Targets using MI and Lasso selection

**Modelling – sub Folder:**

**Gridsearch:**

Long-Term-Strategy-GridSearch\_-\_Target\_0.ipynb

1. Imports data from LongTerm-DataPreparation\_Skew.csv
2. Performs GridSearch for Target 0 for the following models – LogisticRegression, DecisionTree, LightGradientBooster, MLP (Sklearn), Neural Network (Keras)

Long-Term-Strategy-GridSearch\_-\_Target\_15.ipynb

1. Imports data from LongTerm-DataPreparation\_Skew.csv
2. Performs GridSearch for Target 0 for the following models – LogisticRegression, DecisionTree, LightGradientBooster, MLP (Sklearn), Neural Network (Keras)

**Target 0 – sub folder:**

Subset = [‘MI’,’Lasso’,’PCA]

Long-Term-Strategy-Model\_Creation\_-\_Target\_0\_ subset.ipynb

1. Imports data from LongTerm-DataPreparation\_Skew.csv
2. Trains models for Target 0 considering the hyperparameters obtained in Long-Term-Strategy-GridSearch\_-\_Target\_0.ipynb. Models are trained considering the subset

**Target 15 – sub folder:**

Subset = [‘MI’,’Lasso’,’PCA]

Long-Term-Strategy-Model\_Creation\_-\_Target\_15\_ subset.ipynb

1. Imports data from LongTerm-DataPreparation\_Skew.csv
2. Trains models for Target 15 considering the hyperparameters obtained in Long-Term-Strategy-GridSearch\_-\_Target\_15.ipynb. Models are trained considering the subset

**Target 0 - Raw**

Long-Term-Strategy-Model\_Creation\_-\_Target\_0\_ Lasso.ipynb

1. Imports data from LongTerm-DataPreparation\_Skew.csv
2. Trains models for Target 0 considering the hyperparameters obtained in Long-Term-Strategy-GridSearch\_-\_Target\_0.ipynb. Models are trained using Lasso subset and in this case, outlier treatment is not performed.

**Target 15 – Raw**

Subsets = [‘MI’, ‘Lasso’]

Long-Term-Strategy-Model\_Creation\_-\_Target\_15\_ subset.ipynb

1. Imports data from LongTerm-DataPreparation\_Skew.csv
2. Trains models for Target 0 considering the hyperparameters obtained in Long-Term-Strategy-GridSearch\_-\_Target\_0.ipynb. Models are trained using Lasso subset and in this case, outlier treatment is not performed.

**ModelAssess - Folder**

1. Evaluates the performance of the models trained during the project. Both Machine Learning and Economic metrics are assessed.