Report

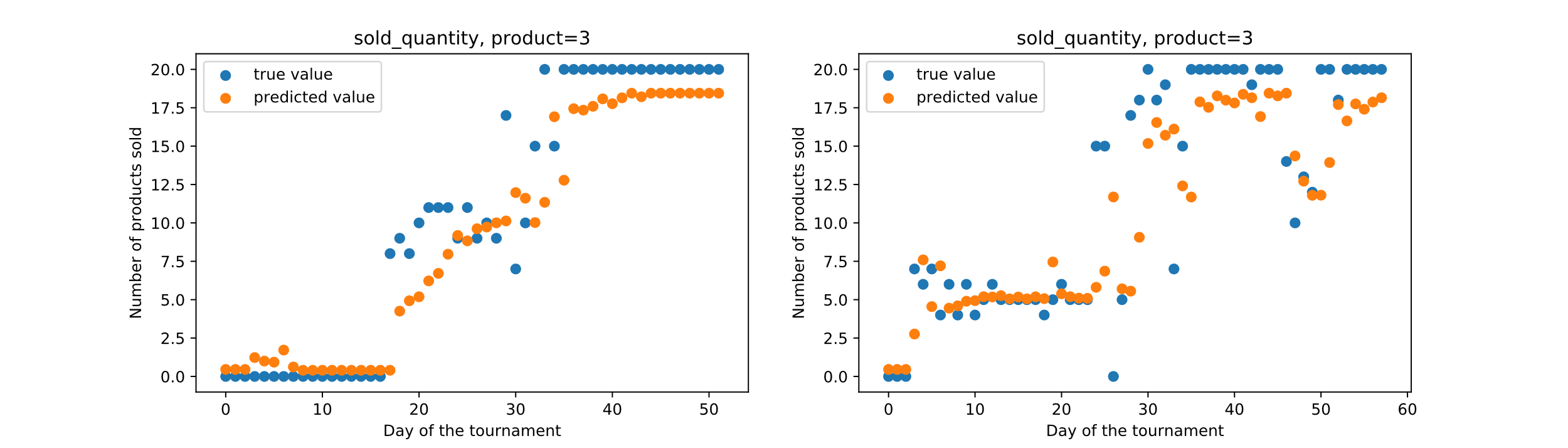
# Trading Strategy

The trading strategy is responsible for deciding the quantity and price for buy and sell at every time-step. It specifies some goal which later the Negotiation Manager will try to fulfill. In our approach, we decided to try to improve upon PredictionBasedTradingStrategy by using a trained scikit-learn model to predict the expected input and output, using the released data. We Implemented a class SklearnTradePredictionStrategy, inheriting from TradePredictionStrategy, to perform the predictions. TradePredictionStrategy allows for predicting an initial prediction for each timestep in the beginning, and then allows for refining the prediction at every step. Therefore, we used two models, one that predict the values (expected input and output) from just product number and timestep, to use in the initial prediction, and one that predict the values based on short history of recent values, for refining the estimates at each timestep.

## Data and Models

We used the data released by the competition organizers, which was gathered from 5000 tournaments. Specifically, we used the stats.csv files which contain detailed statistics for every day in the simulation. For the initial prediction we trained a model for each product to predict values from day of the tournament as a fraction between 0 and 1. For the refining prediction we trained based on the day and previous values. For both kinds of models, we tried several model types and chose a random forest, because it is appropriate for data such as this, which does not have any obvious metric on it, and because it is easier to find good hyperparameters for.

To show the results, here are some plots of actual and predicted values for “sold\_quantity”, for some tournaments:



We can see the models can predict the values at around the true values and are approximately correct for most of the days.