**Problem Definition:**

Part of the work that CRS does involves purchasing and developing budgets for food commodity aid that will be disbursed to various areas/countries within CRS jurisdiction. Planning can be eased if we had tools to leverage that will allow us to peak into accurate economic trends as well as reliable forecasts. This is why we developed the food price forecasting model that uses macroeconomic trends in exchange rate and consumer prices indices as well as fuel costs to predict the price of a commodity within a 6-month forecasting window. The model is based on a deep learning model known as a Temporal Fusion Transformer that is well designed and suited for time series forecasting.

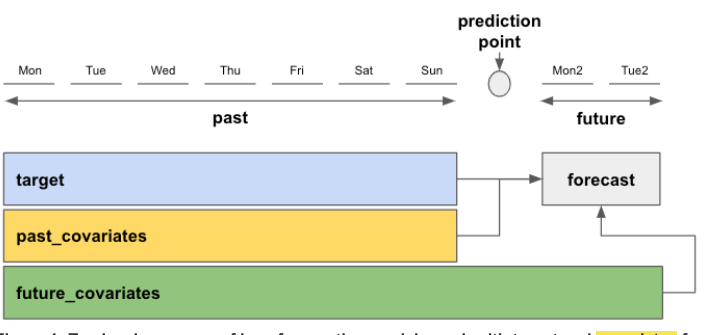
With this tool, we can make better budget decisions for the prices of commodities that CRS programs provide as humanitarian aid. In particular, prices of maize and wheat (Soft Red Winter) can be forecasted for up to 6 months with ~10% error rate whose lower and upper limit values are accounted for.

The walkthrough takes you through wheat (Soft Red Winter) but can easily be replicated for any other commodity with the steps and links to the data given.

Access the colab notebook here: <https://colab.research.google.com/drive/1pKt-uYWYoIzbGQOVZj3or04A3RJDniLU#scrollTo=l6qU0VUc4hdq>.

**Data Used**

Time Series data spanning the dates from **October 2004 to March 2023**.

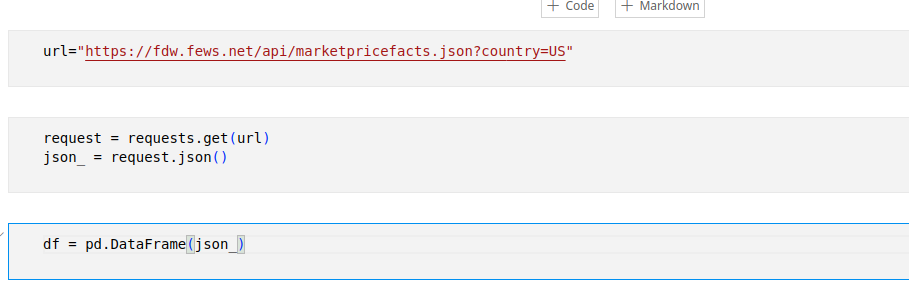
  
Following the above model, we have:

* Target:
  + Soft Red Winter Wheat prices from FEWS NET: <https://fdw.fews.net/api/marketpricefacts.json?country=US>.
* Past covariates:
  + Dollar value from Trading Economics of the defined dates.
  + Consumer price Index: World: International Export Markets (base period 2014 - 2016) from FEWS NET through API link: [*https://fdw.fews.net/api/priceindexvaluefacts.json?fields=simple*](https://fdw.fews.net/api/priceindexvaluefacts.json?fields=simple)*.*
  + Brent Crude Oil from FEWS NET: [*https://fdw.fews.net/api/marketpricefacts.json?country\_code=GB*](https://fdw.fews.net/api/marketpricefacts.json?country_code=GB)
  + *Hard red winter Wheat Prices: https://fdw.fews.net/api/marketpricefacts.json?country=US*
  + Time covariates, model generated. Monthly time covariates spanning the defined dates.
* Future covariates. Model generated and rendered using an encoder that takes in a cyclic seasonality.
* Forecast Horizon: 6 months.
* Historical backtesting timeframe => 2018-2023(5 years).

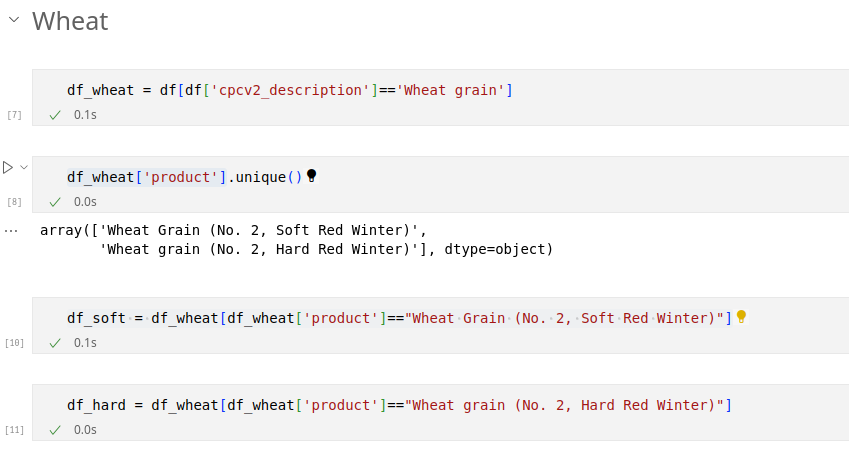
**How to access the different data**:

1. Target Wheat Prices:
   1. Using API and Python:

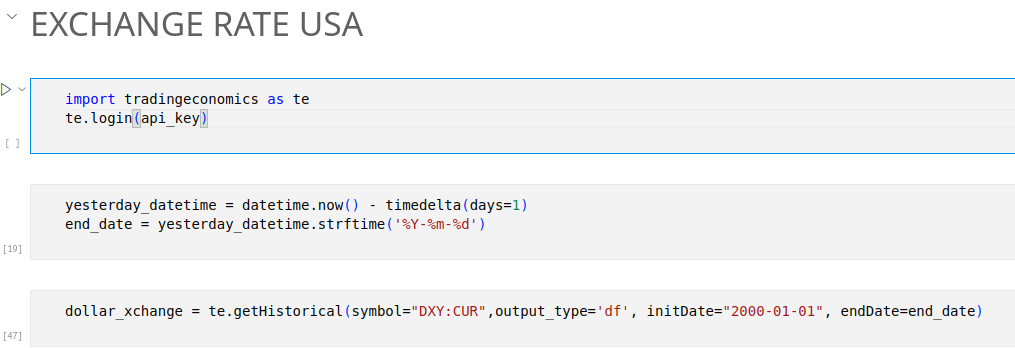
import requests



Select the target wheat series, in this case Soft Red Winter.



1. Past Covariates.
   1. Dollar Value from Trading Economics. I used the python package for Trading economics. Ensure you have the API key saved in your local os environment to ensure it is not visible from the script, hence avoiding leaking.

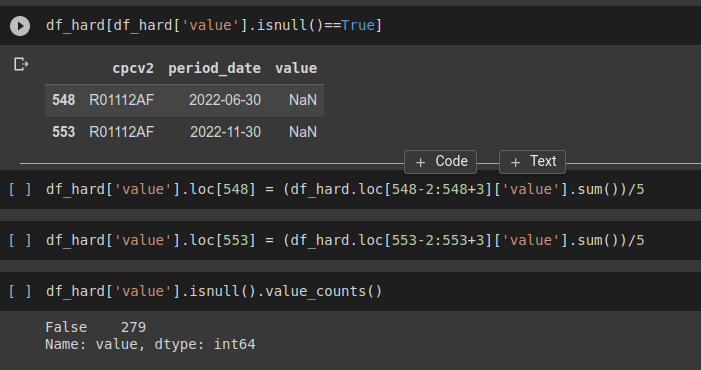
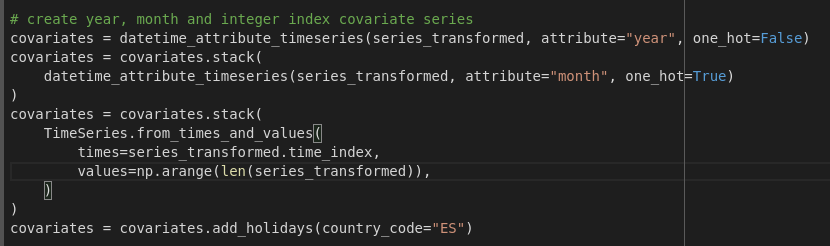


* 1. Consumer Price Index:



* 1. Brent Crude Oil



* 1. Hard Red Winter Series. Access from same API as the target series but specify the product as Hard Red Winter. There are missing values from June and November of 2022 which I imputed with the mean of the previous 2 and the later 3 monthly entries from the missing date. 
  2. Merge the different data-frames into one data-frame then save as csv file. This will then be read as ***Series*** henceforth.
  3. Time Covariates, generated from the same timeframe as the Series timesteps:

**series = TimeSeries.from\_dataframe(df\_final,**

**time\_col='Date',**

**value\_cols=['wheat\_price', 'Hard red winter', 'Dollar value', 'CPI\_value','maize-price','crude oil price']**

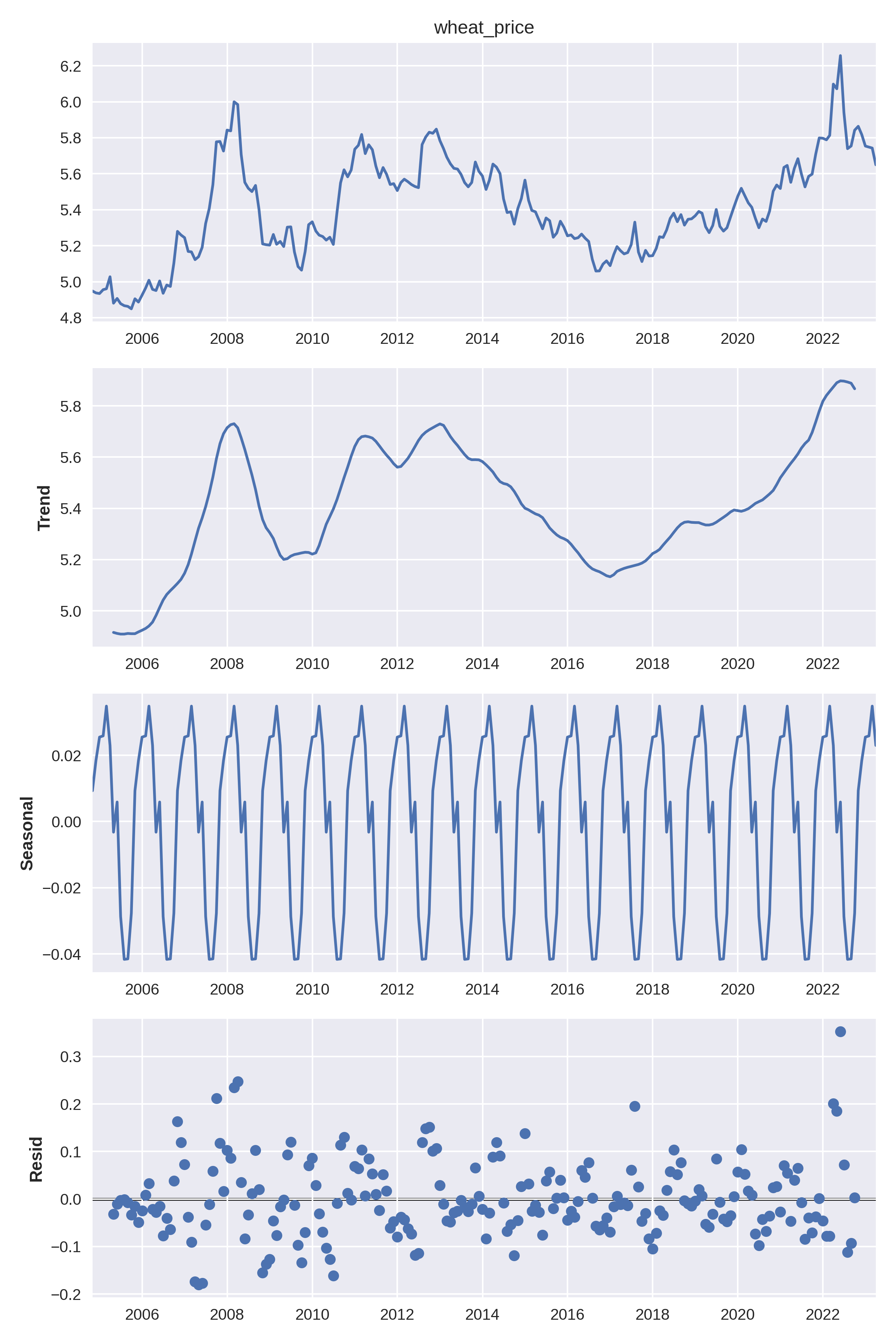
**)**

**Scale all past covariates and target series:**

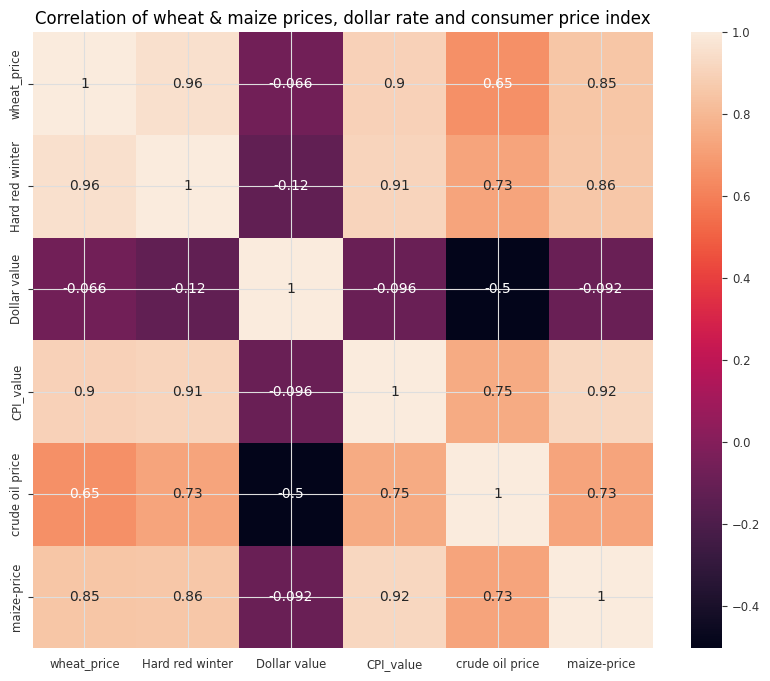
<https://colab.research.google.com/drive/1pKt-uYWYoIzbGQOVZj3or04A3RJDniLU#scrollTo=ABcJhqDkby23&line=6&uniqifier=1>.

**Exploratory Data Analysis and Statistical Tests:**

Trend, Stationarity and Residual plots for the Soft Red Winter Wheat Prices



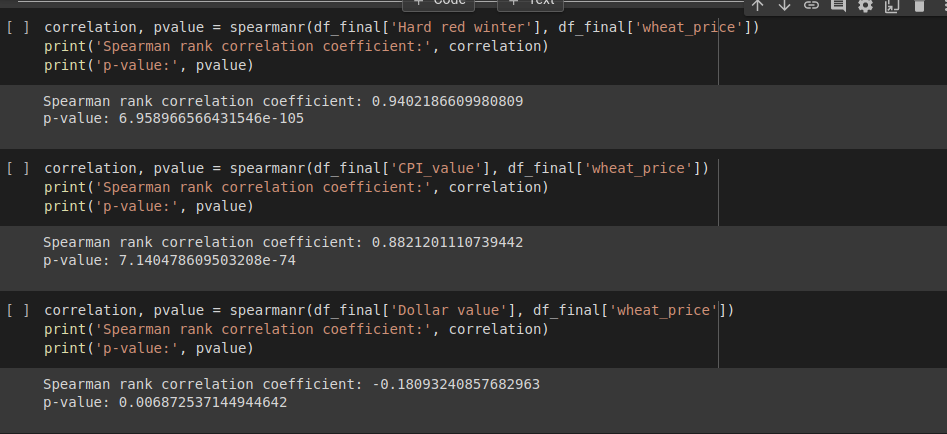
Correlation graph of each covariate to the target series.

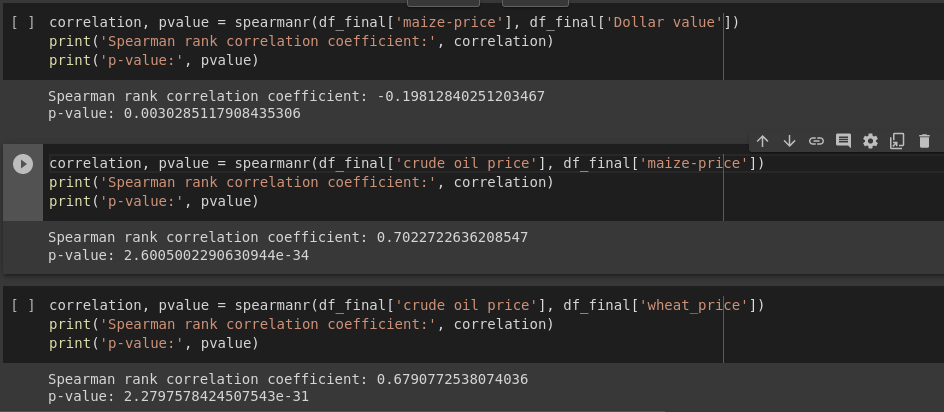


Spearman rank correlation tests.

from scipy.stats import spearmanr

P-value less than 0.05 indicates that there is a significant correlation between the variables being tested. The correlation observed is unlikely to occur by chance alone. All our values have a p-value << 0.05 hence are worth being used when developing a timeseries model.





**Model**

[**Temporal Fusion Transformer Model**](https://unit8co.github.io/darts/generated_api/darts.models.forecasting.tft_model.html) **from Darts Library**

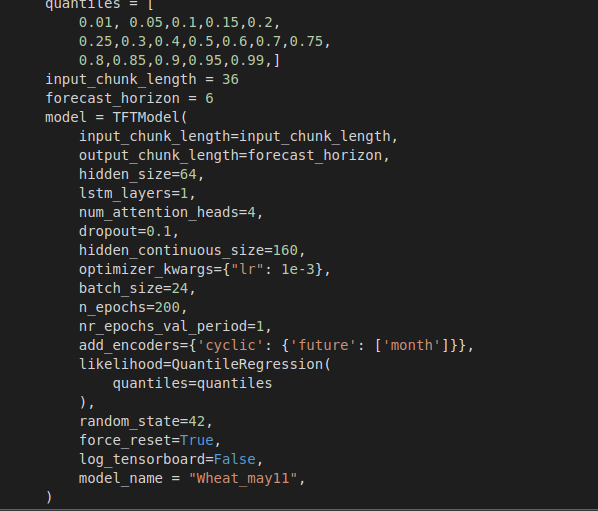
Here identified as TFTModel.

from darts import TimeSeries, concatenate

from darts.models import TFTModel

from darts.utils.likelihood\_models import QuantileRegression

from darts.metrics import mae,mape,smape

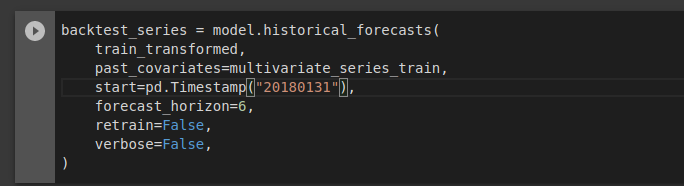


With a GPU machine(either on colab, Azure, or local), fit the model to the data for training:

**model.fit(train\_transformed,past\_covariates=multivariate\_series\_train,verbose=False)**

Training on GPU typically takes < 10 minutes while inference(testing) takes <1 minute.

Backtesting**:**



Saving the model:

**model.save("/content/gdrive\_drive/MyDrive/TIME\_SERIES/Wheat\_may11.pt")**

Modify the path to where you want to save it

Loading the model:

**my\_model=TFTModel.load(“/content/gdrive\_drive/MyDrive/TIME\_SERIES/Wheat\_may11.pt”)**

modify your path to where your model is saved.

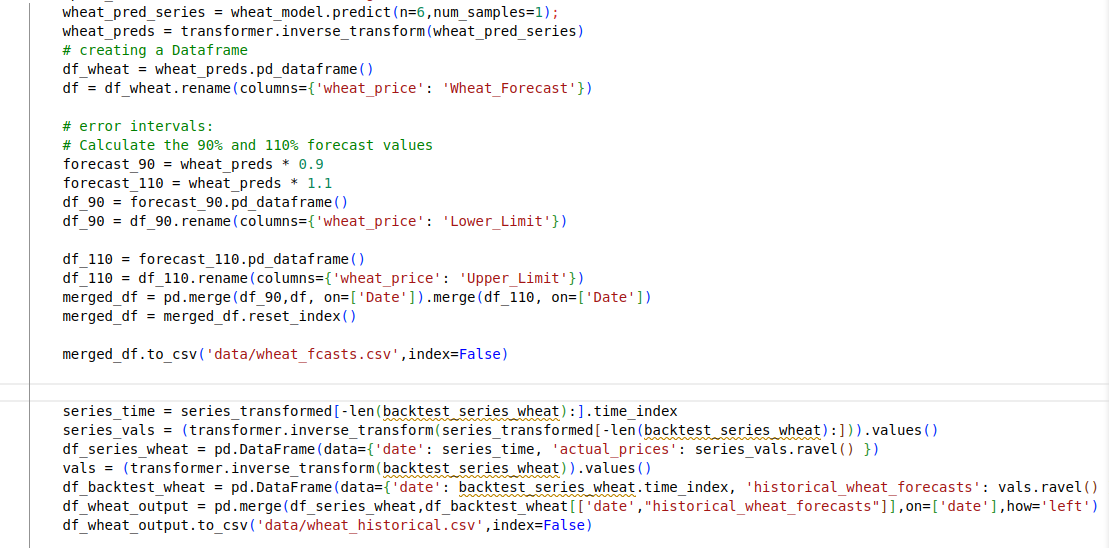
The model will give predictions with period(n) being the forecast horizon in months you are looking to get, in below case n=6:

Num\_samples=1 makes the prediction deterministic as opposed to probabilistic. Num\_samples >>1 will give a probability distribution output of what the predicted values may be.

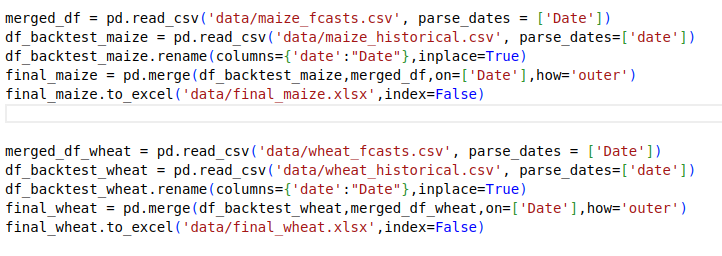
my\_model.predict(n=6,num\_samples=1)

Create a plot of the backtesting results as well as the forecast predictions.

calculate the [mean absolute percentage error](https://colab.research.google.com/drive/1pKt-uYWYoIzbGQOVZj3or04A3RJDniLU#scrollTo=run9jNTSDnd-&line=5&uniqifier=1) as well as a 10% error margin... ie lower limit values for 90% of predicted values, and upper limit values for 110% of predicted values.



Final dataset rendering to be visualized in power bi:



The final\_wheat.xlsx and final\_maize.xlsx files are what will be fed to power bi for visualization.  
The contains dates from 2018 to date, actual historical prices of commodity from 2018, model historical values (aka backtesting values) as well as forecasts of that commodity for the next 6 months, and the respective lower and upper limit values of the forecasts for each month.  
The actual historical prices, model historical prices, model forecasts as well as both the lower and upper limits will be plotted based on the dates.

On Power BI,I used the plotly [Akvelon](https://www.youtube.com/watch?v=mujr5QuR3cc) visualization tool.

The model and data are hosted on a private repository on hugging face spaces. The model requires GPU for training but can perform inference or testing on CPU machines.