Training a Random Forest

MGMT 638: Data-Driven Investments: Equity

Kerry Back, Rice University

Outline

- Suppose we're happy with our random forest model.
- Then, we should train it using all of our data and save the trained model.
- We can load the trained model and use it to make predictions whenever we want.

Read data

In [6]:

```
import pandas as pd

# change path_to_file to "./" if the file is in your working directory
path_to_file = "../../"

df = pd.read_csv(path_to_file + "data-2023-11-08.csv")
df.head()
```

Out[6]:

	ticker	date	marketcap	pb	ret	mom	volume	volatility
0	AACC	2011-01-14	188.3	1.4	-0.014634	-0.184615	2.078000e+04	0.071498
1	AAI	2011-01-14	1012.1	2.0	0.002677	0.438224	2.775580e+06	0.128450
2	AAIC	2011-01-14	189.3	1.0	-0.010119	0.684547	3.466000e+04	0.048505
3	AAON	2011-01-14	479.4	4.2	0.007778	0.528685	2.817291e+05	0.044912
4	AATC	2011-01-14	63.3	1.4	-0.013960	0.008216	6.800000e+03	0.049756

Define model and target variable

In [7]:

```python
from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor(max_depth=3)

df["target"] = df.groupby("date", group_keys=False).ret.apply(
    lambda x: x - x.median()
)
```

Define predictors (features)

In [8]:

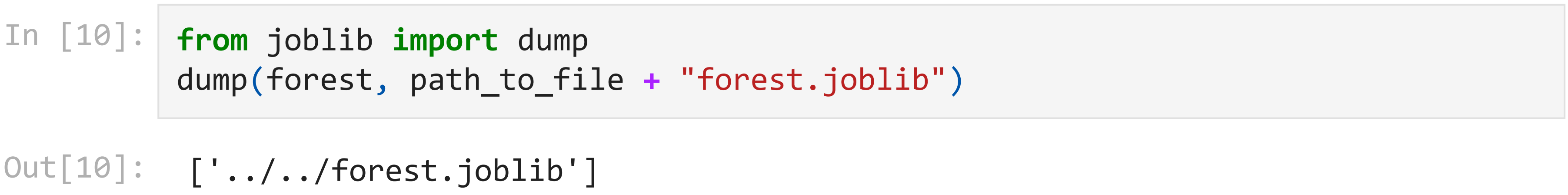
```
features = [
    "marketcap",
    "pb",
    "mom",
    "volume",
    "volatility",
    "roe",
    "accruals"
]
```

Train the model

```
In [9]: forest.fit(X=df[features], y=df.target)
```

```
Out[9]: ▾ RandomForestRegressor
        RandomForestRegressor(max_depth=3)
```

Save the model



```
In [10]:  from joblib import dump
          dump(forest, path_to_file + "forest.joblib")

Out[10]:  ['../../forest.joblib']
```