



به نام خدا



1928

K. N. Toosi University of Technology

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده برق

شبیه سازی و مدلسازی

گزارش تمرین شماره 2

سجاد رجبی باغستان

40005393

استاد : جناب دکتر مهدی علیاری

بهمن 1402

## فهرست مطالب

عنوان	شماره صفحه
بخش ۱: چکیده .....	3
بخش ۱: سوالات تحلیلی .....	4
سوال اول .....	4
بخش ۲: سوالات شبیه سازی .....	5
سوال اول .....	5
سوال دوم .....	6
سوال سوم .....	7
مراجع .....	11

## چکیده

در این تمرین، به بررسی و مقایسه دو روش انتخاب ویژگی Forward Selection و backward elimination میپردازیم.

**Forward Selection** یک روش انتخاب ویژگی‌های مهم در مدل‌های رگرسیون است. در این روش، از یک مدل اولیه شروع می‌شود که فقط یک ویژگی دارد. سپس ویژگی‌های دیگر به تدریج اضافه می‌شوند و هر بار اثر اضافه شدن یک ویژگی را ارزیابی می‌کنیم. ویژگی‌ای که اضافه شدن آن باعث بهبود بیشتری در عملکرد مدل می‌شود، به مدل اضافه می‌شود.

**Backward Elimination** یک روش انتخاب ویژگی در مدل‌های رگرسیون است که به تدریج ویژگی‌های غیرضروری را از مدل حذف می‌کند تا به بهترین مدل برسد. در این روش، از یک مدل اولیه که شامل تمام ویژگی‌ها است شروع می‌شود، سپس ویژگی‌های غیرضروری یکی‌یکی حذف می‌شوند و عملکرد مدل با هر حذف ارزیابی می‌شود.

مزایای این روش شامل کاهش ابعاد داده‌ها، سرعت بالا در مواقعی که تعداد ویژگی‌ها زیاد است، و قابلیت استفاده از معیارهای مختلف برای ارزیابی مدل است. از معایب آن می‌توان به این اشاره کرد که ممکن است در مواقعی ویژگی‌های مهم حذف شوند و همچنین این روش بیشتر در مواقعی مؤثر است که تأثیرات ویژگی‌ها به طور مستقل از هم نباشند.

## بخش ۱: سوالات تحلیلی

### سوال اول

در بحث مدل سازی ،هدف این است که مدلی ارائه شود که نه بیش از حد بی دقت باشد و نه بیش از حد با دقت (اصطلاحاً  $overfit$  یا  $underfit$  نباشد). با داشتن رگرسور های فراوان برای مدل سازی ،مدل ارائه شده به صورت خیلی دقیق عمل میکند به طوری که مدل ما  $overfit$  میشود ، همچنین با داشتن تعداد زیادی رگرسور ، مقدار زیادی نویز وارد مدل ما میشود زیرا هر رگرسور همراه با نویز است. در ضمن این موضوع را باید در نظر داشت که ممکن است اثر یک رگرسور نسبت به بقیه رگرسور ها خیلی بیشتر باشد و همین امر باعث این میشود که استفاده از بقیه رگرسور ها ناصرفه باشد.

### دلایل ضرورت subsection:

- به طور کلی در واقعیت تعداد رگرسور ها میتواند بسیار زیاد باشد و همین امر باعث پیچیدگی زیاد مدل از نظر محاسباتی شود به طوری که انجام این محاسبات ناممکن و یا بسیار زمان بر است.
- در بسیار مواقع سیستم تنها از تعداد محدودی رگرسور تاثیر اصلی را میپذیرد و بقیه رگرسور ها با توجه به هزینه ای که برای مدل سازی صرف میشود (زمان، محاسبات، تجهیزات،...) غیر ضروری هستند.

### فواید subsection:

- کاهش زمان و منابع
- بهتر شدن مدل و دور شدن مدل از  $overfitting$

## بخش ۲: سوالات شبیه سازی

### سوال اول

در این سوال بعد از import کردن دیتا ها ، رگرسور ها را یک به یک انتخاب کرده و اضافه میکنیم  
این کار را تا زمانی ادامه میدهیم که کمترین تغییرات را برای ساخت  $y$  پیدا کنیم.  
رگرسور هایی که انتخاب میشوند بیشترین کواریانس را با  $y$  دارند.

### Result:

Selected Regressors:

1 2 3 4

Parameters:

0.1719

0.2174

0.3016

0.2892

>>

## سوال دوم

در این سوال بعد از import کردن دیتا ها ، رگرسور ها را یک به یک انتخاب کرده و حذف میکنیم  
این کار را تازمانی ادامه میدهیم که بیشترین تغییرات را در  $y$  مشاهده کنیم.

### Result:

Selected Regressors:

1	2	3	4
---	---	---	---

Parameters:

0.1719	0.2174	0.3016	0.2892
--------	--------	--------	--------

>>

## سوال سوم

جواب نهایی در هردو روش یکسان است .

تفاوت در سرعت پاسخگویی:

نکته ای که این دو روش را از هم متمایز میکند الگوریتم آن است که باعث تغییر در سرعت اجرای کد و انجام محاسبات میشود .

Forward Selection: Elapsed time is 0.012016 seconds.

Backward Elimination: Elapsed time is 0.005213 seconds.

با توجه به مقادیر به دست آمده در رابطه با زمان مورد نیاز برای انجام محاسبات نتیجه گیری میشود که زمان اجرا روش Forward Selection تقریباً دو برابر Backward Elimination است و این موضوع زمانی بسیار مهم می شود که حجم دیتا ستی که در اختیار ما است بسیار زیاد باشد، در این صورت احتمالاً ترجیح بر این است که از روش کوتاه تر استفاده شود .

تفاوت در سرعت اجرا بین روش های Forward Selection و Backward Elimination ممکن است به دلایل زیر باشد:

- انتخاب ویژگی ها: در روش Forward Selection، ویژگی هایی که به مدل اضافه می شوند، بر اساس بهبودی که در عملکرد مدل به دنبال اضافه شدن هر ویژگی حاصل می شود، انتخاب می شوند. این می تواند منجر به انتخاب سریع تر ویژگی های مهم تر و کارا باشد.
  - حذف ویژگی ها: در روش Backward Elimination، ما به تدریج ویژگی های غیر ضروری را از مدل حذف می کنیم. این عملیات حذف ممکن است در مواردی زمان برتر باشد، مخصوصاً اگر تعداد ویژگی ها زیاد باشد.
- در کل، اگرچه هر دو روش می توانند به بهبود عملکرد مدل منجر شوند، اما عملکرد و سرعت آنها ممکن است به ویژگی های داده، اندازه داده و پیچیدگی مدل بستگی داشته باشد.

## تفاوت در صحت پاسخگویی:

هرچند که در این آزمایش مشخص نشد اما با توجه به منابع موجود و تجربه های دیگر نتایج این دو روش در گاهی اوقات متفاوت به دست می آید به این صورت که روش Forward Selection با خطا کمتر و صحت بیشتری نسبت به روش Backward Elimination عمل میکند.

احتمال خطا در روش Backward Elimination نسبت به Forward Selection ممکن است به دلایل زیر بیشتر باشد:

- شناسایی تأثیرات متقاطع: در بعضی موارد، ویژگی‌ها ممکن است با هم تعامل داشته باشند و حذف یک ویژگی می‌تواند تأثیرات متقاطع دیگر ویژگی‌ها را تغییر دهد. این تعاملات ممکن است توسط روش Backward Elimination در نظر گرفته نشوند که باعث ایجاد خطاهای نادرست در انتخاب و حذف ویژگی‌ها می‌شود.
- انتخاب متغیرهای نادرست: در صورتی که یک ویژگی به طور نادرست از مدل حذف شود، ممکن است اطلاعات مهمی را که مدل باید به طور صحیح استفاده کند، از دست بدهیم. این موضوع می‌تواند به افزایش خطا در پیش‌بینی‌های مدل منجر شود.
- استقرار متغیرهای تصادفی: در صورتی که متغیرهایی به طور تصادفی از مدل حذف شوند بدون در نظر گرفتن اثرات واقعی آن‌ها، این می‌تواند به خطاهای نادرست در مدل منجر شود.

به طور کلی، عدم در نظر گرفتن اثرات متقاطع و تداخلات بین متغیرها و انجام حذف‌ها به طور نادرست ممکن است باعث افزایش احتمال خطا در روش Backward Elimination شود. از طرفی، روش Forward Selection به طور تدریجی ویژگی‌های مهم را به مدل اضافه می‌کند و این می‌تواند باعث کاهش احتمال خطا در انتخاب متغیرها شود.



## تفاوت در الگوریتم:

### Forward Selection

- 1- شروع با ویژگی خالی: در این روش، مدل با یک ویژگی شروع می‌شود و سپس ویژگی‌های دیگر به ترتیب اضافه می‌شوند تا به بهترین مدل برسیم.
- 2- افزودن ویژگی‌ها: ویژگی‌هایی که با اضافه شدن به مدل، منجر به بهبود عملکرد مدل می‌شوند، به مدل اضافه می‌شوند.

### Backward Elimination

- 1- شروع با تمام ویژگی‌ها: در این روش، مدل با تمام ویژگی‌ها شروع می‌شود و سپس ویژگی‌های غیرضروری به تدریج حذف می‌شوند.
- 2- حذف ویژگی‌ها: ویژگی‌هایی که موجب کاهش خطای مدل نمی‌شوند، یکی یکی از مدل حذف می‌شوند تا به بهترین مدل برسیم.

خلاصه نتایج:

ویژگی	Backward Elimination	Forward Selection
انتخاب اولیه رگرسور	شروع با تمام ویژگی‌ها	شروع با یک ویژگی
افزایش ویژگی‌ها	حذف ویژگی‌هایی که عملکرد را بهبود نمی‌دهند	اضافه کردن ویژگی‌هایی که به عملکرد مدل کمک می‌کنند
تعداد مراحل	بیشتر	کمتر
زمان اجرا	ممکن است کمتر باشد	ممکن است بیشتر باشد
احتمال خطا	بیشتر	کمتر
حساسیت به تعداد ویژگی‌ها	کمتر	بیشتر

جدول 1: مقایسه Forward Selection و backward elimination

Websites:

<https://chat.openai.com/>

<https://en.wikipedia.org>

<https://www.sciencedirect.com/topics/computer-science>

<https://www.simplilearn.com>

book:

Nonlinear system identification 2020 nelles