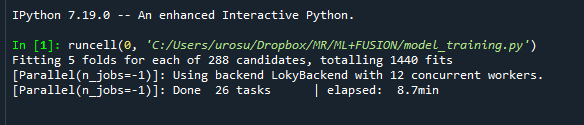
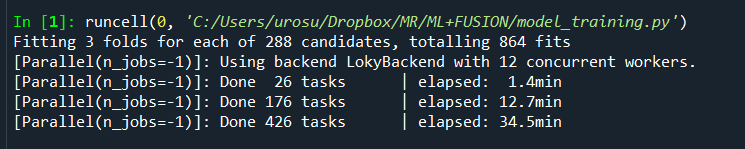
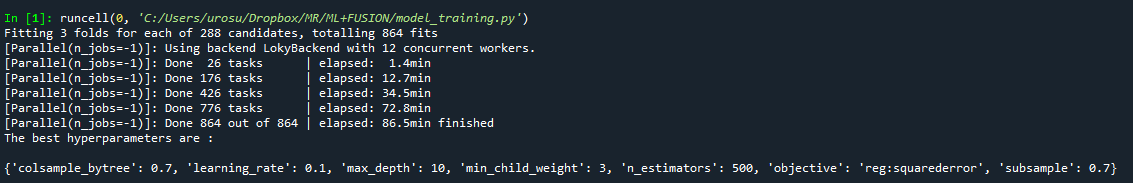
1. Najprej sem pripravil podatke za učenje modela. To je v file extracting\_data.py. Tam pretvorimo neurejene podatke v excelu v podatke primerne za učenje modela.
2. Na začetku sem poizkusil na tak način, da sem za featurje vzel, angle, heat, field in emission. Za target pa posamezne parametre – na primer »Pot«. Tako sem moral za vsak x naučit svoj model. Torej nekje 120k modelov. S tem so problemi, ker je manj podatkov za učenje modela in tudi problem z uporabo naučenih modelov.
3. Odličil sem se, da bo najboljše, da med featurje dodam tudi razdaljo po tokamaku, torej x [m]. Tako dobimo za vsak x 21 različnih kombinacij parametrov. Skupno je tega 2 500 000.
4. Nato sem začel z učenjem modelov. Za prvi target feature, kjer bom preizkušal modele sem izbral parameter »Pot«. Začel sem z ZELO popularnim drevesnim modelom xgboost. Z osnovnimi nastavitvami je model vrnil napovedi, ki so se natančno skladale z enim ali drugim setom podatkov za učenje. V bistvu ni vrnil novih vrednosti ampak je vrnil krivuljo, ki je najbolj podobna enemu setu podatkov iz učenja.
5. Tega nismo iskali… Zato sem najprej probal najti najboljše hyperparametre. To sem delal z metodo GridSearchCV. Najprej sem probal z velikim številom podatkov in veliko variacijo hyperparametrov. To bi trajalo kakšnih 10 ur.

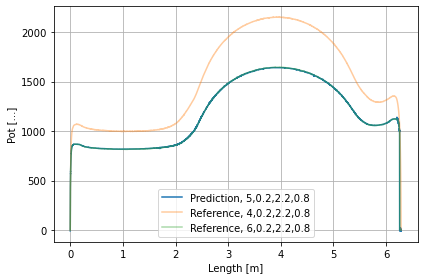


5 folds za 288 kombinacij naših parametrov.

Zato sem zmanjšal podatke na katerih poizkuša. Vzame 20% naključnih vrstic in poizkuša na tem.  
To traja nekje 1 uro.  


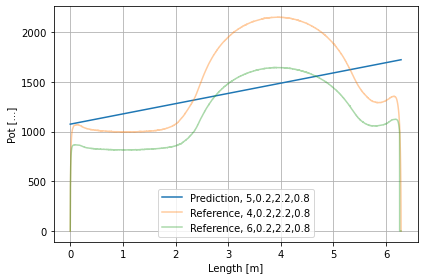
  
  
Parametri so: {'colsample\_bytree': 0.7, 'learning\_rate': 0.1, 'max\_depth': 10, 'min\_child\_weight': 3, 'n\_estimators': 500, 'objective': 'reg:squarederror', 'subsample': 0.7}  
RMSE: 6.524884

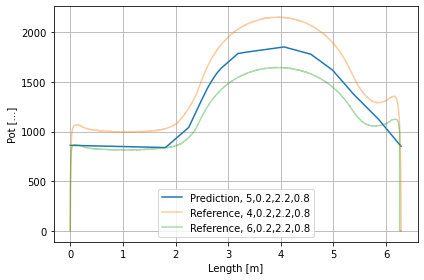
Training score: 0.9998815920732039



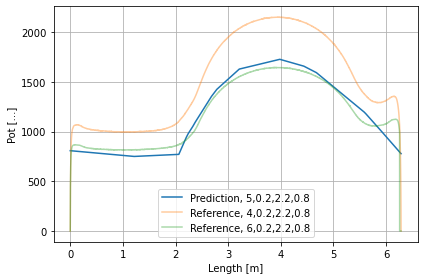
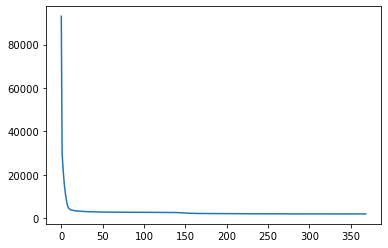
1. To še vedno ni vredu. Zato sem poizkusil z linearno regresijo.

To žal ne bo delalo, ker imamo x- Length za parameter. Bo treba drugače zapakirat? Tako kot sem začel v prvi točki? Za vsak x svoj model?



1. Test z nevronsko mrežo. 20% data, 500 iteracij  
     
   C:\ProgramData\Anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:582: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
2. Test z nevronsko mrežo. 100% data, 1000 iteracij

Iteration 369, loss = 2074.93204292

Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.  
  


To do: Probej samo na max od Pot naučiti