IEOR 242A: Machine Learning and Data Analytics I, Fall 2025

Homework Assignment #4

Due Friday, November 21 at 11:59pm

Problem 1: Bootstrapping for model validation (40 points)

Consider the following synthetic linear model that represents the *ground truth* data generation process of (X, Y):

$$\begin{aligned} &X \sim \mathcal{N}(0,1), \\ &Y \mid X = 1 + 2.5X + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0,1), \end{aligned}$$

where $\mathcal{N}(0,1)$ denotes the standard normal distribution. Fix a random seed at 242.

a) (30 points) Generate a fixed synthetic dataset with $n = 300$ with 80/20 train-test split from the ground truth process. Generate $B = 2000$ bootstraps on the *test* data. Fit an OLS linear regression, compute the *empirical test* OSR^2 using your original test data, and use the resampled test sets to estimate a series of OSR^2. Then fit a default Random Forest regressor, compute the empirical test OSR^2, and estimate the series of 2000 bootstrapped OSR^2 (using the same resampled test data). Use `r2_score` in `sklearn.metrics` for computing the OSR^2.

We denote the population (unseen), empirical (point estimation), and bootstrap-estimated test OSR^2 for methods (OLS and RF) by the following:

$$\begin{aligned} &\theta^{*,\mathrm{OLS}}, \quad \widehat{\theta}^{\mathrm{OLS}}, \quad \{\widehat{\theta}_i^{*,\mathrm{OLS}}\}_{i=1}^{B} \\ &\theta^{*,\mathrm{RF}}, \quad \widehat{\theta}^{\mathrm{RF}}, \quad \{\widehat{\theta}_i^{*,\mathrm{RF}}\}_{i=1}^{B} \end{aligned}$$

For both methods (OLS and RF), estimate the (1) **bias and variance** of the empirical test $\widehat{\theta}$ from the bootstrapped error series (see Lecture 17, Page 20), and (2) **95% bootstrap confidence interval** for the *population test* θ^* (see Lecture 17, Page 28).

Visualize the **histograms** of (1) the series of bootstrapped OSR^2: $\{\widehat{\theta}_i^{*,\mathrm{OLS}}\}_{i=1}^{B}$, $\{\widehat{\theta}_i^{*,\mathrm{RF}}\}_{i=1}^{B}$, and (2) the difference $\{\widehat{\theta}_i^{*,\mathrm{OLS}} - \widehat{\theta}_i^{*,\mathrm{RF}}\}_{i=1}^{B}$

Construct the **95% bootstrap confidence interval** for the difference and **argue why or why not the two methods perform statistically significantly different on this data**. Notice that to use the formula on Lecture 17, Page 28, you need to let $\widehat{\theta}$ be the difference of point estimations of test OSR^2, i.e., $\widehat{\theta} := \widehat{\theta}^{\mathrm{OLS}} - \widehat{\theta}^{\mathrm{RF}}$ and find the quantiles of $\{(\widehat{\theta}_i^{*,\mathrm{OLS}} - \widehat{\theta}_i^{*,\mathrm{RF}}) - \widehat{\theta}\}_{i=1}^{B}$.

b) (10 points) Re-generate a dataset with $n = 5000$ and re-do the above estimations. Does your conclusion regarding the the difference of the two models' test performances change? Reason which of the two conclusions we should trust better.

Problem 2: Predicting Air Passengers (60 points)

The AirPassenger time series dataset contains the monthly totals of international airline passengers from 1949 to 1960. The dataset consists of two columns, where the first column provides the month information, and the second column represents the number of international airline passengers in thousands in that month. The data for this problem is contained in the files **`AirPassengers.csv`**, and it is a built-in dataset available in R. Use the last 2-year's data (i.e., 24 time periods) as the test data and split it from the training data. For R^2 and OSR^2, use `r2_score` in `sklearn.metrics` for the computations.

a) (10 points) Visualize the time series and discuss whether you observe any (1) trend, (2) seasonality, and (3) stationarity. Reason why or why not you believe the time series appear stationary or non-stationary.

b) (10 points) Fit a *linear trend* model and an AR(1) model. Report the training R^2, test OSR^2, and plot the entire fitted time series versus the actual ones. What is the expected yearly change of the total number of international airline passengers? Discuss reasons why or why not the AR(1) model is not performing well.

c) (15 points) Perform feature engineering to (1) visualize the average sales by months and incorporate necessary seasonal features; (2) get inspired from the linear trend model and incorporate the trend feature; and (3) incorporate good candidates of auto-regressive variables of proper order (*hint*: why we may want to shift by 12?).

Write down the linear model in a form similar to the following:

$$\texttt{Passengers}_t = \beta_0 + \beta_1 \texttt{TimePeriod}_t + \sum_i \beta_i \texttt{Passengers}_{t-p_i} + \cdots + \beta_2 \mathbf{1}\{\text{month=Dec.}\} + \cdots + \epsilon_t,$$

and fit this custom auto-regressive model.

Perform feature selections by criteria such as p-value over your candidate features, and state your final linear model. Report the training R^2, test OSR^2, and plot the entire fitted time series versus the actual ones.

d) (10 points) Use the same set of features you constructed in part (c), train a random forest model with the parameter `n_estimators` $\in [10, 50, 100, 200]$ selected by cross validation with folds obtained from the default `TimeSeriesSplit` (from `sklearn.model_selection`). Use OSR^2 as the CV criteria. Plot the time series predicted by the random forest, and compare the prediction with your results in (c).

e) (15 points) We explore stationarity in this part further. There is a hypothesis test called the *Dickey-Fuller test* to verify if a time series is stationary or not. This test is implemented in `statsmodels` and you can use the code shown in Figure 1 to diagnose for an arbitrary time series whether it is stationary or not. Using the code, visualize the following time series and diagnose if each of them is stationary or not:

- (original series) $\texttt{Passengers}_t$
- (differencing) $\Delta\text{P}_t = \texttt{Passengers}_t - \texttt{Passengers}_{t-1}$

- (seasonal-differencing) $\Delta_{12}\mathrm{P}_t = \texttt{Passengers}_t - \texttt{Passengers}_{t-12}$

  You may use `df['Passengers'].diff(h)` in Python to obtain a differencing with a h-period shift.

Fit an AR(1) model for the seasonal-differencing series. Plot the fitted difference series versus the actual difference series, and comment on the model performance.

```
from statsmodels.tsa.stattools import adfuller

def adf_report(series, name='Series'):
    s = series.dropna().values
    result = adfuller(s, autolag='AIC')
    print(f'ADF test on "{name}":')
    print(f'  Test Statistic: {result[0]:.4f}')
    print(f'  p-value       : {result[1]:.4f}')
    print(f'  #Lags Used    : {result[2]}')
    print(f'  #Observations : {result[3]}')
    print('  Critical Values:')
    for k, v in result[4].items():
        print(f'    {k}: {v:.4f}')
    print('  => Stationary?' , 'Yes (reject H0)' if result[1] < 0.05 else 'No (fail to reject H0)')
    print()
```

Figure 1: ADF stationarity test.