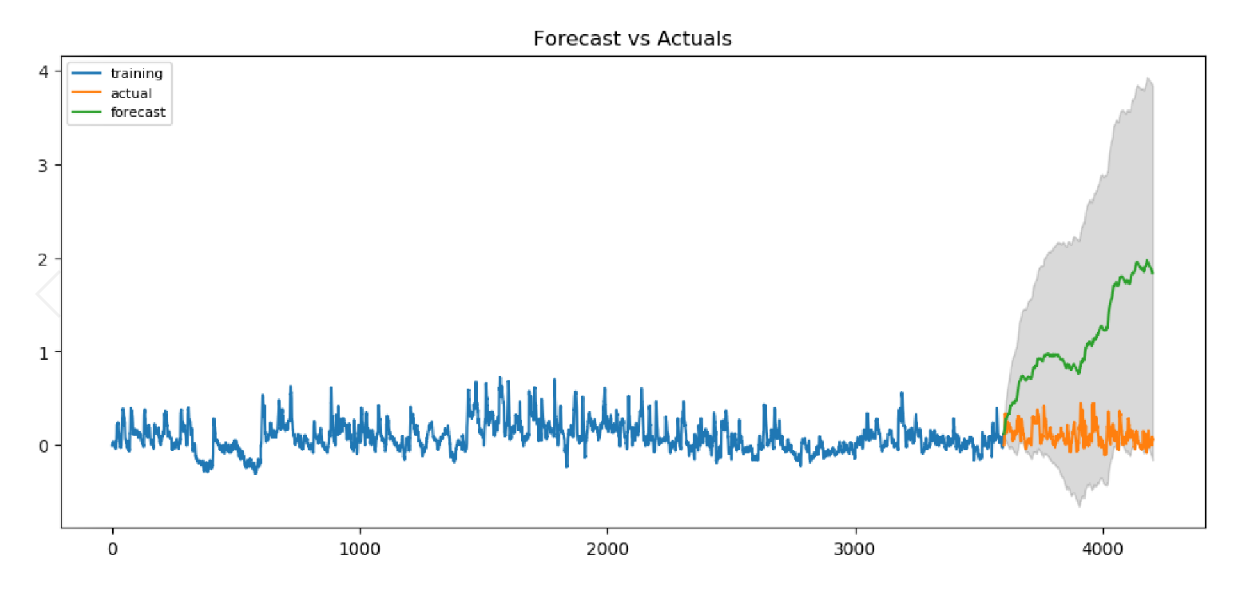
Forecast con step

Allo stato delle cose facciamo un unico forecast con step= length of validation step.Il che genera:



order(3,1,0)

conf = 0.05.

La predizione si discosta troppo dalla serie originale perchè cerchiamo di prevedere troppi valori (step = len(validation set)).

L’idea è dunque quella di fare una predizione per step finchè non si è predetto tutto il validation set.

Ad ogni step si deve:

1. Generare il modello ARIMA e fare il fit
   * order(3,1,0)
   * data: data\_train
   * exog=au\_train
2. fare il forecast di alcuni valori della serie (step =  n valoripredetti)
   * provare con step = 1 (il numero è da bilanciare)
   * passare i valori di au\_valid associati solo ai valori da predire in quello step (non tutto au\_valid)
3. fare l’append dei valori ottenuti con la predizione
   * aggiungere i dati predetti al data\_train
   * aggiungere al set au\_train, le action unit associate ai dati predetti
   * rimuovere da au\_valid i valori delle action unit appena aggiunte al train
4. Ripetere dal punto 2

<https://machinelearningmastery.com/make-sample-forecasts-arima-python/> (se fare one step questo plotta roba credibilie e non da probemi di performance va bene)

Se no bisogna andare più nel profondo :

facendo forecast ad ogni step, serve ogni volta refittare il modello.

Info che ho trovato a riguardo (la letteratura è al quanto confusa sulla cross validazione di modelli per le serie temporali)

<https://stackoverflow.com/questions/56335992/how-to-perform-multi-step-out-of-time-forecast-which-does-not-involve-refitting>

<https://towardsdatascience.com/3-facts-about-time-series-forecasting-that-surprise-experienced-machine-learning-practitioners-69c18ee89387>  
  
<https://machinelearningmastery.com/multi-step-time-series-forecasting/>

Dato finale che possiamo estrarre per la bontà del forecast error (RMSE)

<https://machinelearningmastery.com/time-series-forecasting-performance-measures-with-python/>