

# ensemble

May 10, 2024

## 1 Ensemble model

```
[ ]: import os
import pandas as pd
import sys

sys.path.insert(1, "/Users/simon/Documents/II/Dissertation/")
from src.evaluate import get_prediction_dfs_from_experiment, get_all_metrics

%load_ext autoreload
%autoreload 2
```

Original results aggregated by stock

```
[ ]: path = "./results.csv"
if os.path.exists(path):
    dfs = pd.read_csv(path, header=[0, 1], index_col=0)
dfs["Model Type"] = dfs.index.str.split("_").str[0]
dfs["Stock"] = dfs.index.str.split("_").str[1]
orig = dfs.copy()
orig_by_stock = (
    orig.drop(columns=["Hyperparameters", "Model Type"])
    .groupby("Stock")
    .mean()
    .loc[["NVDA", "JPM", "HD", "UNH"]]["Test set"]
)
orig_by_model = (
    orig.drop(columns=["Hyperparameters", "Stock"])
    .groupby("Model Type")
    .mean()
    .loc[["Linear", "ARIMA", "RandomForest", "CNN", "LSTM", "ConvLSTM"]]["Test_
    ↪set"]
)
orig_by_model
```

/var/folders/d7/ktx3dym91yjgj\_gpmnfs0rh00000gn/T/ipykernel\_18903/432629186.py:7:  
PerformanceWarning: dropping on a non-lexsorted multi-index without a level  
parameter may impact performance.

```
orig_by_stock = orig.drop(columns=["Hyperparameters", "Model
Type"]).groupby("Stock").mean().loc[["NVDA", "JPM", "HD", "UNH"]]["Test set"]
/var/folders/d7/ktx3dym91yjgj_gpmnfs0rh00000gn/T/ipykernel_18903/432629186.py:8:
PerformanceWarning: dropping on a non-lexsorted multi-index without a level
parameter may impact performance.
```

```
orig_by_model = orig.drop(columns=["Hyperparameters", "Stock"]).groupby("Model
Type").mean().loc[["Linear", "ARIMA", "RandomForest", "CNN", "LSTM",
"ConvLSTM"]]["Test set"]
```

```
[ ]:
          R2      MSE      RMSE      MAE      p \
Model Type
Linear      -0.01909141 0.00035809 0.01752181 0.01252915 0.02385229
ARIMA       -1.01053333 0.00071815 0.02469201 0.01912823 -0.00055633
RandomForest -0.10110938 0.00042627 0.01860393 0.01317114 -0.00732916
CNN         -0.16178257 0.00038139 0.01834945 0.01319566 -0.04282749
LSTM        -0.00462845 0.00035063 0.01737026 0.01233547 0.02478478
ConvLSTM    -0.00106495 0.00034991 0.01734557 0.01233434 0.00839879
```

```
          Accuracy  Avg. daily return  Std. daily return \
Model Type
Linear           49.10000000           0.00090831           0.01171694
ARIMA            48.80000000           0.00058564           0.01129133
RandomForest     50.30000000           0.00086942           0.01082230
CNN              50.20000000           0.00119086           0.01503570
LSTM             52.70000000           0.00125459           0.01608904
ConvLSTM         55.00000000           0.00175389           0.01711801
```

```
          Risk adj. return
Model Type
Linear           0.05495246
ARIMA            0.04494016
RandomForest     0.07098626
CNN              0.03647862
LSTM             0.04510046
ConvLSTM         0.08239443
```

Ensemble model of positive accuracy learners

```
[ ]: models = ["RandomForest", "CNN", "LSTM", "ConvLSTM"]
      stocks = ["NVDA", "JPM", "HD", "UNH"]

dfs = []
for m in models:
    for s in stocks:
        exp_name = f"{m}_{s}"
        val_df, test_df, hparams = get_prediction_dfs_from_experiment(
            experiment_name=exp_name
        )
```

```

        test_df["Model Type"] = m
        test_df["Stock"] = s
        dfs.append(test_df)
dfs = pd.concat(dfs)
dfs = dfs.drop(columns=["Model Type"]).groupby(["Date", "Stock"]).mean()

metrics = []
for s in stocks:
    df = dfs.xs(s, level="Stock")
    metrics.append(get_all_metrics(df["Predictions"], df["Actuals"]))
metrics = pd.DataFrame(metrics, index=stocks)
metrics.mean()

```

Loading RandomForest\_NVDA.

Rank 1: trial no. 1, value: 50.59760956175299. Run completed at 2024-04-29 17:44:32.475530

Loading RandomForest\_JPM.

Rank 1: trial no. 8, value: 53.38645418326693. Run completed at 2024-04-29 17:51:43.110122

Loading RandomForest\_HD.

Rank 1: trial no. 6, value: 54.18326693227091. Run completed at 2024-04-29 17:56:56.054294

Loading RandomForest\_UNH.

Rank 1: trial no. 3, value: 55.77689243027888. Run completed at 2024-04-29 17:59:37.088842

Loading CNN\_NVDA.

Rank 1: trial no. 11, value: 0.5737051963806152. Run completed at 2024-04-29 20:22:15.814904

Loading CNN\_JPM.

Rank 1: trial no. 9, value: 0.5737051963806152. Run completed at 2024-04-29 18:34:47.643948

Loading CNN\_HD.

Rank 1: trial no. 0, value: 0.5498008131980896. Run completed at 2024-04-29 18:35:17.552823

Loading CNN\_UNH.

Rank 1: trial no. 8, value: 0.5816733241081238. Run completed at 2024-04-29 18:44:16.965952

Loading LSTM\_NVDA.

Rank 1: trial no. 0, value: 0.5537848472595215. Run completed at 2024-04-29 18:01:30.993116

Loading LSTM\_JPM.

Rank 1: trial no. 3, value: 0.6175298690795898. Run completed at 2024-04-29 18:13:38.976207

Loading LSTM\_HD.

Rank 1: trial no. 4, value: 0.6055777072906494. Run completed at 2024-04-29 18:19:41.955427

Loading LSTM\_UNH.

Rank 1: trial no. 11, value: 0.5896414518356323. Run completed at 2024-04-29 20:23:47.878337  
 Loading ConvLSTM\_NVDA.  
 Rank 1: trial no. 2, value: 0.518652081489563. Run completed at 2024-04-29 18:47:06.218729  
 Loading ConvLSTM\_JPM.  
 Rank 1: trial no. 26, value: 0.5697211027145386. Run completed at 2024-04-29 20:35:01.199259  
 Loading ConvLSTM\_HD.  
 Rank 1: trial no. 6, value: 0.5231999158859253. Run completed at 2024-04-29 19:04:03.502803  
 Loading ConvLSTM\_UNH.  
 Rank 1: trial no. 0, value: 0.5577689409255981. Run completed at 2024-04-29 16:26:05.723444

```
[ ]: R2                -0.02602749
      MSE                0.00036070
      RMSE               0.01758301
      MAE                0.01253625
      p                 -0.03972312
      Accuracy          51.50000000
      Avg. daily return  0.00091445
      Std. daily return  0.01339927
      Risk adj. return   0.04897786
      dtype: float64
```

Ensemble of all learners

```
[ ]: models = ["Linear", "ARIMA", "RandomForest", "CNN", "LSTM", "ConvLSTM"]
      stocks = ["NVDA", "JPM", "HD", "UNH"]

      dfs = []
      for m in models:
          for s in stocks:
              exp_name = f"{m}_{s}"
              val_df, test_df, hparams = get_prediction_dfs_from_experiment(
                  experiment_name=exp_name
              )
              test_df["Model Type"] = m
              test_df["Stock"] = s
              dfs.append(test_df)
      dfs = pd.concat(dfs)
      dfs = dfs.drop(columns=["Model Type"]).groupby(["Date", "Stock"]).mean()

      metrics = []
      for s in stocks:
          df = dfs.xs(s, level="Stock")
          metrics.append(get_all_metrics(df["Predictions"], df["Actuals"]))
```

```
metrics = pd.DataFrame(metrics, index=stocks)
metrics.mean()
```

Loading Linear\_NVDA.

Rank 1: trial no. 0, value: 45.0199203187251. Run completed at 2024-04-29  
16:52:25.570862

Loading Linear\_JPM.

Rank 1: trial no. 0, value: 46.613545816733065. Run completed at 2024-04-29  
16:52:27.015066

Loading Linear\_HD.

Rank 1: trial no. 0, value: 52.589641434262944. Run completed at 2024-04-29  
16:52:28.379346

Loading Linear\_UNH.

Rank 1: trial no. 0, value: 47.808764940239044. Run completed at 2024-04-29  
16:52:29.689755

Loading ARIMA\_NVDA.

Rank 1: trial no. 18, value: 56.97211155378486. Run completed at 2024-04-29  
17:11:04.256046

Loading ARIMA\_JPM.

Rank 1: trial no. 1, value: 52.98804780876494. Run completed at 2024-04-29  
17:15:08.552541

Loading ARIMA\_HD.

Rank 1: trial no. 4, value: 52.98804780876494. Run completed at 2024-04-29  
17:27:20.232495

Loading ARIMA\_UNH.

Rank 1: trial no. 1, value: 47.01195219123506. Run completed at 2024-04-29  
17:35:18.839560

Loading RandomForest\_NVDA.

Rank 1: trial no. 1, value: 50.59760956175299. Run completed at 2024-04-29  
17:44:32.475530

Loading RandomForest\_JPM.

Rank 1: trial no. 8, value: 53.38645418326693. Run completed at 2024-04-29  
17:51:43.110122

Loading RandomForest\_HD.

Rank 1: trial no. 6, value: 54.18326693227091. Run completed at 2024-04-29  
17:56:56.054294

Loading RandomForest\_UNH.

Rank 1: trial no. 3, value: 55.77689243027888. Run completed at 2024-04-29  
17:59:37.088842

Loading CNN\_NVDA.

Rank 1: trial no. 11, value: 0.5737051963806152. Run completed at 2024-04-29  
20:22:15.814904

Loading CNN\_JPM.

Rank 1: trial no. 9, value: 0.5737051963806152. Run completed at 2024-04-29  
18:34:47.643948

Loading CNN\_HD.

Rank 1: trial no. 0, value: 0.5498008131980896. Run completed at 2024-04-29

```

18:35:17.552823
Loading CNN_UNH.
Rank 1: trial no. 8, value: 0.5816733241081238. Run completed at 2024-04-29
18:44:16.965952
Loading LSTM_NVDA.
Rank 1: trial no. 0, value: 0.5537848472595215. Run completed at 2024-04-29
18:01:30.993116
Loading LSTM_JPM.
Rank 1: trial no. 3, value: 0.6175298690795898. Run completed at 2024-04-29
18:13:38.976207
Loading LSTM_HD.
Rank 1: trial no. 4, value: 0.6055777072906494. Run completed at 2024-04-29
18:19:41.955427
Loading LSTM_UNH.
Rank 1: trial no. 11, value: 0.5896414518356323. Run completed at 2024-04-29
20:23:47.878337
Loading ConvLSTM_NVDA.
Rank 1: trial no. 2, value: 0.518652081489563. Run completed at 2024-04-29
18:47:06.218729
Loading ConvLSTM_JPM.
Rank 1: trial no. 26, value: 0.5697211027145386. Run completed at 2024-04-29
20:35:01.199259
Loading ConvLSTM_HD.
Rank 1: trial no. 6, value: 0.5231999158859253. Run completed at 2024-04-29
19:04:03.502803
Loading ConvLSTM_UNH.
Rank 1: trial no. 0, value: 0.5577689409255981. Run completed at 2024-04-29
16:26:05.723444

```

```

[ ]: R2                -0.03926330
     MSE                0.00036640
     RMSE               0.01770864
     MAE                0.01272426
     p                 -0.01848382
     Accuracy          48.70000000
     Avg. daily return  0.00066838
     Std. daily return  0.01147748
     Risk adj. return   0.04684723
     dtype: float64

```

Ensemble of deep learning models

```

[ ]: models = ["CNN", "LSTM", "ConvLSTM"]
     stocks = ["NVDA", "JPM", "HD", "UNH"]

     dfs = []
     for m in models:
         for s in stocks:

```

```

exp_name = f"{m}_{s}"
val_df, test_df, hparams = get_prediction_dfs_from_experiment(
    experiment_name=exp_name
)
test_df["Model Type"] = m
test_df["Stock"] = s
dfs.append(test_df)
dfs = pd.concat(dfs)
dfs = dfs.drop(columns=["Model Type"]).groupby(["Date", "Stock"]).mean()

metrics = []
for s in stocks:
    df = dfs.xs(s, level="Stock")
    metrics.append(get_all_metrics(df["Predictions"], df["Actuals"]))
metrics = pd.DataFrame(metrics, index=stocks)
metrics.mean()

```

Loading CNN\_NVDA.

Rank 1: trial no. 11, value: 0.5737051963806152. Run completed at 2024-04-29 20:22:15.814904

Loading CNN\_JPM.

Rank 1: trial no. 9, value: 0.5737051963806152. Run completed at 2024-04-29 18:34:47.643948

Loading CNN\_HD.

Rank 1: trial no. 0, value: 0.5498008131980896. Run completed at 2024-04-29 18:35:17.552823

Loading CNN\_UNH.

Rank 1: trial no. 8, value: 0.5816733241081238. Run completed at 2024-04-29 18:44:16.965952

Loading LSTM\_NVDA.

Rank 1: trial no. 0, value: 0.5537848472595215. Run completed at 2024-04-29 18:01:30.993116

Loading LSTM\_JPM.

Rank 1: trial no. 3, value: 0.6175298690795898. Run completed at 2024-04-29 18:13:38.976207

Loading LSTM\_HD.

Rank 1: trial no. 4, value: 0.6055777072906494. Run completed at 2024-04-29 18:19:41.955427

Loading LSTM\_UNH.

Rank 1: trial no. 11, value: 0.5896414518356323. Run completed at 2024-04-29 20:23:47.878337

Loading ConvLSTM\_NVDA.

Rank 1: trial no. 2, value: 0.518652081489563. Run completed at 2024-04-29 18:47:06.218729

Loading ConvLSTM\_JPM.

Rank 1: trial no. 26, value: 0.5697211027145386. Run completed at 2024-04-29 20:35:01.199259

Loading ConvLSTM\_HD.

Rank 1: trial no. 6, value: 0.5231999158859253. Run completed at 2024-04-29 19:04:03.502803

Loading ConvLSTM\_UNH.

Rank 1: trial no. 0, value: 0.5577689409255981. Run completed at 2024-04-29 16:26:05.723444

```
[ ]: R2                -0.02423907
      MSE                0.00035377
      RMSE              0.01749355
      MAE               0.01245824
      P                 -0.01024205
      Accuracy          52.00000000
      Avg. daily return 0.00138459
      Std. daily return 0.01606900
      Risk adj. return  0.05566150
      dtype: float64
```

```
[ ]: models = ["CNN", "LSTM"]
      stocks = ["NVDA", "JPM", "HD", "UNH"]

      dfs = []
      for m in models:
          for s in stocks:
              exp_name = f"{m}_{s}"
              val_df, test_df, hparams = get_prediction_dfs_from_experiment(
                  experiment_name=exp_name
              )
              test_df["Model Type"] = m
              test_df["Stock"] = s
              dfs.append(test_df)
      dfs = pd.concat(dfs)
      dfs = dfs.drop(columns=["Model Type"]).groupby(["Date", "Stock"]).mean()

      metrics = []
      for s in stocks:
          df = dfs.xs(s, level="Stock")
          metrics.append(get_all_metrics(df["Predictions"], df["Actuals"]))
      metrics = pd.DataFrame(metrics, index=stocks)
      metrics.mean()
```

Loading CNN\_NVDA.

Rank 1: trial no. 11, value: 0.5737051963806152. Run completed at 2024-04-29 20:22:15.814904

Loading CNN\_JPM.

Rank 1: trial no. 9, value: 0.5737051963806152. Run completed at 2024-04-29 18:34:47.643948

Loading CNN\_HD.



Rank 1: trial no. 0, value: 0.5498008131980896. Run completed at 2024-04-29 18:35:17.552823  
Loading CNN\_UNH.  
Rank 1: trial no. 8, value: 0.5816733241081238. Run completed at 2024-04-29 18:44:16.965952  
Loading LSTM\_NVDA.  
Rank 1: trial no. 0, value: 0.5537848472595215. Run completed at 2024-04-29 18:01:30.993116  
Loading LSTM\_JPM.  
Rank 1: trial no. 3, value: 0.6175298690795898. Run completed at 2024-04-29 18:13:38.976207  
Loading LSTM\_HD.  
Rank 1: trial no. 4, value: 0.6055777072906494. Run completed at 2024-04-29 18:19:41.955427  
Loading LSTM\_UNH.  
Rank 1: trial no. 11, value: 0.5896414518356323. Run completed at 2024-04-29 20:23:47.878337

```
[ ]: R2                -0.04951123
      MSE                0.00035996
      RMSE               0.01767374
      MAE                0.01262452
      p                 -0.01063027
      Accuracy          50.60000000
      Avg. daily return  0.00120291
      Std. daily return  0.01558573
      Risk adj. return   0.04666328
      dtype: float64
```