1. Model selection for sales forecasting:

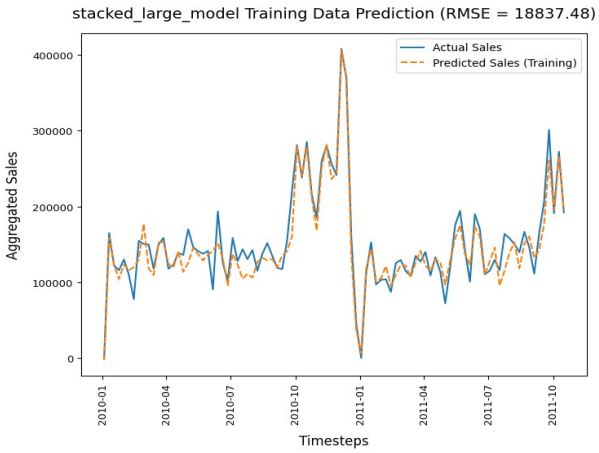
1. 1. ARIMA Results

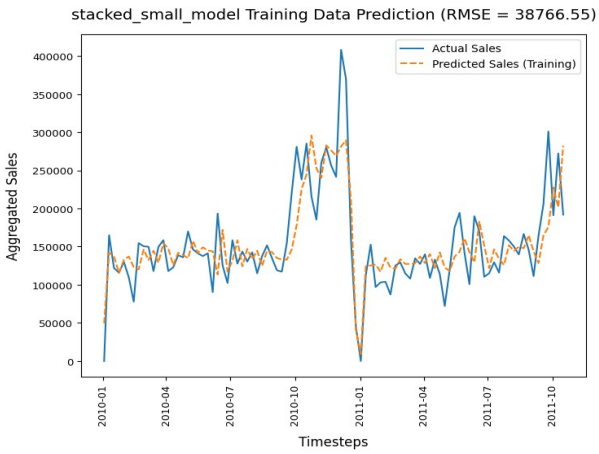
Model Name	Mean CV Test AIC	Mean CV Test RMSE	Hold-Out Test RMSE
arima_model	913.7046	67168.8763	51318.5077

1. 2. LSTM Results

Model Name	Mean CV Test RMSE	Training RMSE	Hold-Out Test RMSE
single_small_model	46870.0676	44424.1684	24463.8177
single_medium_model	48675.8986	37791.6512	37075.7188
single_large_model	51251.3842	29559.2887	29559.2887
stacked_small_model	43967.7002	38766.5459	36868.5032
stacked_medium_model	47608.2670	40604.3960	23587.3019
stacked_large_model	43899.1130	18837.4776	80085.3877

From all the created sales forecasting models the LSTM model with the name "stacked_small_model" seems to have performed the best overall and was therefore chosen as final forecasting model. Although the LSTM model with the name "stacked_large_model" achieved a better scoring, looking at the traininig vs. test performance it appears to overfit, while the smaller model caputures the overall trend better. This is also indicated by the plotted training data prediction.



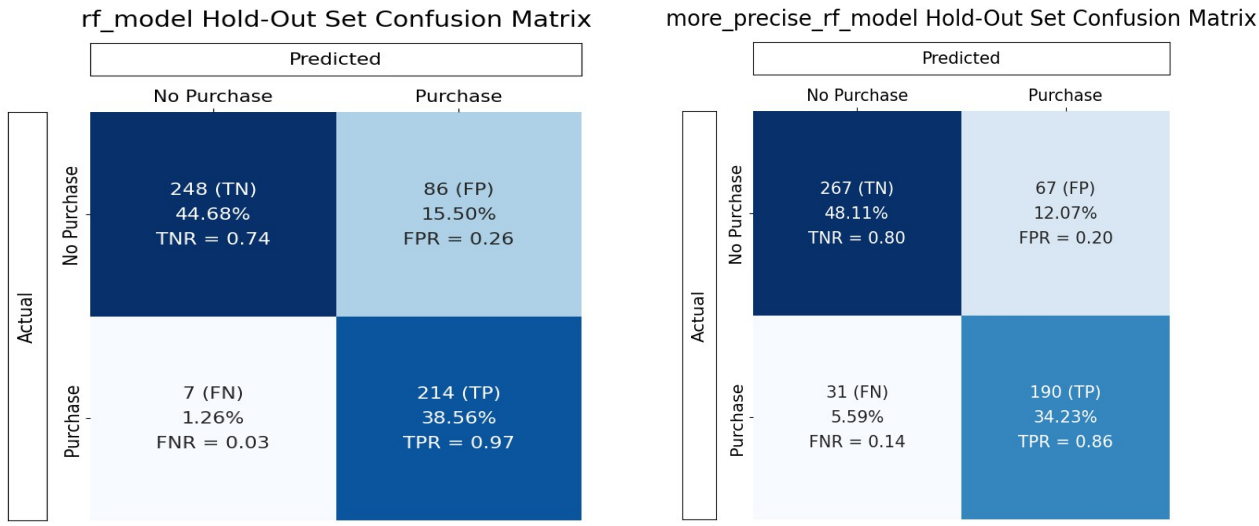


However, it can be observed that both models, in general, underestimate positive peaks and overestimate smaller drops in sales.

2. Model selection for purchase prediction

Score	Mean CV Test Score	Training Score	Test Score
more_precise_lr_model (threshold = 0.58)			
F0.5	0.7827	0.7657	0.7930
Balanced Accuracy	0.8062	0.7830	0.8165
more_precise_rf_model (threshold = 0.67)			
F0.5	0.8037	0.8355	0.7606
Balanced Accuracy	0.8563	0.8732	0.8296
lr_model (threshold = 0.32)			
F1	0.8123	0.8070	0.8137
Balanced Accuracy	0.8236	0.8408	0.8479
rf_model (threshold = 0.58)			
F1	0.8326	0.8345	0.8215
Balanced Accuracy	0.8408	0.8663	0.8554

Looking at the scores of the created purchase prediction models, the random forest classifier outperforms the logisitc regression for both the best F1 (balanced) model and the F0.5 (precision is 2 times as important) score.



Comparing the confusion matrix of the normal "rf_model" with the more precise one we can decide which to pick for the usecase at hand (when to pick which model is exemplified in example_usage.py).

3. Model selection for product recommendatio

	Hit-Rate	Reciprocal Hit-Rate
kNN-IBCF Recommender	0.1337	0.3464

Here only one model was trained, so there are no models to compare.