به نام خدا

حسام دامغانیان – (96216009)

تاریخ: 10/ خرداد /00

گزارش پروژه سوم

درس: داده‌کاوی

استاد درس: دکتر فراهانی

A. به طور کلی هنگامی که مسئله کلاس بندی دارای مرز های تصمیم خطی نیست نیاز است تا با استفاده از هسته ها بازنمایی دیگری از آن ها بدست آوریم تا بتوان مرزهای تصمیم خطی داشته باشیم. در استفاده از روش ماشین های پشتیبان هسته های رایجی وجود دارند که هر کدام با توجه به توزیع داده ها و مورد استفاده های خاص مورد استفاده قرار می‌گیرند:

a. هسته چند جمله ای:

$$k(x_i, x_j) = (x_i . x_j + 1)^d$$

که در روش های پردازش تصویر بیشتر بکار گرفته می‌شود. (d درجه ی چندجمله ای است.)

b. هسته گوسی:

$$k(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$$

این هسته هنگامی استفاده می‌شود که فرض پیشینی راجع به داده ها وجود ندارد.

c. هسته RBF:

$$k(x, y) = exp(-\gamma\|x - y\|^2)$$

این هسته نیز هسته ای همه منظوره است و هنگامی استفاده می‌شود که فرض پیشینی راجع به داده ها وجود ندارد.

B. نتایج استفاده از روش ماشین های پشتیبان بر داده های قیمت های موبایل:

هسته	درجه	گاما	دقت (درصد)
خطی	3	Scale	96.5
چندجمله ای	3	Scale	99.2
چندجمله ای	4	Scale	99.95
چندجمله ای	5	Scale	99.95
چندجمله ای	6	scale	99.9
سیگموید	3	auto	22.45
RBF	3	auto	97.2

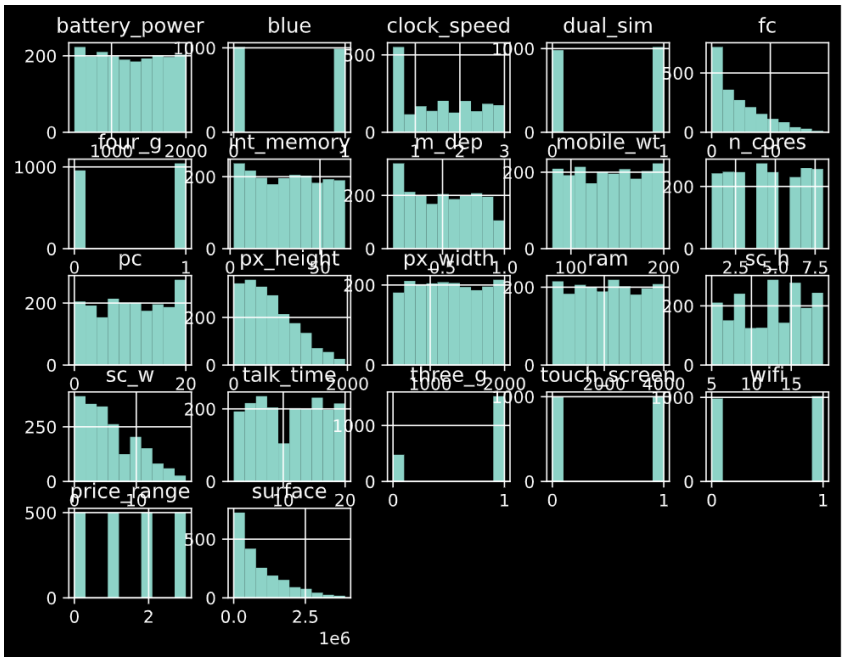
چندجمله ای	3	auto	99.2
RBF	3	Scale	95.5
RBF	3	Scale	97.2

C. می‌توان با تغییر دادن مقدار c حاشیه نرم و سخت را در روش ماشینهای پشتیبان بررسی کرد.

C	هسته	درجه	گاما	دقت (درصد)
0.1	چندجمله ای	4	Scale	97
2	چندجمله ای	4	Scale	100

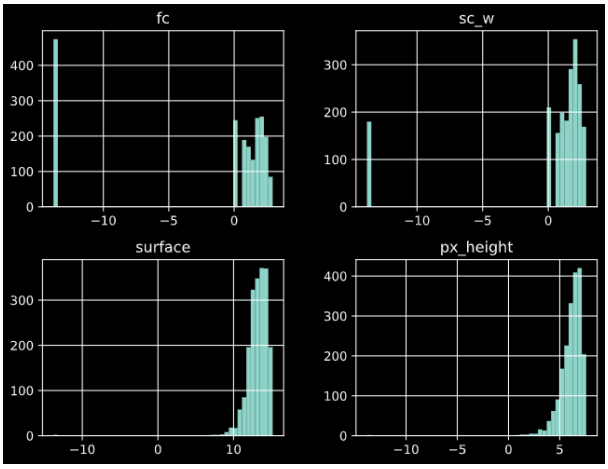
D. هنگامی استفاده از داده های دسته‌ای اگر مقادیر آن های به صورت دنباله ای از اعداد طبیعی باشد بعضی از روش ها ممکن است بین آن ها ترتیب قائل شوند در صورتی که همچین ترتیبی در داده ها وجود ندارد. برای همین یکی از روش هایی که می‌توان این داده ها را کدگذاری کرد و در روش های یادگیری ماشین از آن ها استفاده کرد به کارگیری کد گذاری one hot encoding است که در این روش به تعداد دسته های متفاوت موجود در داده ها ستون جدید اضافه می‌شود و هر دسته از داده های به ستون متناظر با خود نگاشت می شود. (در آن ستون مقدار یک و دیگر ستون ها مقدار صفر بر اساس دسته ای که داده به آن تعلق دارد قرار داده می‌شود.)

E. نرمال بودن شرط بسیاری از روش های آماری است و نتایج ریاضی مهمی مانند قضیه حد مرکزی را در بر دارد؛ درنتیجه نیاز داریم که توزیع داده های ما به این صورت باشد ولی در مواردی توزیع داده های موجود به این صورت نیست یا به اصطلاح انحراف دارد. تبدیل لگاریتمی در عمل هر داده را با لگاریتم آن جایگزین می‌کند و در نتیجه توزیع داده های حاصل انحراف قبلی را نخواهد داشت و به توزیع نرمال نزدیک تر خواهد بود. تبدیل نمایی نیز هر داده را با توان دلخواهی از آن جایگزین می‌کند و به این ترتیب توزیع داده ها را به توزیع نرمال نزدیک تر می‌کند. توزیع داده های موجود به صورت زیر است:

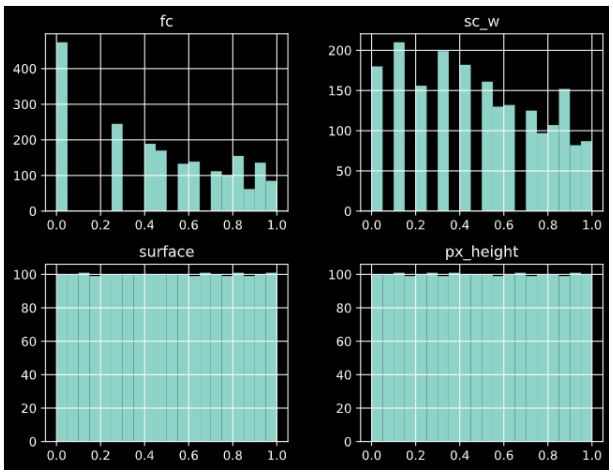


همانطور که مشاهده می‌کنید داده های fc ، sc_w ، surface ، px_height از توزیع نرمال طبعیت نمی‌کنند. برای برطرف کردن این موضوع از سه تبدیل لگاریتمی، box-cox، و yeo-johnson استفاده می‌کنیم: (تبدیلات به صورت ریاضی در نوتبوک توضیح داده شده اند.)

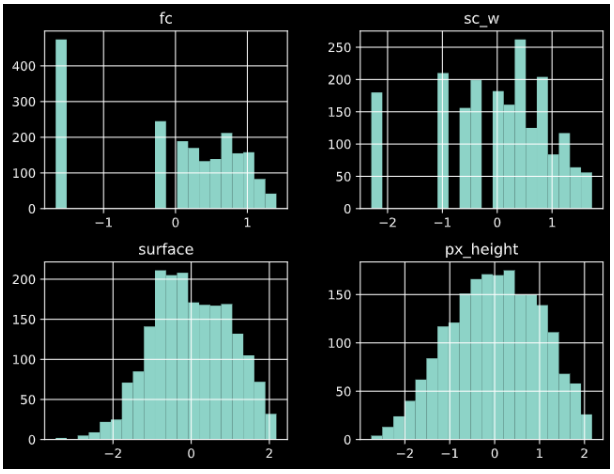
تبدیل لگاریتمی:



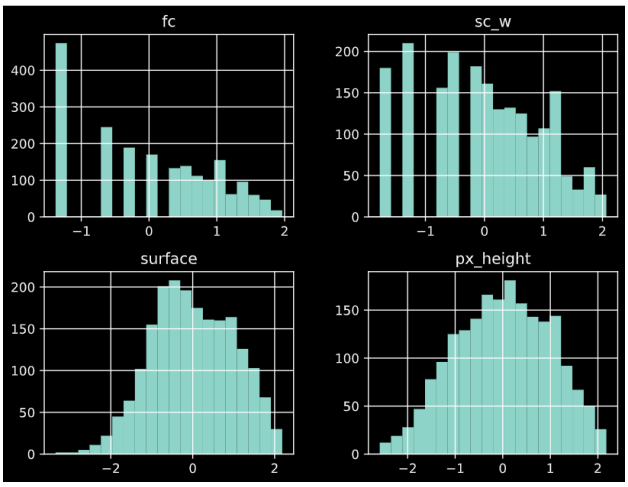
تبدیل چارکی: (تبدیل به توزیع یکپارچه)



تبدیل **Box-Cox**:



تبدیل **Yeo-Johnson**:



F. اعمال تغییرات مربوط به مهندسی ویژگی ها؛ هسته استفاده شده در تمامی مدل ها RBF با درجه سه است و تغییرات در دو حالت با داده های استاندارد شده و داده های خام بررسی شده و بهترین نتیجه گزارش شده است. (تمامی نتایج در نوتبوک های پیوست شده موجود است.)

با استفاده از	دقت(درصد)
حالت پایه	97.2
بسته بندی توان باطری	80.9
کدگذاری **one hot encoding**	97.35
تبدیل **Box-Cox**	93.7
تبدیل **Yeo-Johnson**	93.7
تبدیل **log transform**	89.9
تبدیل چارکها	97.1
ستون اضافه شده "مساحت"	29.6
تمام تغییرات	79

G. درخت های تصمیم ساختار درختی مانند جدول کنترل دارد که هر راس، غیر برگ، آزمونی برای هر ویژگی است و داده های موجود در هر راس فرزند را می‌توان براساس ویژگی و معیار راس والد آن ها دسته بندی کرد و نشان داد. قسمت اصلی درخت تصمیم، آزمونی است که در هر راس قرار دارد. انتخاب کردن آزمون، ویژگی مورد آزمون و معیار توقف درخت از مسائلی است که در الگوریتم های مختلف درخت تصمیم متفاوت است. برای مثال در الگوریتم ID3 ویژگی مورد آزمون در هر راس بر اساس آنتروپی یا information gain آن ویژگی انتخاب می‌شود بر این اساس که آن ویژگی‌ای انتخاب می‌شود که آنتروپی داده های در دسته بعدی بوجود آمده کمتر باشد یا معادلا information gain آن بیشتر باشد. الگوریتم CART که مخفف درخت های تصمیم و رگرسیون است به این صورت عمل می‌کند که به صورت بازگشتی با تمام ویژگی ها داده آزمون را بررسی می‌کند و به صورت حریصانه نقطه مرز را پیدا می‌کند و در نهایت ویژگی ای را که بر حسب تابع هزینه بهتر عمل کرده باشد را برای آزمودن در راس فعلی انتخاب می‌کند. برای مسائل دسته بندی شاخص جینی و برای مسائل رگرسیون mse از توابع هزینه پرکاربرد هستند. معیار توقف در این مدل نیز حداقل مقدار داده در راس برگ است که معیار رایجی است.

H. نتایج درخت تصمیم:

میانگین 10-Fold CV: 83.85%

I. نتایج درخت تصمیم با پارامتر های مختلف:

عمق درخت	1	5	9	13	17
دقت (درصد)	50	88.2	98.80	100	100

حداقل نمونه در برگ	20	15	10	5
دقت (درصد)	88.25	89.2	90.85	94.4

با توجه به نتایج بدست آمده افزایش حداقل نمونه و عمق باعث افزایش دقت مدل می‌شود. (البته واریانس مدل افزایش می‌یابد.)

J. حرص کردن درخت باعث افزایش تفسیرپذیری و کاهش احتمال بیش برازش می‌شود و بخشی ضروری در آموزش مدل های درخت تصمیم است. یکی از روش های ساده حرص کردن به این شکل است که راسی را حذف کرده و تاثیر آن راس را در دقت مدل با مجموعه داده های آزمون مورد بررسی قرار می‌دهیم و راس با کمترین تاثیر در دقت حذف می‌شود. روش دیگر روش حرص کردن پیچیدگی هزینه است، یا ضغیف ترین حلقه، به این صورت که ضریبی مانند آلفا استفاده می‌شود تا ارزیابی کند که راس بر اساس اندازه ی زیر درخت هایش قابل حذف کردن هست یا خیر. نتایج استفاده از مقادیر مختلف آلفا در درخت تصمیم در جدول زیر آمده است:

مقادیر بالاتر	**0.11**	**0.06**	**0**	مقدار **C**
25	**50**	**75.4**	**83.85**	دقت (درصد)

K. نتایج **Random Forrest** :

میانگین **10-Fold CV**: **88.2%**

نتایج درخت تصمیم از random forest ضعیف تر است به این دلیل که زیر مجموعه هایی تصادفی از داده ها را انتخاب کرده و با میانگین گرفتن هم دقت مدل را افزایش داده و هم از بیش برازش جلوگیری می‌کند.

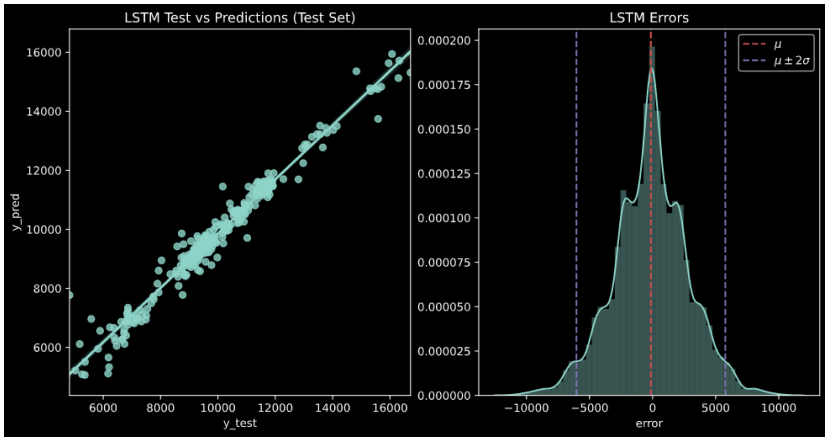
L. محبوبیت درخت های تصمیم: یکی از دلایل اصلی، پارامتر های کم در این مدل ها است که آموزش آن ها را با داشتن حجم داده های بزرگ همچنان در زمان معقول نگه می‌دارد و از لحاظ عملکرد نیز قابل رقابت با مدل های جدید است. تفسیر پذیری این مدل ها در برابر مدل های یادگیری عمیق که به اصطلاح جعبه سیاه هستند، نیز موجب استمرار استفاده از این مدل ها شده است.

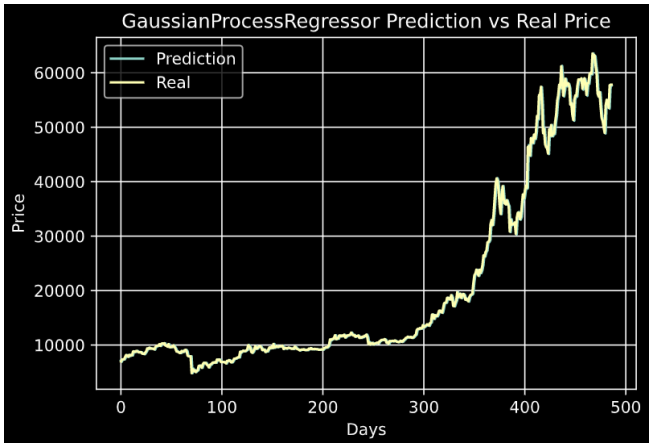
M. بله با تبدیل داده های سری زمانی به قالب مورد نظر برای یادگیری نظارتی توسط روش هایی نظیر پنجره‌ی متحرک می‌توان از درخت های تصمیم در مسائل مربوط به دسته بندی و رگرسیون سری های زمانی استفاده کرد.

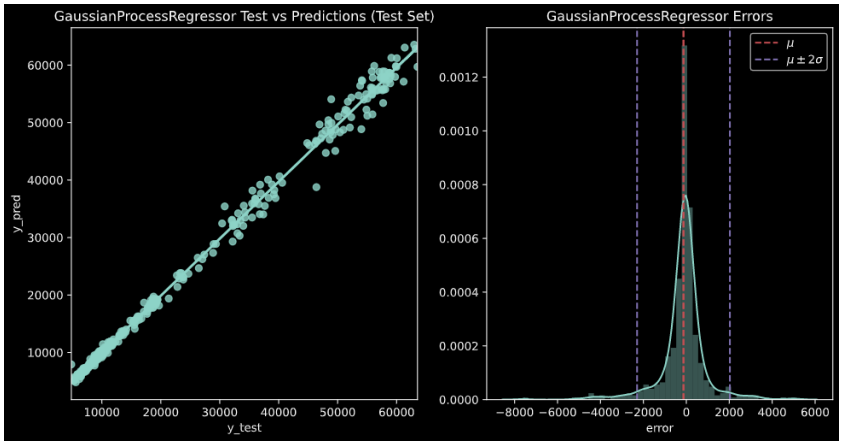
N. گزارش سوال های **17** تا **22**

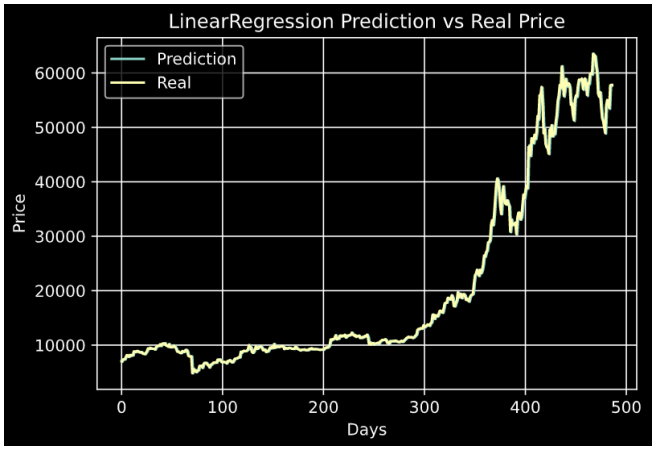
روش رگرسیون	RMSE	Performance	Precision	R2 score	MAE
Gradient Boost	15685.62	57	64.07	0.176	7873.44
Histogram Grad. Boost	16963.88	53	62.22	0.037	0.037
Random forest (DTree)	15660.11	59	62.03	0.179	7841.93
Linear Regression	1090.08	85	70.64	0.996	586.69
Bayesian Ridge	1090.13	85	70.64	0.996	586.67
RANSAC	1277.64	80	68.17	0.995	714.42
KNeighbour	17027.02	16	56.88	0.029	9599.24
MLP	1209.74	82	66.32	0.995	606.90
Decision Tree	15577.35	40	56.88	0.188	8001.00
Gaussian Process	1088.99	85	70.64	0.996	585.94
LSTM	437.74	84	66.33	-	9740.30

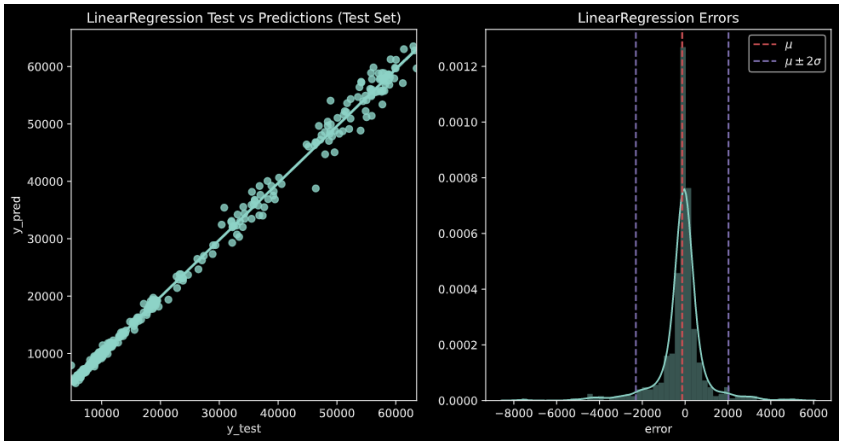
نمودار های خطا و پیشبینی مربوط به مدل های برتر:

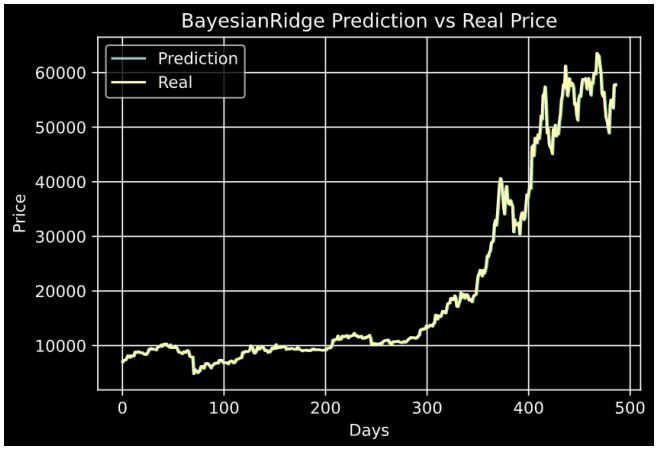


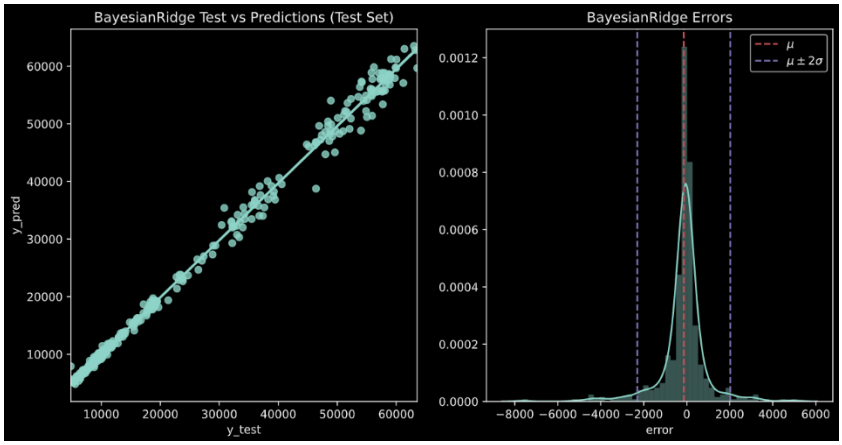

GaussianProcessRegressor Prediction vs Real Price
Prediction
Real
Price
Days

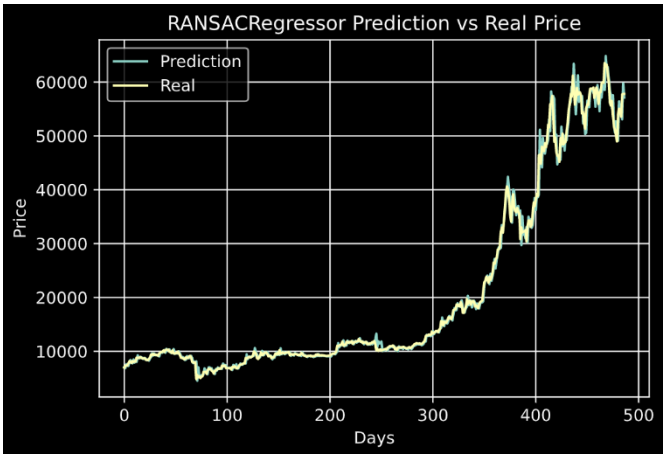

GaussianProcessRegressor Test vs Predictions (Test Set)
y_pred
y_test
GaussianProcessRegressor Errors
μ
μ ± 2σ
error

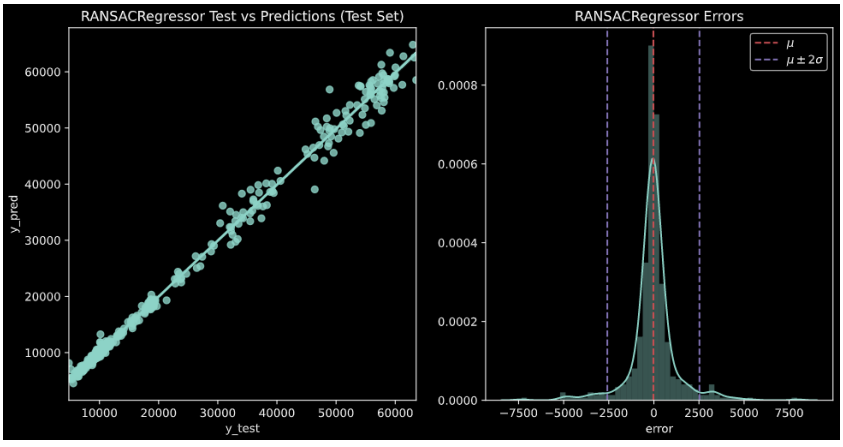

LinearRegression Prediction vs Real Price
Prediction
Real
Price
Days


LinearRegression Test vs Predictions (Test Set)
y_pred
y_test
LinearRegression Errors
μ
μ ± 2σ
error


BayesianRidge Prediction vs Real Price
Prediction
Real
Price
Days

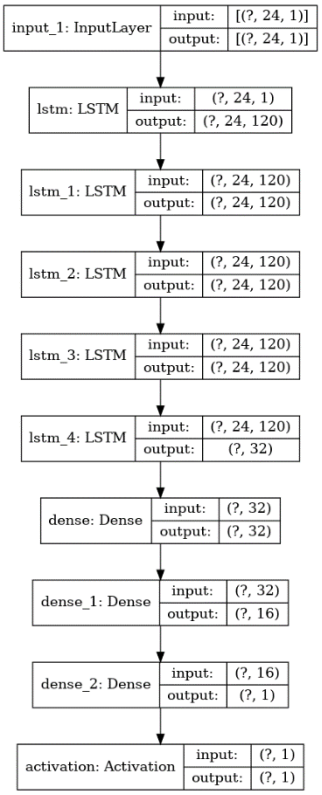

BayesianRidge Test vs Predictions (Test Set)
y_pred
y_test
BayesianRidge Errors
μ
μ ± 2σ
error


RANSACRegressor Prediction vs Real Price
Prediction
Real
Price
Days


RANSACRegressor Test vs Predictions (Test Set)
y_pred
y_test
RANSACRegressor Errors
μ
μ ± 2σ
error

روش های **Ensemble**: (استفاده از **5** مدل کلاسیک برتر)

	RMSE	Performance	Precision	R2 Score	MAE
Voting	3216.92	64	66.32	0.965	1699.37
Linear Reg. Bagging	1092.45	85	70.43	0.996	589.04
Bayesian Ridge Bagging	1092.52	85	70.64	0.996	589.00
MLP Bagging	1262.60	81	70.43	0.995	704.24
RANSAC Bagging	1108.70	85	71.25	0.996	509.92
DecisionTree Bagging	15754.73	57	63.04	0.169	7933.83



❋ساختار مدل LSTM: (شکل روبرو)

❋ معیار Performance قرار داشتن پیشبینی در بازه ٪5± قیمت واقعی است.

❋ معیار Precision پیشبینی صعودی/نزولی بودن روند قیمت است.

گزارش دقت مدل ها با استفاده از AdaBoost:

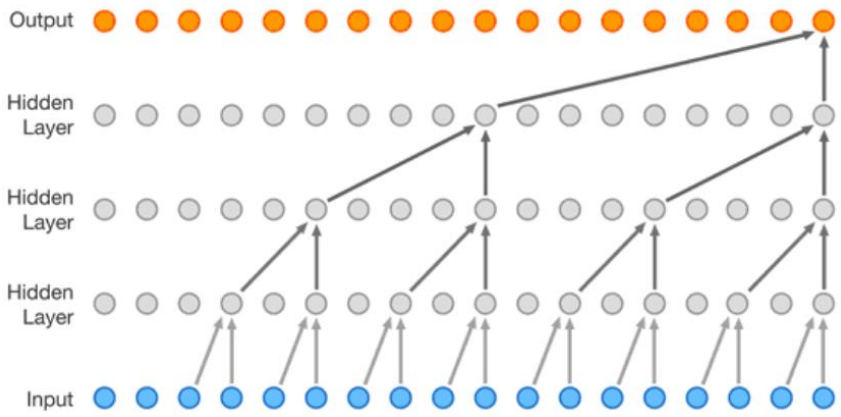
AdaBoost	RMSE	Performance	Precision	R2 Score	MAE
DTree	16272.33	53	60.99	0.113	8270.00
Linear Reg.	1209.98	82	67.97	0.995	670.75
KNeighbour	16611.38	16	57.91	0.076	9285.46
BayesianRidge	1162.19	83	70.64	0.995	644.06
RANSAC	1263.35	80	68.99	0.995	691.80

گزارش دقت مدل ها با استفاده از **Random Forest**:

Random Forest	RMSE	Performance	Precision	R2 Score	MAE
With MAE Criterion	15719.47	60	62.63	0.173	7880.60
With Pruning	15704.51	60	63.45	0.174	7880.92
Base	15761.91	59	63.45	0.168	7905.70

O. گزارش سوال **24**:

یکی از مدل هایی جدید که برای پیشبینی سری های زمانی از آن استفاده می‌شود شبکه های کانولوشنی زمانی، tcn، است. این مدل در سال 2016 در مقاله‌ی wavenet که برای پردازش صوت در تولید صدای های مورد نیاز برای بات های سخنگو استفاده می‌شد معرفی شد (استفاده شد). این مدل با توجه به ساختارش ایده آل برای پیشبینی سری های زمانی است به دلیل اتصالات جانبی، بر خلاف مدل های حافظه دار مانند GRU و LSTM مشکلی در یادگیری الگوهای قدیمی ندارد. ساختار کلی این مدل را در شکل زیر می‌توان مشاهده کرد:



نتایج این مدل:

	RMSE	Performance	R2 Score	MAE
TCN	1815.44	55	0.989	1290.78

P. گزارش سوال **25**:

از فیچر های RSI، MA، ATR، BollingerBands و بیشترین و کمترین قیمت استفاده شد. نتایج به صورت زیر است:

روش رگرسیون	RMSE	Performance	Precision	R2 score	MAE
Linear Regression	1092.52	85	70.84	0.996	596.52
Bayesian Ridge	1092	85	71.05	0.996	595.20
RANSAC	1272.73	81	68.38	0.995	699.10
MLP	12563.14	24	51.75	0.472	4975.14

در سه مدل اول دقت دسته بندی افزایش داشت ولی در مدل شبکه عصبی به کلی مشکل ایجاد شد. بهبود نیز به این دلیل است که با مهندسی فیچر های موجود (روش های اقتصادی) به فیچر های جدیدی رسیدیم که توضیح بهتری از داده ها داشتند و در نتیجه دقت افزایش داشت.

Q. گزارش سوال **26** و **27**:

	RMSE	Precision	MAE
Hist. Gradient Boost	0.22	68.76	0.152
Linear Reg.	0.22	68.78	0.153
Bayesian Ridge	0.22	68.76	0.152
MLP Reg.	0.22	73.35	0.155
DTree	0.33	70.62	0.245

روش Bagging:

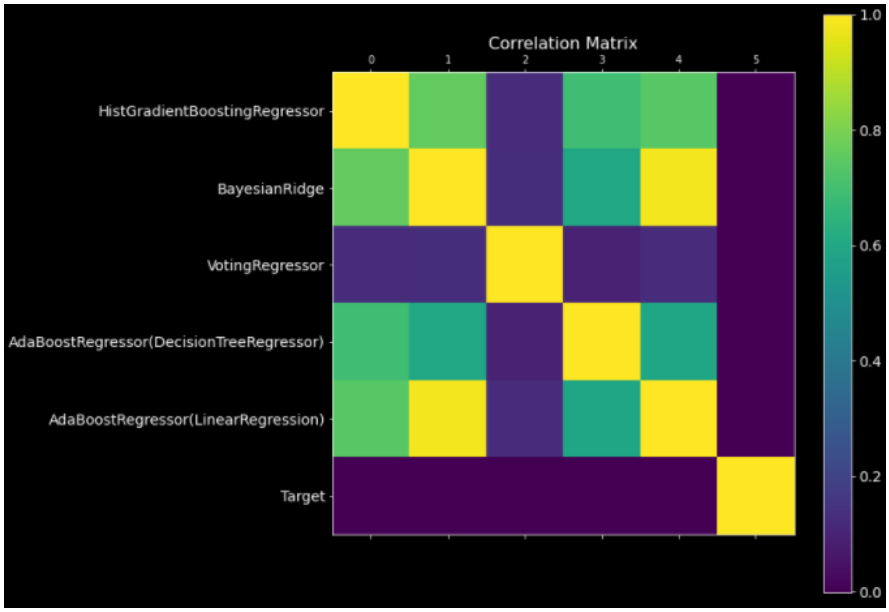
	RMSE	Precision	MAE
Hist. Gradient Boost	0.22	67.89	0.151
Bayesian Ridge	0.22	68.57	0.152

روش Voting:

	RMSE	Precision	MAE
Voting Reg.	0.23	67.67	0.167

روش AdaBoost:

AdaBoost	RMSE	Precision	MAE
DTree	0.22	66	0.152
Linear Reg.	0.22	68.70	0.153



در روش های Ensemble بهتر است ورودی های داده شده نسبت به هم کمترین همبستگی را داشته باشند. برای مثال در voting رگرسیون/دسته‌بندی هر چه ورودی ها با یکدیگر همبستگی بیشتری داشته باشند، اضافه کردن شان بی دلیل است و با یک نماینده از آن ها نیز همان خروجی به دست خواهد آمد، در نتیجه بهتر است که کمترین همپوشانی را داشته باشند که در صورت اشتباه کردن یک مدل، خطای آن توسط پیشبینی درست دیگر مدل ها پوشانده شود.

R. روش های یادگیری قانون: (IREP, RIPPER)

روش های استخراج قانون دسته ای از روش های یادگیری ماشین هستند که با بدست آوردن شرط های به حالت اگر–آنگاه که در مجموع اطلاعات موجود در سیستم یادگیری ماشین را نمایش می‌دهند. برای مثال:

```
R1: IF age = youth AND student = yes
   THEN buy_computer = yes
```

یک سری از این روش ها، روش های مبنی بر IREP هستند که مخفف Incremental Reduced Error Pruning هستند.

Ripper (هرس افزایشی مکرر برای کاهش خطا) یکی از کارآمدترین و مورد استفاده ترین الگوریتم های یادگیری قانون است. یک استراتژی تقسیم و غلبه را برای استخراج قوانین اجرا می کند. Ripper اعمال به اصطلاح افزایشی کاهش هرس خطا (IREP) را برای بدست آوردن مجموعه ای اولیه از قوانین برای هر کلاس انجام می‌دهد. سپس یک گام بهینه سازی اضافی هر قاعده در مجموعه فعلی را به نوبه خود در نظر می گیرد و دو قانون جایگزین از آن ها ایجاد می کند: یک قاعده جایگزینی و یک قاعده تجدید نظر. پس از آن، تصمیمی در مورد اینکه آیا مدل باید قاعده اصلی، جایگزینی یا قانون تجدید نظر را بر اساس معیار حداقل طول توصیف نگه دارد یا نه، گرفته می شود.

روش دیگر روش از این روش های استخراج قانون، (Projective Adaptive Resonance Theory) PART است. این روش یک الگوریتم درخت تصمیم جزئی است. به طور خاص، PART مجموعه ای از قوانین را با توجه به استراتژی تقسیم و غلبه تولید می کند، تمام نمونه ها را از مجموعه آموزشی که تحت پوشش این قاعده قرار می گیرند حذف می کند و تا زمانی که هیچ نمونه ای باقی نماند، به صورت بازگشتی پیش می رود. برای تولید یک قانون، PART یک درخت تصمیم گیری جزئی C۴.۵ برای مجموعه نمونه های فعلی می سازد و برگی را با بزرگترین پوشش به عنوان قانون جدید انتخاب می کند. پس از آن، درخت تصمیم جزئی همراه با نمونه های تحت پوشش قانون جدید از داده های آموزشی حذف می شود، تا از عمومی شدن زودهنگام، Early Generalization، جلوگیری شود. این فرایند تا زمانی تکرار می شود که تمام نمونه ها تحت پوشش قوانین استخراج شده قرار گیرند.