

بسم الله الرحمن الرحيم



گزارش پروژه درس داده کاوی

مینا بیرامی

راضیه زمانی

فهرست

۲	پژوهش های انجام شده در این پروژه:
۲	موضوع پروژه:
۲	مراحل پژوهش:
۲	مرحله اول
۳	مرحله دوم:
۳	مرحله سوم
۴	مرحله چهارم
۶	مرحله پنجم
۸	نتیجه گیری:

پژوهش های انجام شده در این پروژه:

در این پژوهش جهت پیش بینی کوتاه مدت میزان بار مصرفی از تکنیک های دسته بندی مختلفی از استفاده شده است که عبارتند از :

- تکنیک های ماشین بردار پشتیبان

SVM Regression (SVR)

- الگوریتم درخت تصمیم

Decision Tree

- تکنیک های شبکه عصبی

شبکه عصبی MLP

موضوع پروژه:

در این پژوهش داده ای مربوط به نیمه دوم سال ۹۳ تا نیمه اول سال ۹۴ به عنوان داده ها آزمایشی به دسته بند های فوق

داده شده و در نهایت یک روز را به عنوان داده آزمایشی به مدل تهیه شده اعمال نموده و بار مصرفی پیش بینی شده را

مورد ارزیابی قرار می دهیم.

مراحل پژوهش:

مرحله اول: انتخاب دیتا ست مناسب و زبان برنامه نویسی

دیتا ست همان گونه که در موضوع پروژه گفته شد، داده های نیمه دوم سال ۹۳ تا نیمه اول ۹۴ است که میزان بار برق مصرفی در

یک روز را مشخص می کند.

زبان برنامه نویسی پایتون است که مشخصاً از کتابخانه های مخصوص داده کاوی این زبان به نام sklearn استفاده شده است.

این کتابخانه یک مرجع تقریباً کامل برای تمام متود های استفاده شده در machin learning و data minig است .

۳ فایل با پسوند ipynb موجود است که نام آنها متناسب با متودی که مورد بررسی قرار گرفته است گذاشته شده است.

مرحله دوم: استاندارد کردن و scale کردن دیتاست

در مرحله ی preprocessing هر دیتاستی می بایست ویژگی های دیتاست را با یکدیگر متناسب کرد. از این با یکی از کتابخانه های sklearn به نام StandardScaler استفاده شده است. با بهره گیری از pca و gussian NB (پارامترهای تعریف شده در standar scaler) به مقایسه آن ها زمانی که scale شده و زمانی که scale نشده است، پرداختیم. نتایج در شکل زیر مشاهده میشود. که scale کردن با standard scaler به درد ما نمیخورد. این بدین معنا نیست که نباید دیتاست را scale کنیم. بلکه standard scaler برای دیتاست ما مناسب نیست. شاید متود های دیگر برای آن مناسب باشد.

Prediction accuracy for the normal test dataset with PCA
3.43%

Prediction accuracy for the standardized test dataset with PCA
2.84%

مرحله سوم: به کارگیری svm

از کتابخانه SVR که برای regression نوشته شده است، استفاده می کنیم. چرا که هدف ما پیدا کردن تعداد کلاس مشخصی نیست. ما میخواهیم با توجه به ویژگی های یک روز بخصوص میزان برق مصرفی آنکه عددی در یک بازه طولانی است را پیش بینی کنیم. برای بررسی میزان خوب بودن این متود از r^2 -score استفاده می کنیم:

$$\begin{aligned}SS_{\text{reg}} &= \sum_i (f_i - \bar{y})^2, \\SS_{\text{tot}} &= \sum_i (y_i - \bar{y})^2, \\SS_{\text{res}} &= \sum_i (y_i - f_i)^2 = \sum_i e_i^2 \\SS_{\text{res}} + SS_{\text{reg}} &= SS_{\text{tot}}. \\R^2 &= \frac{SS_{\text{reg}}}{SS_{\text{tot}}} = \frac{SS_{\text{reg}}/n}{SS_{\text{tot}}/n}\end{aligned}$$

نتیجه برای دیتا ست ما به شکل زیر درآمد:

```
1 #print(metrics.precision_score(Ytest,predictions, average='macro'))
2 print(r2_score(Ytest, predictions,multioutput='variance_weighted'))
```

0.0625416260226

۰.۰۶ عدد بسیار پایینی است. از این رو SVM را برای این دیتا ست مناسب ندانستیم و به سراغ متود بعدی رفتیم.

مرحله چهارم: استفاده از درخت تصمیم

ویژگی های دیتا ست به شکل زیر است. هدف پیش بینی sum است.

Sesion Month DayOfNumber DayOfWeek OnOffDay Hour Sum

Sesion: فصل

month: ماه

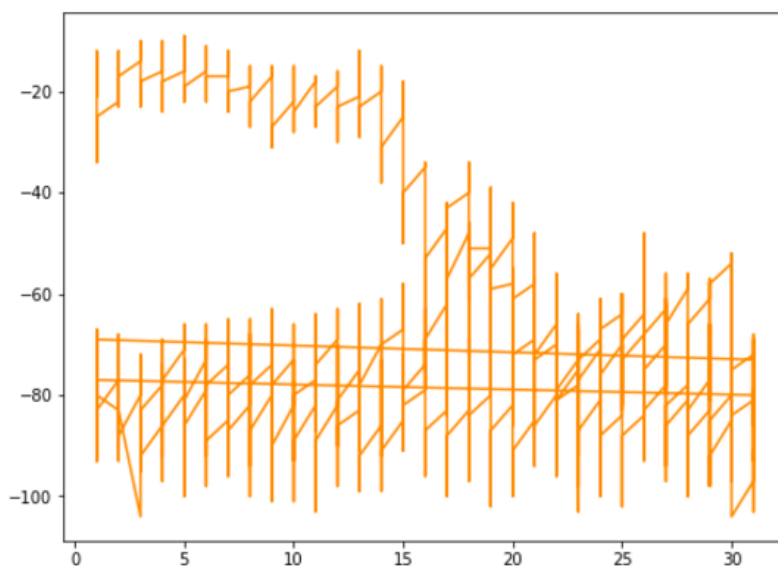
DayOFNumber: شماره روز در ماه (از ۱ تا ۳۱)

DayOfWeek: شماره روز در هفته (از ۱ تا ۷)

onOffDay: روز کاری یا روز تعطیل

hour: ساعت (از ۱ تا ۲۴)

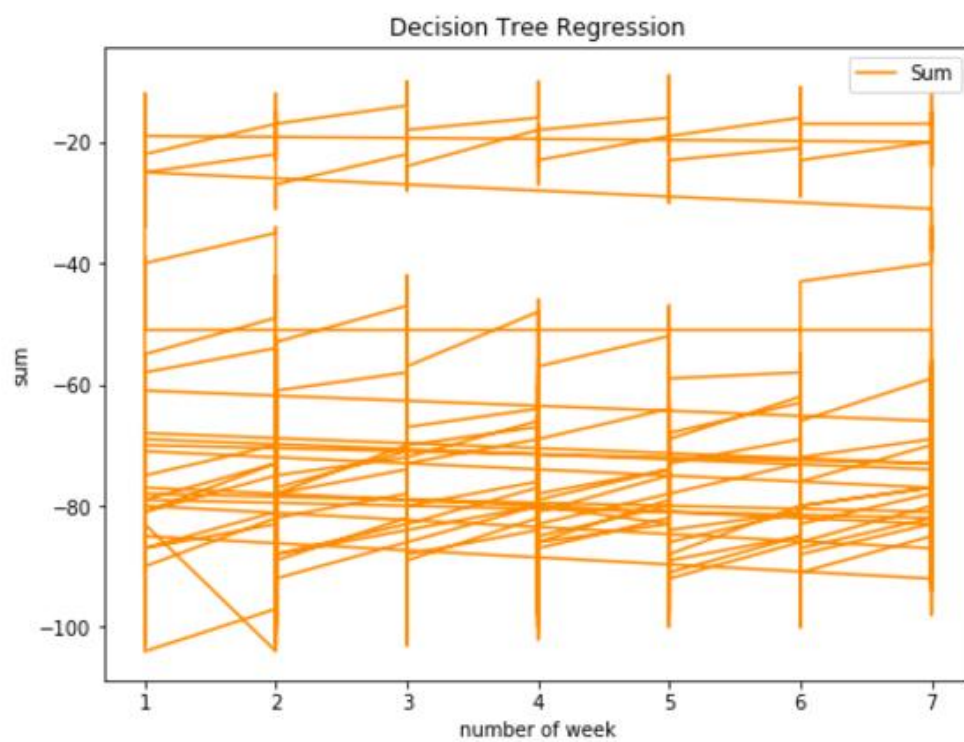
برای پیدا کردن یک نمای بهتر نسبت به دیتا ست چند نمودار برای آن رسم کردیم تا دید شهودی نسبت به آن بدست آوریم:



رسم توضیحی نمودار جمع بار مصرفی در روز های ماه



رسم توضیحی ۳ میزان مصرفی بر حسب ساعت



رسم توضیحی ۲ میزان مصرفی بر حسب روز های هفته

با توجه به نمودار های بالا پیچیدگی دیتا ست را درک کردیم. حال برای آن از کتابخانه DecisionTreeRegressor استفاده می کنیم و r2-score آن را محاسبه می کنیم.

```
1 print(r2_score(Ytest, treepredicttop))  
0.982822572954
```

این عدد بسیار بالا و خوب است . دقت ۰,۹۸ بسیار ایده آل است . برای درک بهتر درخت تصمیم آن از ویژگی های گرافیکی sklearn استفاده کردیم تا آن را برا ما رسم کند. تحلیل عکس کمی سخت و دشوار است چرا که درخت بزرگی دارد. (زیرا یک مسئله regression است)

```
1 image(graph[0].create_png())
```



عکس اصلی در فولدر پروژه ضمیمه می شود . (dtree.png)

مرحله پنجم: استفاده از شبکه عصبی

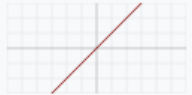
تمرکز اصلی پژوهش ما بر روی این بخش است. زیرا با توجه به ویژگی های این دیتا ست به نظر می رسد که با شبکه عصبی

نتیجه خوبی را به عمل بیاورد. کتابخانه استفاده شده در این بخش MLPRegressor است.


ما ۴ بخش اصلی شبکه عصبی را در نظر گرفتیم :

- تعداد لایه ها: از بین ۲۵ تا ۵۰ لایه می خواهیم بهترین تعداد لایه انتخاب شود.
- نرخ تغییر: می خواهیم در هر iteration به میزان ۱.۰ تغییر کنیم.
- تعداد iteration ها
- نوع activation function: اکتیویشن های مختلف را انتخاب کردیم. یکی از ۴ اکتیویشن زیر به عنوان تابع نهایی انتخاب میشود.

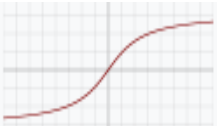
Identity ۱

Identité/Rampe		$f(x) = x$	$f'(x) = 1$
----------------	---	------------	-------------


logestic ۲

	$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
---	-------------------------------	--------------------------

tanh ۳

	$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
--	-----------------------	-----------------------------

relu ۴

	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
---	--	---

نتیجه بدست آمده از شبکه عصبی فوق به شکل زیر است:

```
1 print ("r2_score is",Max)
2 print (w)
```

```
r2_score is 0.899114437131
[41, 0.10000000000000001, 1, 120]
```

❖ تعداد لایه ها: ۴۱

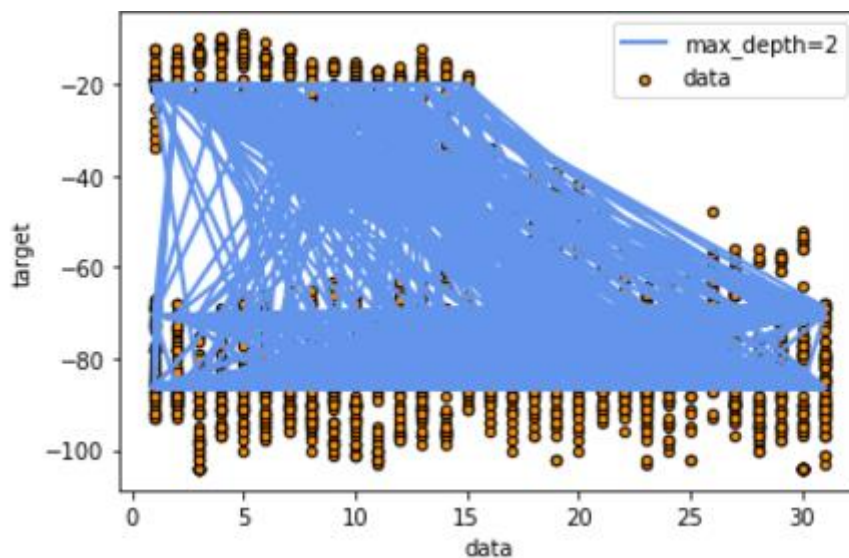
❖ نرخ تغییر: ۱۰۰۰.۰

❖ تابع فعال سازی: identity

❖ تعداد iteration ها: ۱۲۰

❖ میزان r^2 -score: برابر با ۰,۸۹

دقت بسیار خوب و در زمان بسیار کمی انجام شد. شکل لایه های عصبی که برای دیتا ست تشخیص داده شده است را میتوانید در شکل زیر مشاهده کنید.



نتیجه گیری:

از بین روش های انتخاب شده درخت تصمیم و شبکه عصبی بهترین دقت r^2 -score را دارا بودند:

شبکه عصبی: ۰,۸۹

درخت تصمیم: ۰,۹۸

اما درخت تصمیم زمان طولانی تری را برای انجام عملیات صرف میکند. از طرفی با کمی جا به جا کردن ۴ عامل مطرح شده در

شبکه عصبی می توان حتی نتیجه ای بهتر هم گرفت. بنا براین الگوریتم پیشنهادی برای این دیتا ست , **شبکه عصبی** است.