درس علم داده

پروژه پایانی

نام دانشجو: شقایق شهبازی

نام استاد: دکتر رضاپور

شماره دانشجویی: ۹۹۳۶۱۳۰۴۰

فهرست مطالب

مجموعه داده

در این بخش از پروژه مجموعه داده "Energy Efficiency Data Set" درنظر گرفته شده است. این مجموعه داده شامل اطلاعاتی درباره ویژگی‌های ساختمان‌ها و بازده انرژی آن‌ها است که برای تحلیل کارایی انرژی استفاده می‌شود. این مجموعه داده شامل دو ستون هدف با نام‌های "Heating Load" و "Cooling Load" است که به ترتیب بیان‌گر بار سرمایشی و گرمایشی ساختمان است.

این مجموعه داده شامل ۸ ویژگی است از جمله نوع ساختمان، سال ساخت، نسبت نواحی شیشه‌ای، مساحت سطح، عمق دیوار، ضخامت سقف، مساحت شیشه‌ای و جهت ساختمان. این ویژگی‌ها از نظر تکنیکی و آماری متنوع هستند و می‌توانند برای بررسی ارتباطات مختلف بین متغیرها و بازده انرژی مورد استفاده قرار گیرند.

این مجموعه داده به دلیل اهمیت اقتصادی و محیطی که مصرف انرژی ساختمان‌ها دارد، انتخاب شده است. با تحلیل دقیق این داده‌ها می‌توان الگوهایی را در مصرف انرژی ساختمان‌ها شناسایی نموده و بهینه‌سازی‌های لازم را برای افزایش کارایی انرژی ارائه دهیم.

۱- تحلیل اکتشافی داده‌ها

در این بخش هدف ما بررسی و تحلیل مجموعه داده‌های ویژگی‌های ساختمان‌های مسکونی است تا الگوها، وابستگی‌ها و ویژگی‌های مهم داده‌ها را شناسایی کنیم. این مرحله به عنوان پایه‌ای برای پیش‌پردازش داده‌ها و مدل‌سازی یادگیری ماشین عمل می‌کند و باعث می‌شود تا تصمیمات مبتنی بر داده‌ها با دقت و اطمینان بیشتری اتخاذ شوند. در این بخش، گام‌های مختلفی برای بررسی و تحلیل داده‌ها انجام می‌شود تا داده‌ها به خوبی درک و آماده استفاده در مراحل بعدی شوند.

در ابتدا لازم است داده‌ها را بارگذاری کرده تا با ساختار کلی داده‌ها آشنا شویم. این کار به ما کمک می‌کند تا نگاهی اولیه به داده‌ها داشته باشیم و نوع داده‌ها، تعداد ستون‌ها و ردیف‌ها را بشناسیم. همانطور که در تصویر ۱-۱ قابل مشاهده است، داده‌های موردنظر را که در پیوست ۱ موجود هستند، بارگذاری می‌کنیم و ۵ سطر ابتدایی آن را مشاهده می‌کنیم. در تصویر ۱-۲ نیز اطلاعات آماری کلی داده‌ها بررسی شده و قابل مشاهده هستند.

```
data = pd.read_csv('energy_efficiency_data.csv')
data.head()
```

✓ 0.4s

	Relative_Compactness	Surface_Area	Wall_Area	Roof_Area	Overall_Height	Orientation	Glazing_Area	Glazing_Area_Distribution	Heating_Load	Cooling_Load
0	0.98	514.5	294.0	110.25	7.0	2	0.0	0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3	0.0	0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4	0.0	0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5	0.0	0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2	0.0	0	20.84	28.28

شکل ۱-۱ پنج سطر ابتدایی داده ها

```
data.describe()
```
✓ 0.3s

	Relative_Compactness	Surface_Area	Wall_Area	Roof_Area	Overall_Height	Orientation	Glazing_Area	Glazing_Area_Distribution	Heating_Load	Cooling_Load
count	768.000000	768.000000	768.000000	768.000000	768.00000	768.000000	768.000000	768.00000	768.000000	768.000000
mean	0.764167	671.708333	318.500000	176.604167	5.25000	3.500000	0.234375	2.81250	22.307201	24.587760
std	0.105777	88.086116	43.626481	45.165950	1.75114	1.118763	0.133221	1.55096	10.090196	9.513306
min	0.620000	514.500000	245.000000	110.250000	3.50000	2.000000	0.000000	0.00000	6.010000	10.900000
25%	0.682500	606.375000	294.000000	140.875000	3.50000	2.750000	0.100000	1.75000	12.992500	15.620000
50%	0.750000	673.750000	318.500000	183.750000	5.25000	3.500000	0.250000	3.00000	18.950000	22.080000
75%	0.830000	741.125000	343.000000	220.500000	7.00000	4.250000	0.400000	4.00000	31.667500	33.132500
max	0.980000	808.500000	416.500000	220.500000	7.00000	5.000000	0.400000	5.00000	43.100000	48.030000

شکل ۲-۱ اطلاعات آماری داده ها

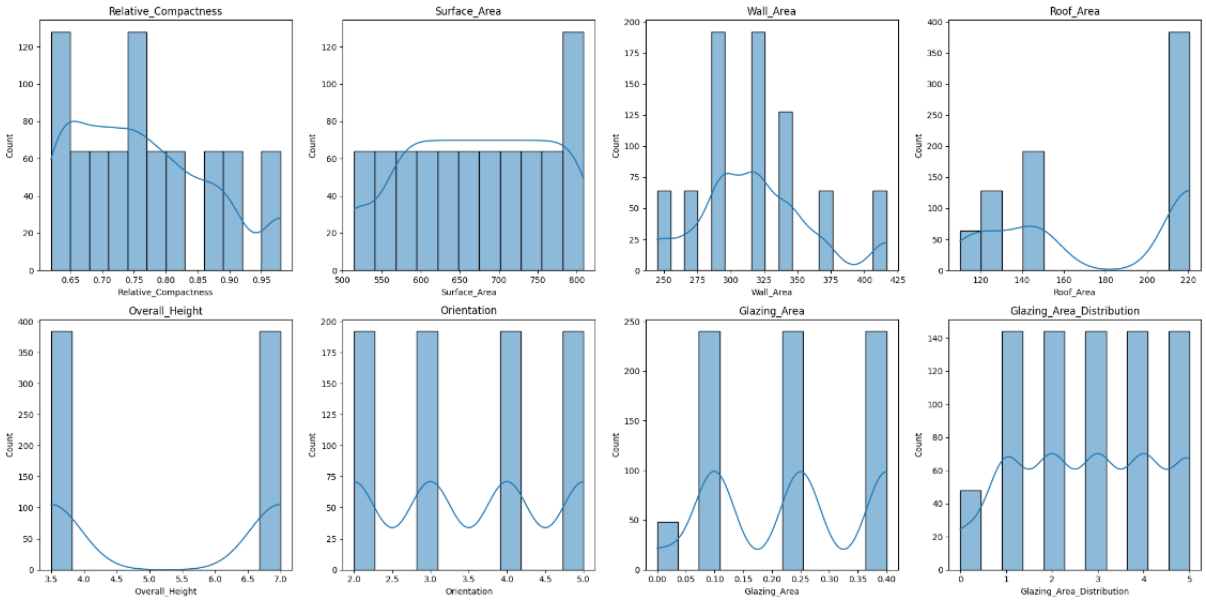
پس از بررسی کلی داده‌ها لازم است تا وجود داده‌های گمشده مورد بررسی قرار گیرند. شناسایی و مدیریت داده‌های گمشده یکی از مراحل مهم در پیش‌پردازش داده‌ها به شمار می‌روند. ابتدا مطابق تصویر ۱-۳ تعداد مقادیر گمشده در هرستون را لازم است مشخص کنیم. همانطو که از نتیجه اجرای این کد مشخص است هیچ‌گونه داده گمشده‌ای در هیچ ستونی وجود ندارد؛ لذا نیازی به انجام اقدامی در این زمینه وجود ندارد.

```
missing_values = data.isnull().sum()
print(missing_values)
```
✓ 0.0s

```
Relative_Compactness         0
Surface_Area                 0
Wall_Area                    0
Roof_Area                    0
Overall_Height               0
Orientation                  0
Glazing_Area                 0
Glazing_Area_Distribution    0
Heating_Load                 0
Cooling_Load                 0
dtype: int64
```

شکل ۳-۱ وضعیت داده های گمشده

اکنون می‌توانیم به تحلیل و بررسی چگونگی توزیع داده‌ها بپردازیم. به این منظور نمودارهای توزیع مانند هیستوگرام و KDE ترسیم می‌شوند. مطابق با تصویر ۱-۴ این نمودارها به کمک می‌کنند تا الگوهای موجود در داده‌ها را شناسایی کنیم و درک کنیم که توزیع آن‌ها به چه صورت است.



شکل ۴-۱ هیستوگرام داده ها

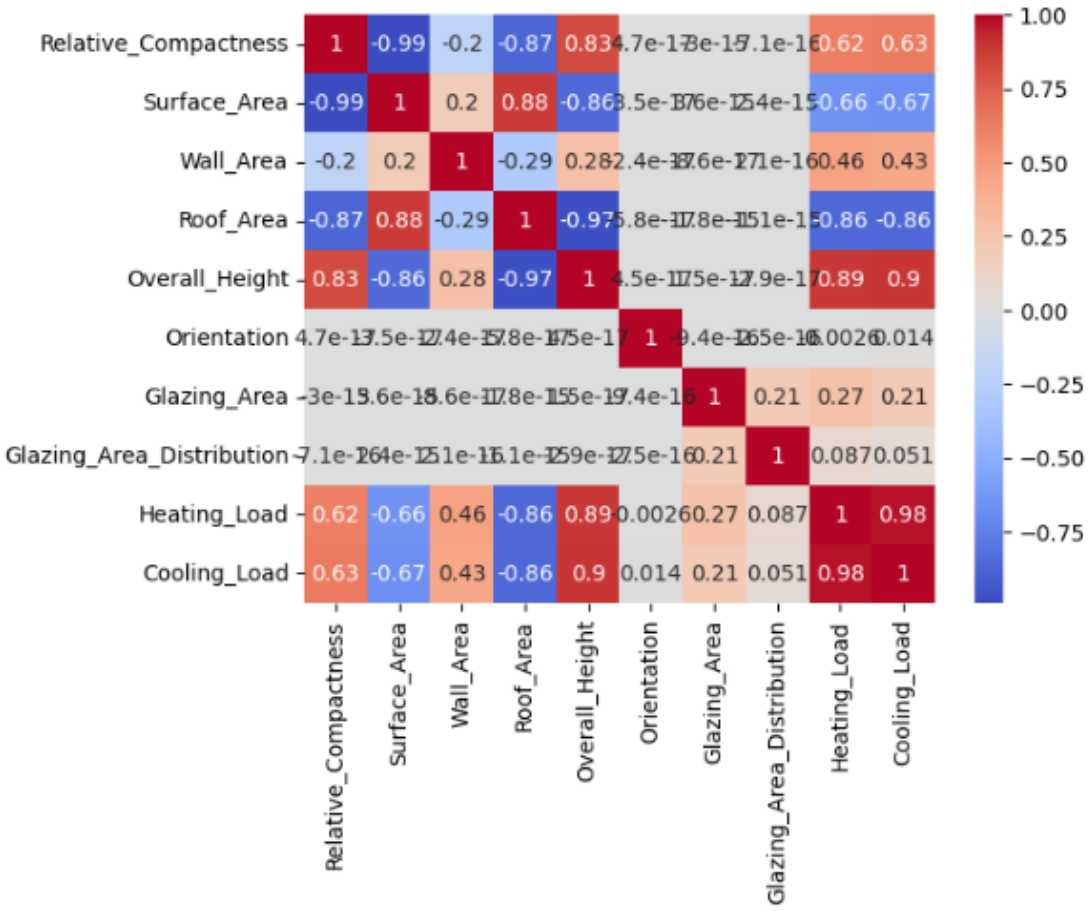
در ادامه لازم است به بررسی روابط موجود بین متغیرها بپردازیم. بررسی روابط ویژگی‌ها یکی از مراحل کلیدی در تحلیل اکتشافی داده است. با استفاده از نمودارهای پراکندگی و ماتریس همبستگی، می‌توانیم روابط خطی و غیرخطی بین ویژگی‌ها را شناسایی کنیم. در تصویر ۱-۵ ماتریس همبستگی ویژگی‌های این مجموعه داده قابل مشاهده است. این ماتریس اطلاعات مختلفی درمورد وضعیت وابستگی متغیرها نسبت به یکدیگر را به ما نشان می‌دهد. به طورکلی همبستگی‌های موجود بین ویژگی‌ها به شرح زیر هستند:

- **همسبتگی بین ویژگی‌ها:**
  - ✓ **Relative_Compactness و Surface_Area**: مشاهده می‌شود که این دو ویژگی با ضریب همبستگی -۰.۹۹، همبستگی منفی بسیار قوی دارند. این به این معناست که با افزایش Relative_Compactness، میزان Surface_Area کاهش می‌یابد و بالعکس.
  - ✓ **Roof_Area و Wall_Area**: این دو ویژگی با ضریب همبستگی -۰.۲۹، همبستگی منفی کمی دارند.
  - ✓ **Overall_Height و Roof_Area**: این دو ویژگی نیز همبستگی مثبت قوی با ضریب همبستگی ۰.۹۷ دارند، که نشان می‌دهد با افزایش Overall_Height، Roof_Area نیز افزایش می‌یابد.

- **همسبتگی بین ویژگی‌ها و متغیرهای هدف:**
  - ✓ **Heating_Load**: همبستگی قوی مثبتی با Relative_Compactness (۰.۶۲) و همبستگی منفی با Surface_Area (-۰.۶۶) دارد. این نشان می‌دهد که با افزایش Relative_Compactness، مقدار Heating_Load افزایش می‌یابد و با افزایش Surface_Area، مقدار Heating_Load کاهش می‌یابد.
  - ✓ **Cooling_Load**: همبستگی مثبتی با Relative_Compactness (۰.۶۳) و همبستگی منفی با Wall_Area (-۰.۴۳) دارد. این به معنای این است که با افزایش Relative_Compactness، مقدار Cooling_Load افزایش می‌یابد و با افزایش Wall_Area، مقدار Cooling_Load کاهش می‌یابد.

- ✓ **Heating_Load و Cooling_Load**: همبستگی بسیار قوی ۰.۹۸ بین این دو متغیر نشان می‌دهد که این دو متغیر به شدت به هم وابسته هستند و با تغییر یکی، دیگری نیز به صورت مشابه تغییر می‌کند.

بنابراین به طور کلی می‌توان اظهار داشت که ویژگی Relative_Compactness و Overall_Height تأثیر قابل توجهی بر هر دو متغیر Heating_Load و Cooling_Load دارد. این اطلاعات می‌تواند برای انتخاب ویژگی‌های مهم در مدل‌سازی و پیش‌بینی مفید باشد. همچنین، همبستگی قوی بین Heating_Load و Cooling_Load نشان می‌دهد که این دو متغیر می‌توانند به صورت مشترک در مدل‌ها استفاده شوند یا اثرات یکسانی بر برخی ویژگی‌ها داشته باشند.



شکل ۵-۱ ماتریس همبستگی

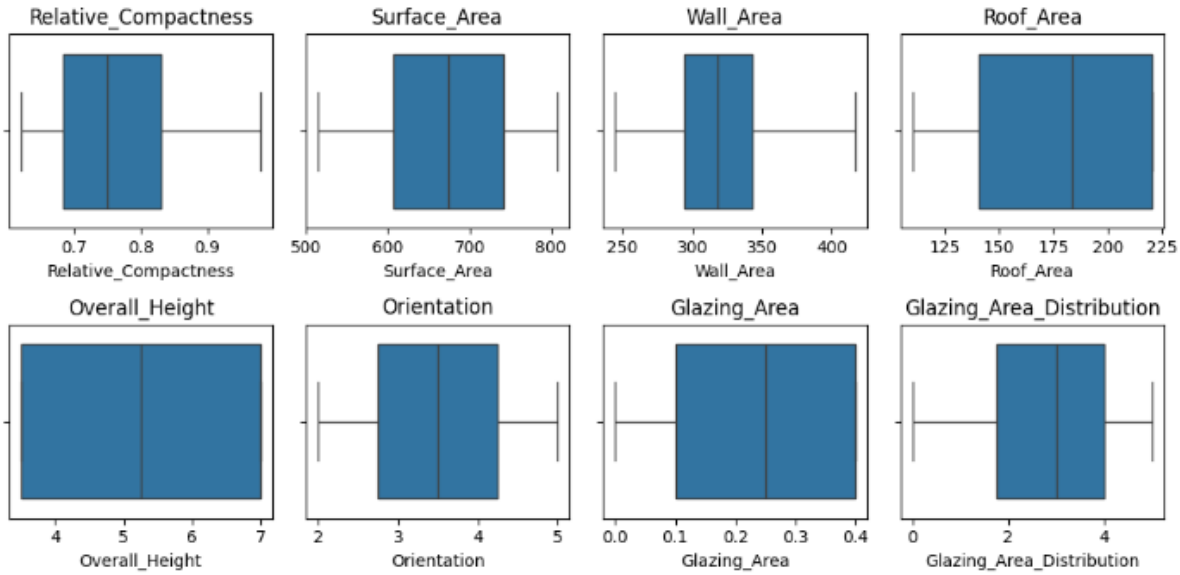
علاوه بر موارد مطرح شده، لازم است تا ناهنجاری‌ها و نقاط خارج از محدوده شناسایی شوند. این‌کار به ما کمک خواهد کرد تا داده‌های غیرمعمولی را که می‌توانند مدل‌های یادگیری ماشین را تحت تاثیر قرار دهند، شناسایی کنیم. یکی از روش‌های رایج برای این مورد رسم نمودار جعبه‌ای است که به ما کمک می‌کند تا ناهنجاری‌ها را به صورت بصری مشاهده کنیم و تصمیم بگیریم با آن‌ها چگونه برخورد کنیم. در تصویر ۱-۶ نمودار جعبه‌ای مجموعه داده موردنظر قابل مشاهده است.

```
fig, axes = plt.subplots(2, 4, figsize=(10, 5))
axes = axes.flatten()

for i, column in enumerate(data.columns[:8]):
    sns.boxplot(x=data[column], ax=axes[i])
    axes[i].set_title(column)

plt.tight_layout()
plt.show()
```

✓ 1.5s



شکل ۶-۱ نمودار جعبه ای برای بررسی ناهنجاری

۲- پیش‌پردازش داده‌ها

پیش‌پردازش داده‌ها مرحله‌ای حیاتی در فرآیند علم داده است که داده‌های خام را به داده‌های تمیز و قابل استفاده برای مدل‌سازی یادگیری ماشین تبدیل می‌کند. این مرحله شامل تمیزسازی داده‌ها، مدیریت داده‌های گمشده، تبدیل و مقیاس‌بندی ویژگی‌ها و تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی است. پیش‌پردازش صحیح داده‌ها می‌تواند به بهبود دقت و کارایی مدل‌های یادگیری ماشین کمک کند.

در ابتدا داده‌های غیرضروری را بهتر از مجموعه داده‌ها حذف کنیم تا مدل عملکرد بهتری داشته باشد به این منظور مجدد ماتریس همبستگی موجود در تصویر ۱-۵ را مورد بررسی قرار می‌دهیم. درحقیقت لازم است تا ویژگی‌هایی را که بیشترین تاثیر را بر متغیرهای هدف دارند شناسایی کنیم و ویژگی‌هایی را که همبستگی قوی با یکدیگر دارند را حذف کنیم. نتایج حاصل از بررسی مجدد ماتریس همبستگی به شرح زیر است:

- **ویژگی‌هایی با همبستگی قوی با متغیرهای هدف:**

✓ **Roof_Compactness**: همبستگی قوی مثبتی با Heating_Load (۰.۶۲) و Cooling_Load (۰.۶۳) دارد. این ویژگی باید حفظ شود زیرا تأثیر قابل توجهی بر هر دو متغیر هدف دارد.

✓ **Surface_Area**: همبستگی قوی منفی با Heating_Load (۰.۶۶-) دارد. این ویژگی نیز باید حفظ شود.

- ✓ **Wall_Area**: همبستگی منفی با Cooling_Load (۰.۴۳-) دارد. این ویژگی ممکن است مفید باشد اما همبستگی آن نسبت به ویژگی‌های دیگر کمتر است.
- ✓ **Overall_Height**: همبستگی مثبتی با Heating_Load (۰.۸۹) دارد و به طور غیرمستقیم با Cooling_Load هم تأثیرگذار است. این ویژگی نیز باید حفظ شود.
- ✓ **Heating_Load و Cooling_Load**: این دو متغیر هدف همبستگی بسیار قوی با هم دارند (۰.۹۸)، بنابراین می‌توان یکی از آن‌ها را برای مدل‌سازی انتخاب کرد اگر نیاز به ساده‌سازی باشد.

- **ویژگی‌هایی با همبستگی قوی با دیگر ویژگی‌ها:**

- ✓ **Surface_Area و Relative_Compactness**: همبستگی بسیار قوی منفی (۰.۹۹-) بین این دو ویژگی نشان می‌دهد که یکی از آن‌ها می‌تواند حذف شود. با توجه به تأثیر قوی Relative_Compactness بر متغیرهای هدف، پیشنهاد می‌شود که Relative_Compactness حفظ و Surface_Area حذف شود.
- ✓ **Overall_Height و Roof_Area**: همبستگی بسیار قوی مثبت (۰.۹۷) دارند، بنابراین یکی از آن‌ها می‌تواند حذف شود. با توجه به تأثیر Overall_Height بر متغیرهای هدف، بهتر است Overall_Height حفظ و Roof_Area حذف شود.
- ✓ **Glazing_Area و Glazing_Area_Distribution**: هر دو ویژگی همبستگی کم با متغیرهای هدف دارند و همبستگی قوی با دیگر ویژگی‌ها ندارند، بنابراین می‌توانند حفظ شوند مگر اینکه ساده‌سازی مدل اولویت داشته باشد.

بنابراین همانطور که در تصویر ۲-۱ می‌توان مشاهده کرد برخی از ویژگی‌ها مطابق با تحلیل بالا برای بهبود عملکرد مدل حذف خواهند شد.

```
data_cleaned = data.drop(columns=['Surface_Area', 'Roof_Area', 'Heating_Load'])
data_cleaned
```

✓ 0.0s

	Relative_Compactness	Wall_Area	Overall_Height	Orientation	Glazing_Area	Glazing_Area_Distribution	Cooling_Load
0	0.98	294.0	7.0	2	0.0	0	21.33
1	0.98	294.0	7.0	3	0.0	0	21.33
2	0.98	294.0	7.0	4	0.0	0	21.33
3	0.98	294.0	7.0	5	0.0	0	21.33
4	0.90	318.5	7.0	2	0.0	0	28.28
...	...	...	...	...	...	...	...
763	0.64	343.0	3.5	5	0.4	5	21.40
764	0.62	367.5	3.5	2	0.4	5	16.88
765	0.62	367.5	3.5	3	0.4	5	17.11
766	0.62	367.5	3.5	4	0.4	5	16.61
767	0.62	367.5	3.5	5	0.4	5	16.03

768 rows × 7 columns

شکل ۲-۱ داده ها پس از حذف برخی ستون ها

در ادامه برای بهبود عملکرد از آن‌جایی که ویژگی‌ها داده‌های مختلفی را دربرمی‌گیرند، لازم است تبدیل و مقیاس‌بندی انجام پذیرد. در تصویر ۲-۲ فرآیند انجام این مرحله نمایش داده شده است. مقیاس‌بندی داده‌ها باعث می‌شود که تمام ویژگی‌ها در یک بازه مشخص مقیاس شوند و مدل‌ها بهتر بتوانند الگوها و روابط بین ویژگی‌ها را شناسایی کنند.

```
scaler = StandardScaler()
X = data_cleaned.drop(columns=['Cooling_Load'])
y = data_cleaned['Cooling_Load']
data_scaled_x = scaler.fit_transform(X)
print(data_scaled_x)
```

✓ 0.0s

```
[[ 2.04177671 -0.56195149  1.         -1.34164079 -1.76044698 -1.81457514]
 [ 2.04177671 -0.56195149  1.         -0.4472136  -1.76044698 -1.81457514]
 [ 2.04177671 -0.56195149  1.          0.4472136  -1.76044698 -1.81457514]
 ...
 [-1.36381225  1.12390297 -1.         -0.4472136   1.2440492   1.41133622]
 [-1.36381225  1.12390297 -1.          0.4472136   1.2440492   1.41133622]
 [-1.36381225  1.12390297 -1.          1.34164079  1.2440492   1.41133622]]
```

شکل ۲-۲ مقیاس بندی داده ها

در نهایت مطابق با تصویر ۲-۳، داده‌ها را به مجموعه‌های آموزشی و آزمایشی تقسیم می‌کنیم تا بتوانیم مدل‌ها را آموزش داده و عملکرد آن‌ها را ارزیابی کنیم.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(X_test.shape[0])
print(X_train.shape[0])
```

✓ 0.0s

```
154
614
```

شکل ۲-۳ تقسیم داده ها

۳- مدل‌سازی

در این بخش، مدل‌های یادگیری ماشین مختلف شامل شبکه عصبی، جنگل تصادفی و XGBoost را آموزش می‌دهیم و سپس آن‌ها را ارزیابی می‌کنیم. هدف این بخش، ارزیابی عملکرد این مدل‌ها و مقایسه دقت آن‌ها با استفاده از معیارهای مختلف است. این مرحله به ما کمک می‌کند تا بهترین مدل را برای پیش‌بینی و تحلیل داده‌ها انتخاب کنیم.

۳-۱- شبکه عصبی

شبکه عصبی به عنوان یک مدل قوی برای مسائل پیچیده استفاده می‌شود. در تصویر ۳-۱، یک شبکه عصبی ساده با استفاده از پکیج Scikit-Learn با یک لایه پنهان تعریف کرده‌ایم و آن را با داده‌های آموزشی، در حداکثر ۵۰۰ دور، آموزش

می‌دهیم. سپس مدل را با داده‌های آزمایشی ارزیابی کرده و معیارهای MAE و RMSE را محاسبه می‌کنیم تا به عملکرد این مدل بر روی داده تست پی ببریم. که همانطور که از نتایج حاصل از ارزیابی مشاهده می‌شود این مدل عملکرد به نسبت خوبی داشته است.

```
mlp_model = MLPRegressor(hidden_layer_sizes=(100,), activation='relu', solver='adam', max_iter=500, random_state=42)

mlp_model.fit(X_train, y_train)

y_pred_mlp = mlp_model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred_mlp)
rmse = mean_squared_error(y_test, y_pred_mlp, squared=False)
print(f'Neural Network (MLP) - MAE: {mae}, RMSE: {rmse}')
```

✓ 1.7s

```
Neural Network (MLP) - MAE: 2.903109277895018, RMSE: 3.7692115382552465
```

شکل۳-۱ مدل شبکه عصبی

۳-۲- جنگل تصادفی

مدل جنگل تصادفی به دلیل قابلیت بالا در دستیابی به دقت مناسب و مقاومت در برابر بیش‌برازش، برای بسیاری از مسائل کاربردی مناسب است. در تصویر ۳-۲، با استفاده از ۱۰۰ درخت تصمیم‌گیری، مدل را آموزش داده و ارزیابی کرده‌ایم، که عملکرد مدل خیلی خوب بوده است.

```
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

rf_model.fit(X_train, y_train)

y_pred = rf_model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f'Random Forest - MAE: {mae}, RMSE: {rmse}')
```

✓ 0.5s

```
Random Forest - MAE: 1.0648337662337661, RMSE: 1.707626691084412
```

شکل۳-۲ مدل جنگل تصادفی

۳-۳- XGBoost

XGBoost به عنوان یکی از الگوریتم‌های قدرتمند برای مدل‌سازی داده‌ها شناخته می‌شود که از بهبودهای مختلف در الگوریتم‌های Boosting بهره می‌برد. این مدل با استفاده از پکیج Xgboost ایجاد و با هدف کاهش خطای مربعی آموزش داده شده و سپس ارزیابی شده است. پیاده‌سازی این مدل در تصویر ۳-۳ قابل رویت است.

```
xgb_model = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=100, random_state=42)

xgb_model.fit(X_train, y_train)

y_pred = xgb_model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f'XGBoost - MAE: {mae}, RMSE: {rmse}')
```

✓ 4.0s

```
XGBoost - MAE: 0.4396061146723759, RMSE: 0.8094383190341791
```

شکل ۳-۳ مدل XGBoost

٤- تحلیل نتایج

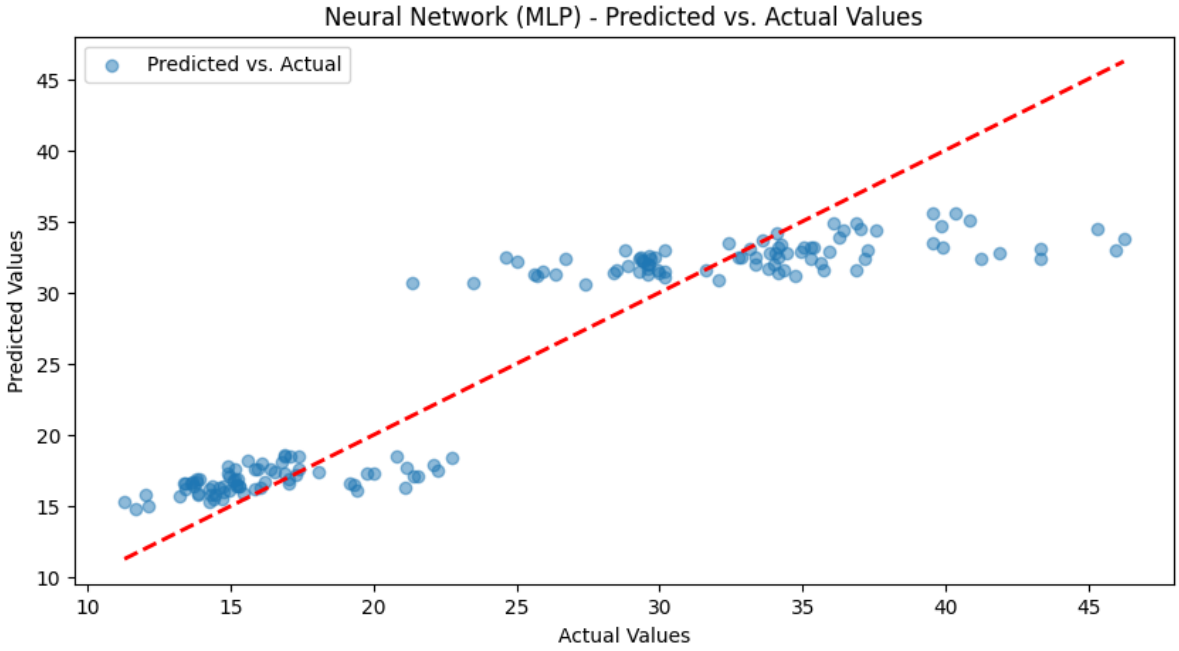
در این بخش، به بررسی و تحلیل نتایج حاصل از ارزیابی هر سه مدل خواهیم پرداخت و بیان خواهیم کرد که کدام مدل نسبت به سایرین عملکرد بهتری داشته است. همانطور که در بخش قبل مشاهده شد، برای هر سه مدل اقدام به محاسبه دو پارامتر میانگین خطای مطلق [۱] و ریشه میانگین مربعات خطا[۲] کرده‌ایم. این دو پارامتر بیانگر میزان خطاهای مدل ما در پیش‌بینی صفت هدف برای داده‌های آزمایش است و به طور قطع کمتر بودن این مقادیر بیانگر بهتر بودن عملکرد مدل است. در جدول ۱-۴ تمامی این مقادیر برای هر سه مدل مجدد گردآوری شده و با تقریب نمایش داده شده است.

جدول ۴-۱ ارزیابی سه مدل پیاده‌سازی شده

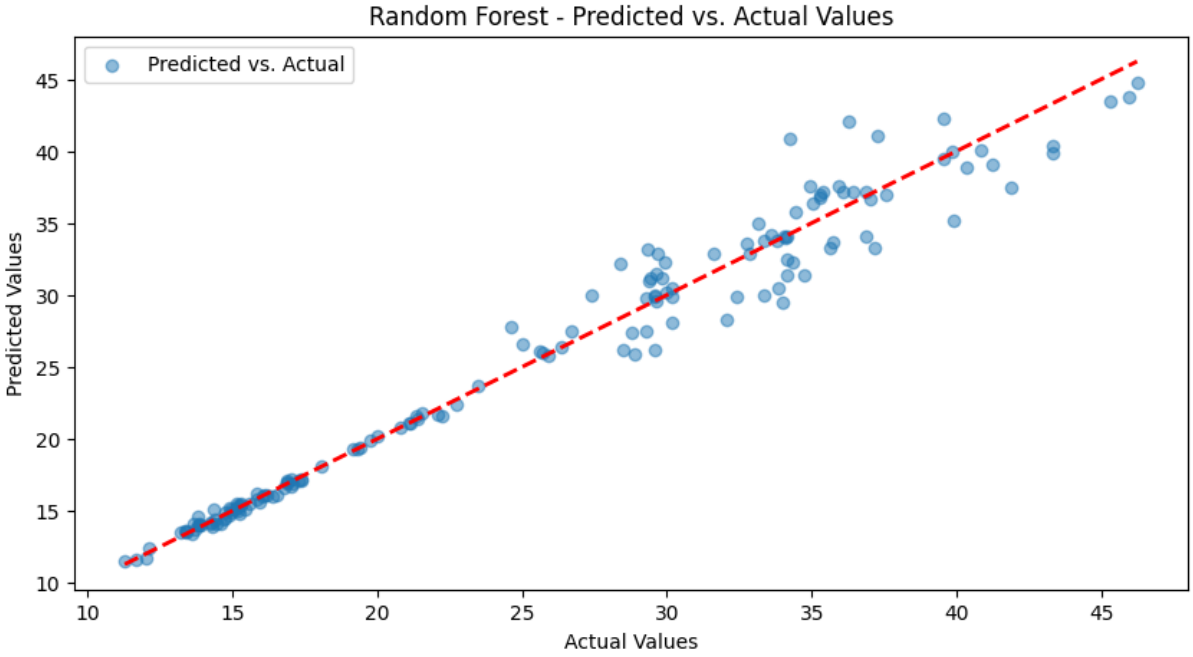
نام الگوریتم	MAE	RMSE
شبکه عصبی	۲,۹۰۳۱	۳,۷٦۹۲۱
جنگل تصادفی	۱,۰٦٤۸	۱,۷۰۷٦
XGBoost	۰,٤۳۹٦	۰,۸۰۹٤

علاوه بر محاسبه این سه پارامتر، می‌توان عملکرد مدل را نیز بصری‌سازی کرد. به این منظور نموداری برای نمایش مقدار واقعی و مقدار پیش بینی شده برای هر سه مدل در تصاویر ۴-۱، ۴-۲ و ۴-۳ موجود است که به ترتیب مربوط به مدل‌های شبکه عصبی، جنگل تصادفی و XGBoost است.

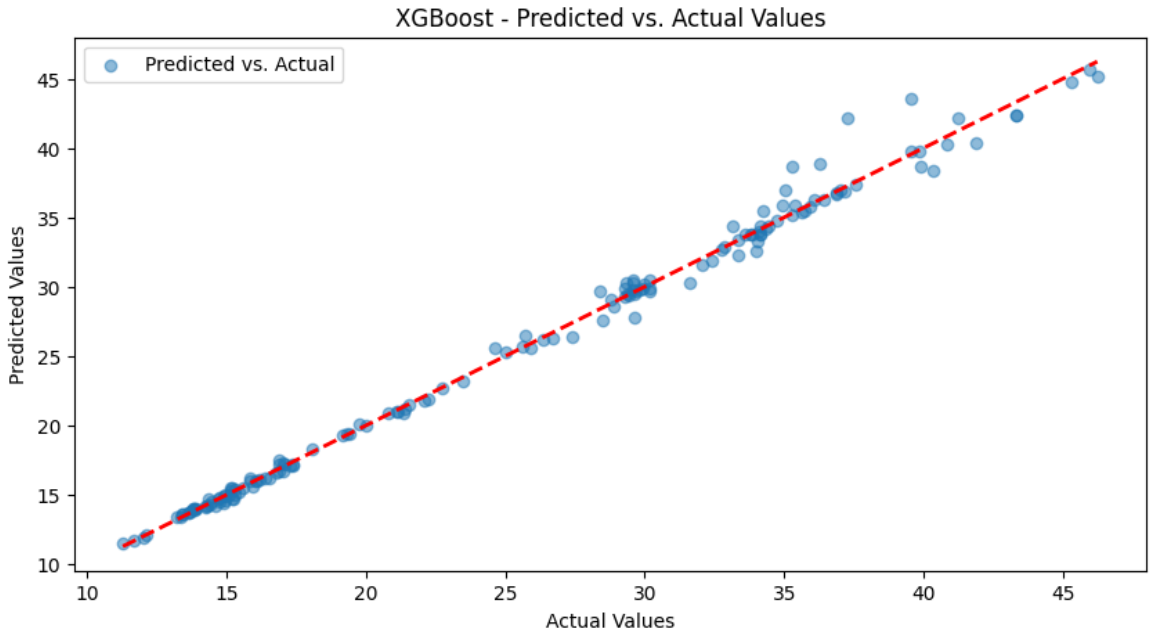
[۱] MAE
[۲] RMSE



شکل ۴-۱ نمودار مدل شبکه عصبی



شکل۴-۲ نمودار مدل جنگل تصادفی



شکل۴-۳ نمودار مدل XGBoost

همانطور که از نمودارها و اعداد موجود در جدول مشخص است:

- شبکه عصبی در مقایسه با دو مدل دیگر عملکرد ضعیف‌تری دارد. مقدار MAE برابر ۲.۹۰۳ و RMSE برابر ۳.۷۶۹ نشان می‌دهد که مدل دقت کمتری در پیش‌بینی‌ها دارد. همانطور که در نمودار مربوطه مشاهده می‌شود، نقاط پیش‌بینی شده نسبت به خط ۴۵ درجه که نشان‌دهنده پیش‌بینی‌های کامل و دقیق است، پراکندگی بیشتری دارند. این نشان می‌دهد که مدل شبکه عصبی (MLP) توانایی کمتری در تطابق با داده‌های واقعی دارد.
- مدل جنگل تصادفی عملکرد بهتری نسبت به شبکه عصبی دارد. مقدار MAE برابر ۱.۰۶۵ و RMSE برابر ۱.۷۰۸ نشان می‌دهد که مدل دقت بیشتری در پیش‌بینی‌ها دارد. در نمودار مربوط به جنگل تصادفی، نقاط پیش‌بینی شده به خط ۴۵ درجه نزدیک‌تر هستند که نشان‌دهنده دقت بالاتر این مدل در مقایسه با شبکه عصبی است.
- مدل XGBoost بهترین عملکرد را در بین این سه مدل دارد. مقدار MAE برابر ۰.۴۴۰ و RMSE برابر ۰.۸۰۹ نشان‌دهنده دقت بسیار بالای این مدل در پیش‌بینی‌ها است. نمودار مربوط به XGBoost نشان می‌دهد که نقاط پیش‌بینی شده تقریباً منطبق بر خط ۴۵ درجه هستند، که نشان‌دهنده تطابق بسیار خوب مدل با داده‌های واقعی است.

۵- نتیجه‌گیری

به طورکلی از تمامی فرآیندهای انجام شده در بخش‌های قبل دریافتیم که با داشتن برخی ویژگی‌های یک خانه مسکونی می‌توان میزان مصرف انرژی برای سرمایش و گرمایش آن را تخمین زد. بنابراین با یافتن ویژگی‌های تاثیرگذار بر مصرف انرژی یک خانه می‌توانم با تمرکز بیشتر بر روی آن‌ها، مصرف انرژی یک ساختمان را بهینه کنیم. علاوه برآن با اجرای هر سه مدل بر روی مجموعه داده خود و ارزیابی آن‌ها دریافتیم که مدل XGBoost عملکرد بهتری در تخمین مصرف انرژی سرمایشی، یا گرمایشی، با داشتن ویژگی‌های مناسب از آن خانه داشته است.