گزارش کار

درس اینترنت اشیا

دکتر محمد یغمایی

پروژه :۹ هوشمند سازی و تحلیل داده ها

اعضا تیم :

مبینا بیژنی

شمیم انوری

سحر محمدی

فائزه سید موسوی

نیم سال دوم ۱۴۰۱

فهرست عناوین

۱. پیش بینی مصرف روز بعد کاربر و محاسبه خطای پیش بینی

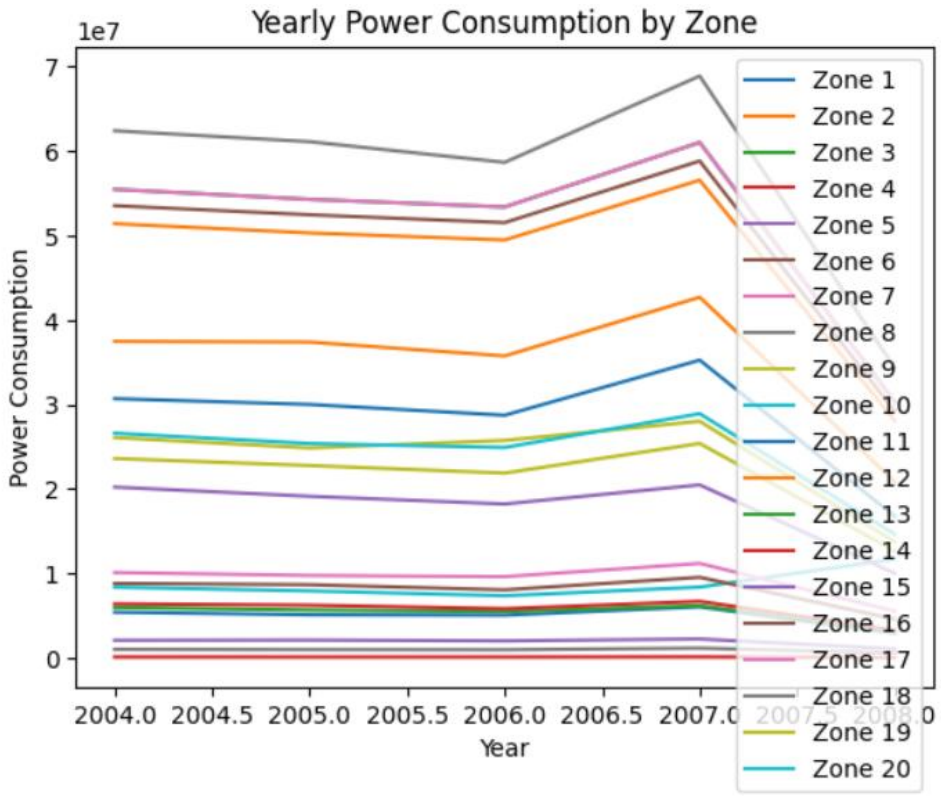
داده های مورد بررسی، مقادیر مصرف برق ۲۰ خانوار در طی سال های ۲۰۰۴ تا ۲۰۰۸ می باشد بصورت وات بر ساعت یعنی بطور روزانه در ۲۴ ساعت شبانه روز اندازه گیری شده است.

برای شناسایی بهتر داده ها انها را در جدول و چند نمودار مختلف نشان میدهیم.

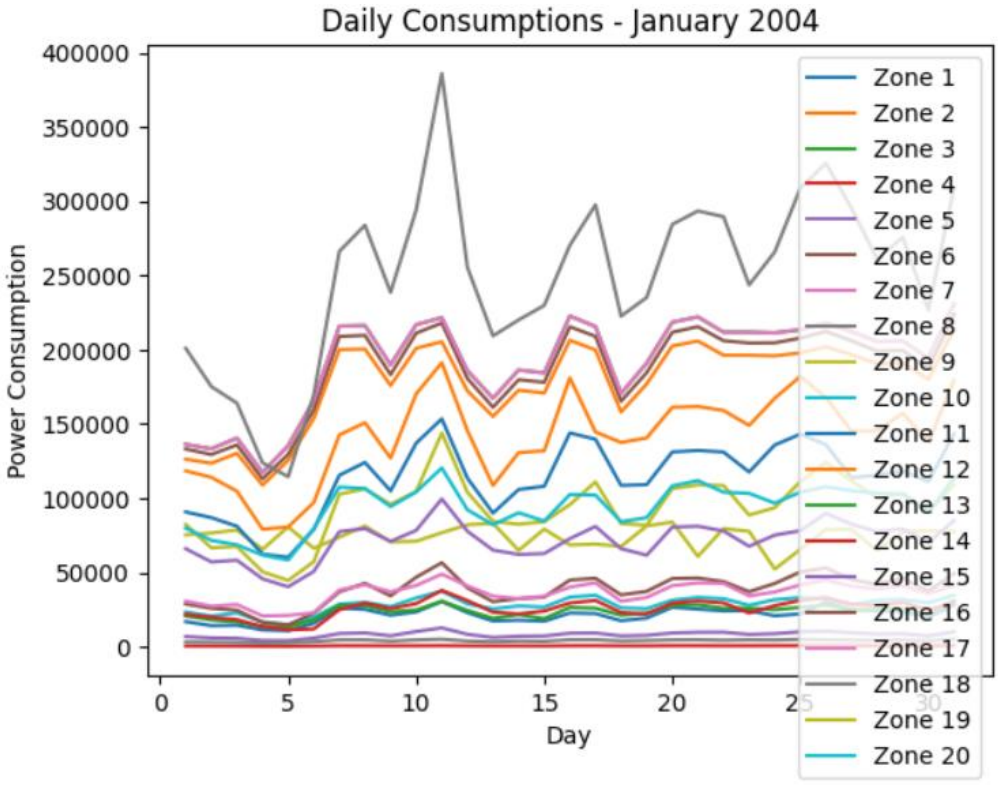
```
df.head()
```

	zone_id	year	month	day	h1	h2	h3	h4	h5	h6	...	h15	h16	h17	h18	h19	h20	h21	h22	h23	h24
0	1	2004	1	1	16853	16450	16517	16873	17064	17727	...	13518	13138	14130	16809	18150	18235	17925	16904	16162	14750
1	1	2004	1	2	14155	14038	14019	14489	14920	16072	...	16127	15448	15839	17727	18895	18650	18443	17580	16467	15258
2	1	2004	1	3	14439	14272	14109	14081	14775	15491	...	13507	13414	13826	15825	16996	16394	15406	14278	13315	12424
3	1	2004	1	4	11273	10415	9943	9859	9881	10248	...	14207	13614	14162	16237	17430	17218	16633	15238	13580	11727
4	1	2004	1	5	10750	10321	10107	10065	10419	12101	...	13845	14350	15501	17307	18786	19089	19192	18416	17006	16018

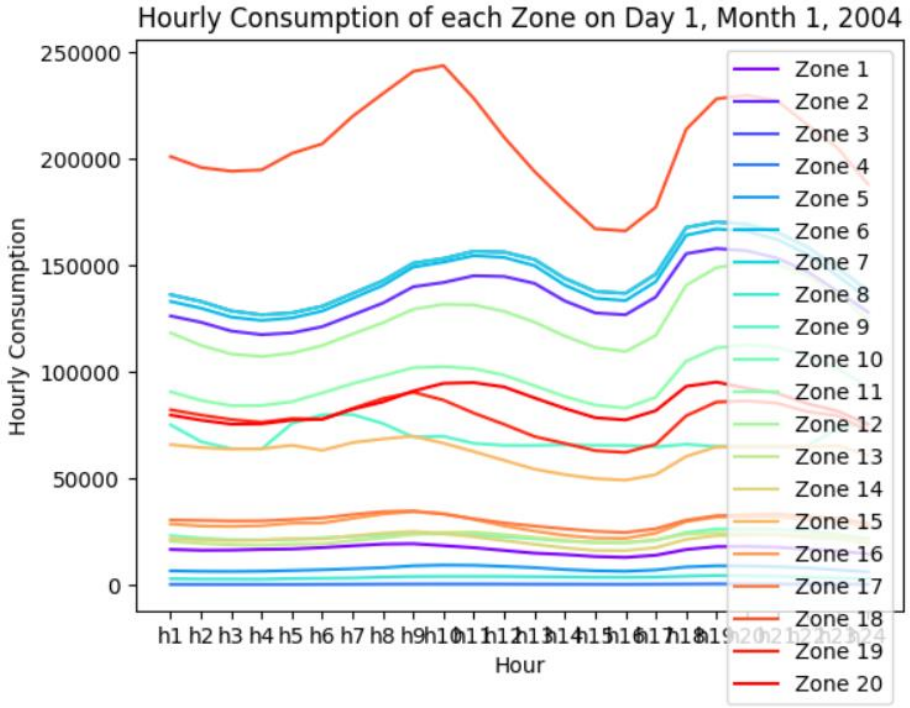
جدول۱.جدول نمونه داده ها



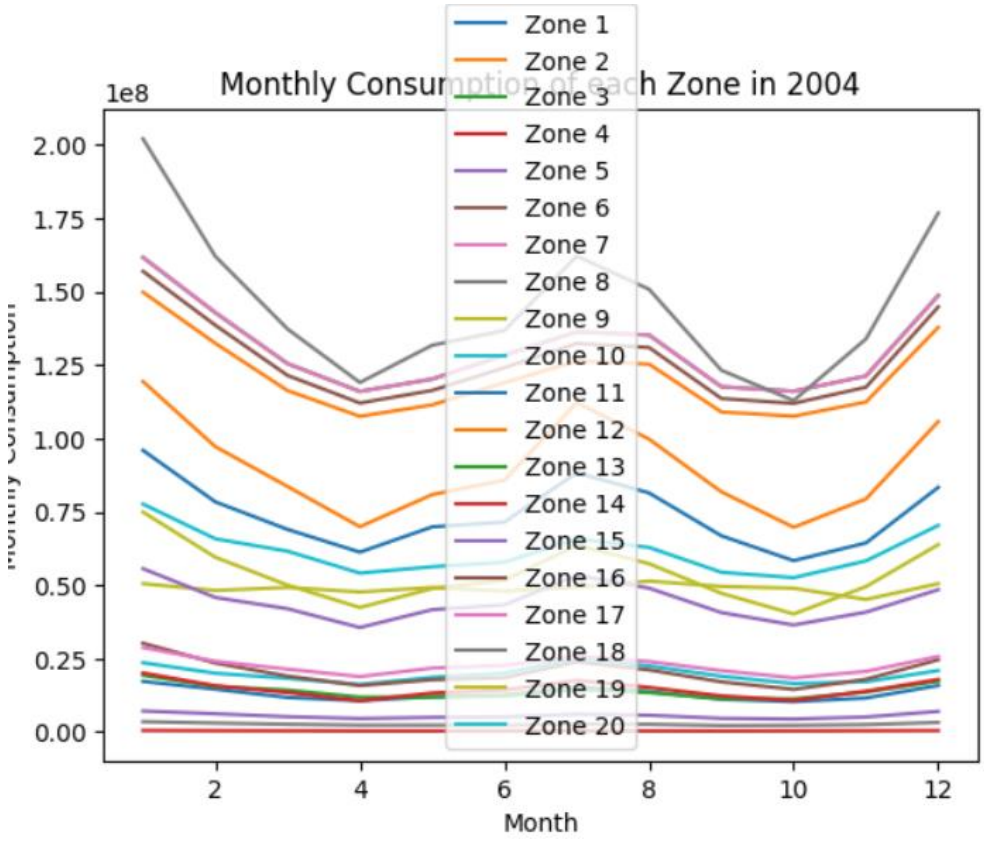
نمودار۱. مصرف انرژی(وات) هر کاربر بر اساس سال



نمودار۲. مصرف انرژی(وات) هر کاربر بر اساس روز در ژانویه ۲۰۰۴



نمودار۳. مصرف انرژی(وات) هر کاربر بر اساس ساعت در ۱ ژانویه ۲۰۰۴



نمودار۴. مصرف انرژی(وات) هر کاربر بر اساس ماه در سال ۲۰۰۴

برای اینکار ابتدا داده ها را پیش پردازش کردیم. سطر هایی که بدون مقدار بودند را حذف و "," را با تهی جایزگزین کردیم تا بتواند عبارت را بصورت عدد در نظر بگیرد و قابل استفاده برای محاسبات باشد.

بطور کلی به منظور پیشبینی، از یک مدل یادگیری ماشین استفاده کردیم که با ۸۰ درصد داده ها learn می‌شود و برای ۲۰ درصد دیگر predictمی‌کند.

خانوار ها یا کاربران مختلف با zone_id مختلف مشخص شده اند. پس مقادیر ستون های zone_id, year, month, day را بعنوان ویژگی ها در نظر میگیریم که یعنی طبق این موارد شرایط تغییر میکند. و مقادیر ستون های h1 تا h24 را بعنوان هدف قرار دادیم که این مقادیر باید پیشبینی شوند.

سپس داده ها را به کمک MinMaxScaler نرمال کرده و به مقادیر بین ۰ تا ۱ تبدیل میکنیم.

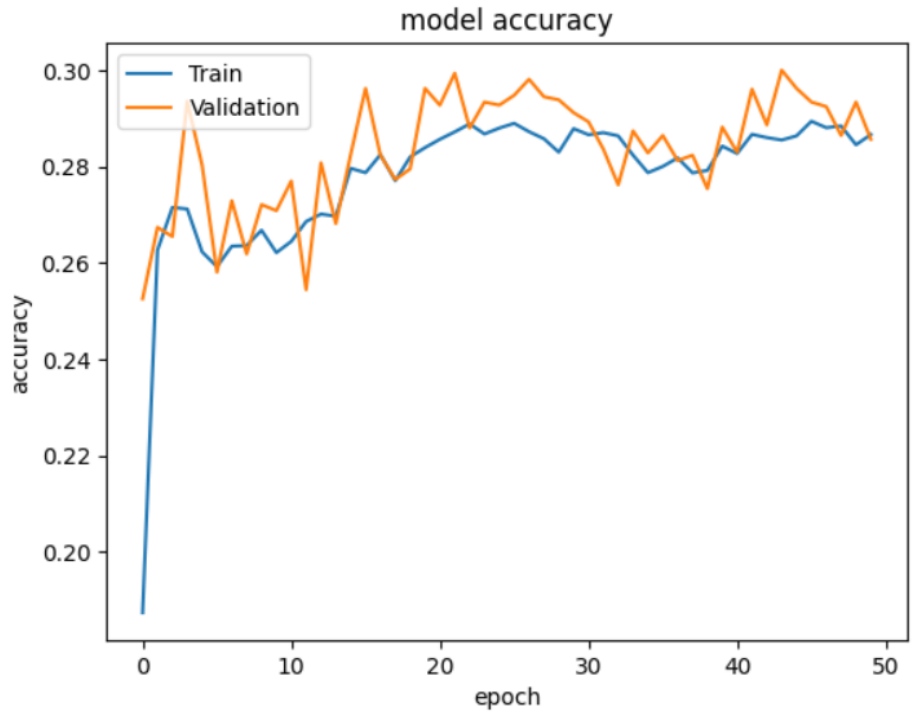
برای اینکار از دو مدل LSTM و Seq2Seq-BiLSTM استفاده کردیم. در سال های اخیر محققان بسیاری از شبکه های بازگشتی یا (RNN) برای پیشبینی مصرف انرژی استفاده کرده‌اند. شبکه های 11(GRU)، RNN،(LSTM(زیرمجموعه شبکه های عصبی بازگشتی جهت پردازش داده‌های سری زمانی و استخراج الگوهای توالی می‌باشند. شبکه RNN دارای دو ورودی حافظه و ورودی اصلی است که به ورودی حافظه hidden stateمیگویندRNN .دنباله ای از دادههای ورودی را با استفاده از حافظه داخلیاش پردازش میکند و دچار مشکل محوشدگی گرادیان میشود به این دلیل که این شبکه نمیتواند از وابستگیهای بلندمدت پشتیبانی کند بنابراین روی دقت پیشبینی اثر

منفی میگذارد. برای حل مشکلات گفته شده شبکههای LSTM که شامل گیتهای ورودی ،خروجی و فراموشی هستند جهت استخراج توالی در دادههای سری زمانی و به خاطر سپردن دادهه ای گذشته در حافظه به وجود آمده اند.

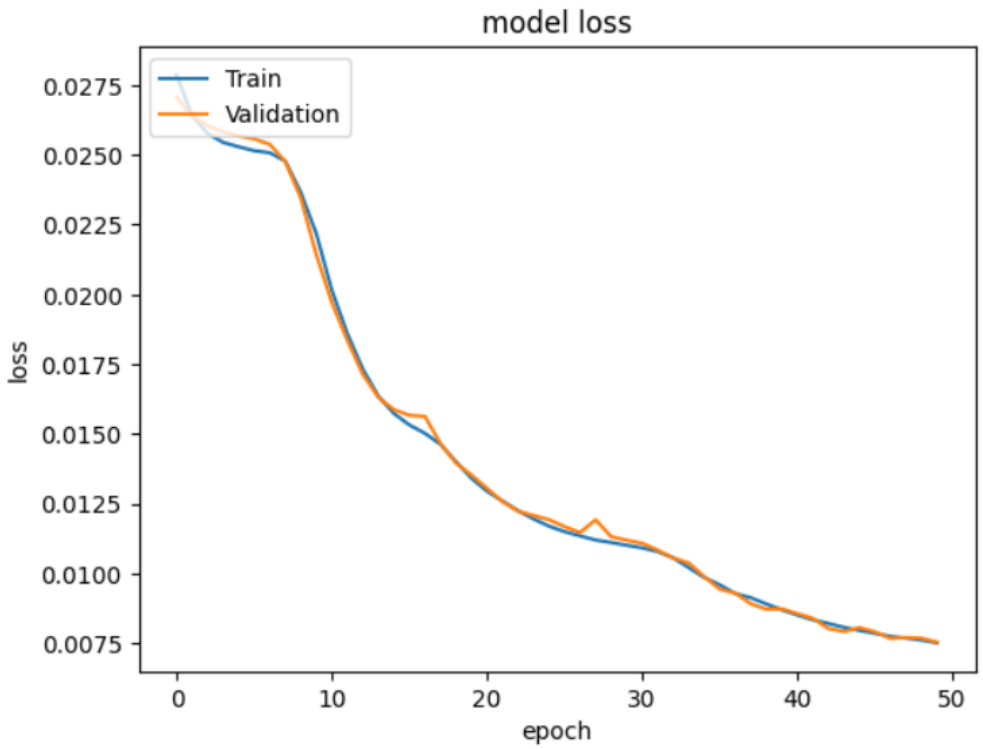
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_1 (LSTM)               (None, 50)                11000

 dense_1 (Dense)             (None, 24)                1224

=================================================================
Total params: 12,224
Trainable params: 12,224
Non-trainable params: 0
_________________________________________________________________
```

عکس۱. Model summary



نمودار۴. پیشرفت دقت مدلLSTM



نمودار۵. LSTM model loss

برای محاسبه خطای پیشبینی، MAE یا Mean Absolute Error را محاسبه کردیم.

نتیجه 6٪ شد و وقتی مقادیر را reverseScale کردیم و MAE برابر 27622.276133289193 شد.

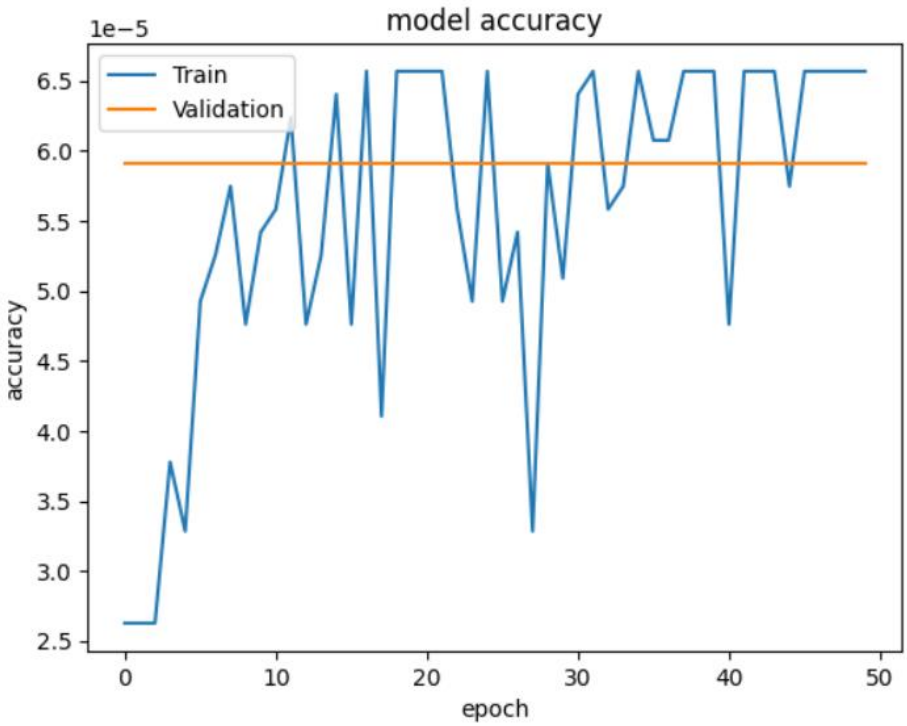
در مقایسه با میانگین داده ها که برابر 70921.46125472887 می باشد، تقریبا ارور قابل قبولی است. البته میتواند بهتر باشد.

Seq2Seq-BiLSTM

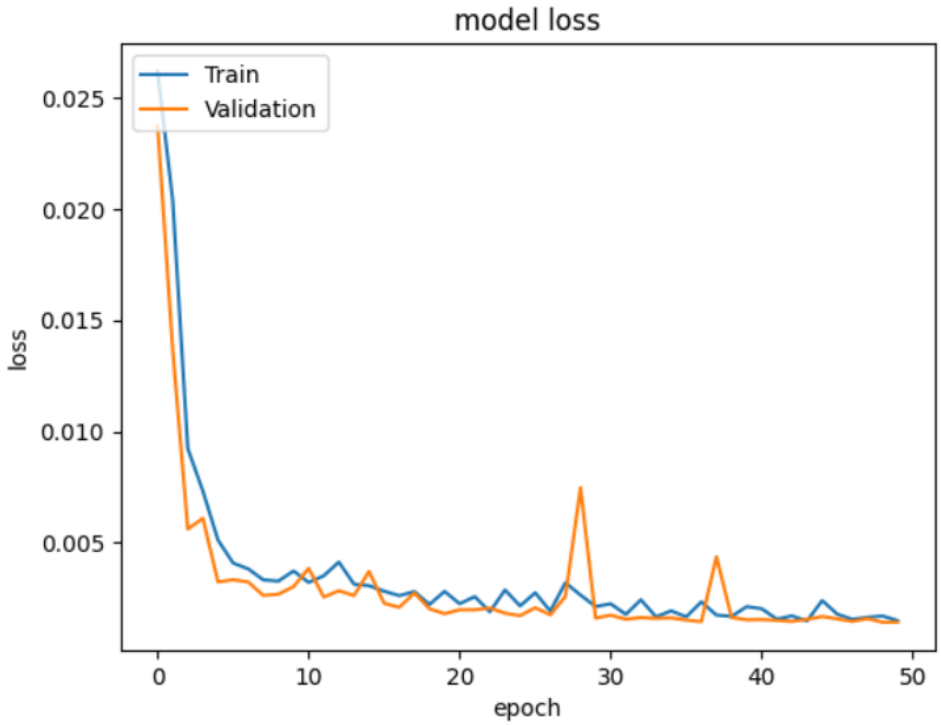
یک مدل عمیق (deep model) برای ترجمه ماشینی و وظایف تولید دنباله دیگر است. این مدل از دو قسمت اصلی تشکیل شده است: مدل ترجمه از دنباله به دنباله (Sequence-to-Sequence) یا Seq2Seq و شبکه عصبی برگشتی با لایه‌های LSTM (BiLSTM).

مدل Seq2Seq-BiLSTM از ساختار کدگذار–کدگشا استفاده می‌کند. در مرحله کدگذاری، دنباله ورودی (مانند جملات در ترجمه ماشینی) با استفاده از شبکه عصبی برگشتی با لایه‌های LSTM کدگذاری می‌شود. این لایه‌های LSTM اطلاعات جریان پیشین را به خروجی‌های فعلی متصل می‌کنند. برای بهره‌برداری از اطلاعات قبل و بعد در دنباله ورودی، از لایه‌های LSTM دوطرفه (BiLSTM) استفاده می‌شود که برای هر لحظه زمانی، یک لایه LSTM اطلاعات از جلو به عقب و یک لایه LSTM دیگر اطلاعات از عقب به جلو را بررسی می‌کند.

سپس، در مرحله کدگشایی، دنباله نهایی از مدل کدگذار به عنوان ورودی به مدل کدگشا (مانند جملات هدف در ترجمه ماشینی) می‌رسد. مدل کدگشا نیز از لایه‌های LSTM برای تولید دنباله خروجی استفاده می‌کند. این لایه‌های LSTM براساس اطلاعات دریافتی از مدل کدگذار و وضعیت داخلی خود، هر کلمه یا توکن خروجی را تولید می‌کنند. این فرایند به طور تکراری انجام می‌شود تا دنباله خروجی کامل شود.



نمودار۶. پیشرفت دقت مدل Seq2Seq-BiLSTM



نمودار۷. Seq2Seq-BiLSTM model loss

در اینجا MAE برابر 2% و بعد از reverseScale کردن برابر 10629.870308800702 شد. که بصورت چشمگیری کاهش یافت و بهتر از LSTM معمولی جواب داد.

۲. پیش بینی مصرف کل ماه آینده کاربر و محاسبه خطای پیش بینی

برای اینکار ابتدا ستونی در جدول اضافه کردیم بنام total_consumption که مجموع مصرف هر روز را در آن ذخیره کردیم.

ابتدا ۸۰ درصد داده ها را train و ۲۰ درصد دیگر را test در نظر گرفتیم.

در اینجا از مدل LSTM استفاده کردیم.

MAE برابر ۲۱٪ شد.

در حالت بعدی داده های ۱۵ روز اول ماه را train و ۱۵ روز بعدی را test در نظر گرفتیم.

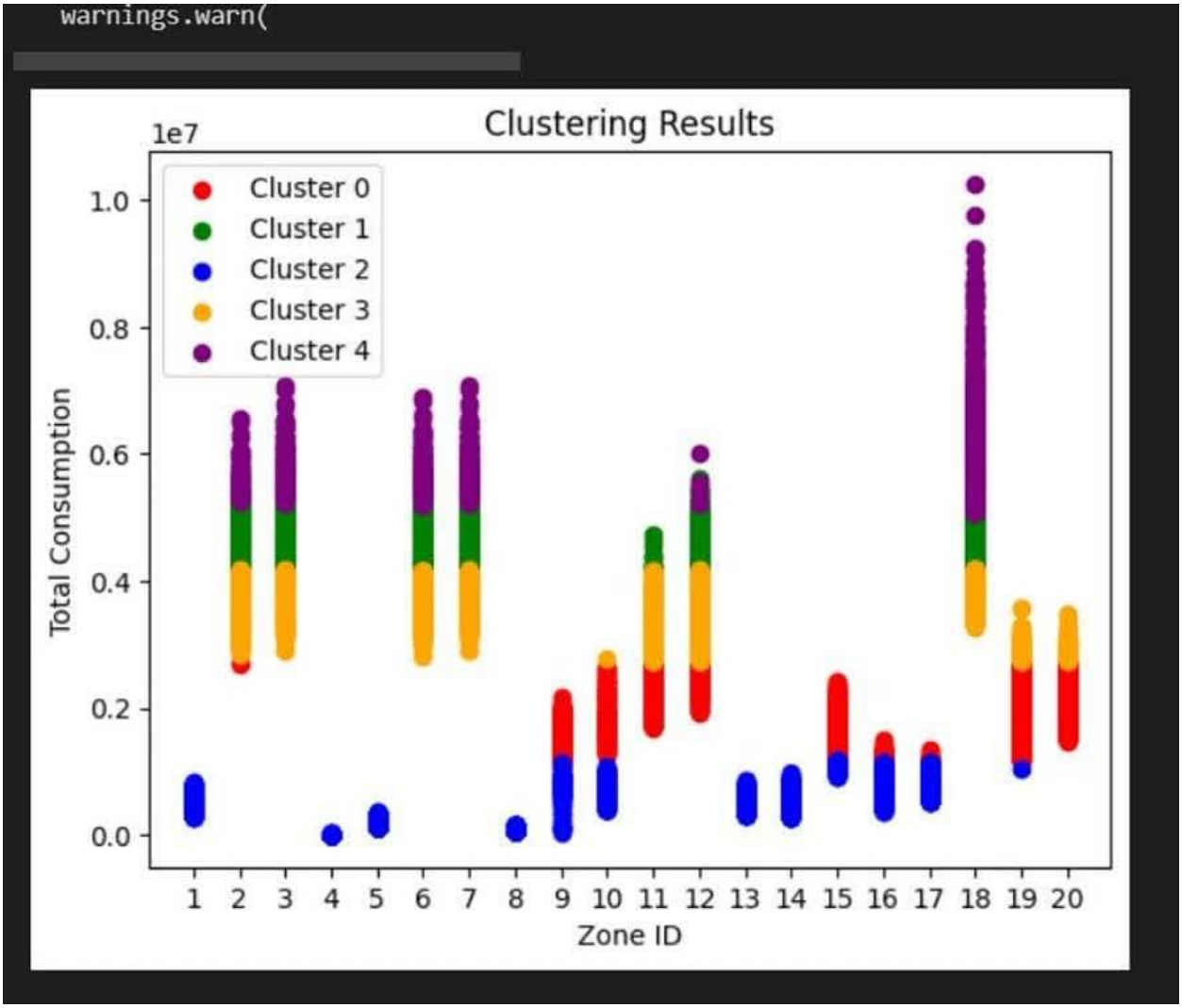
در این حالت MAE برابر ۱۴٪ شد.

۳. خوشه بندی کاربران

این کد، خوشه‌بندی K-means را روی مجموعه داده‌ای انجام می‌دهد که از یک فایل CSV به نام « «month.csvخوانده می‌شود. ستون های مربوطه برای خوشه بندی با استفاده از StandardScaler انتخاب و استاندارد می شوند. الگوریتم KMeans با ۵ خوشه اعمال می شود و برچسب های خوشه به DataFrame اصلی اضافه می شوند. در نهایت، خوشه ها با رنگ های مختلف رسم می شوند و zone_id های مربوطه نمایش داده می شوند. مکان ها و برچسب های تیک محور x برای تجسم بهتر تنظیم شده اند. نمودار حاصل نتایج خوشه بندی را با رنگ های مختلف نشان می دهد که خوشه های مختلف را نشان می دهد.

مطمئنا، در اینجا توضیح دقیق تری از کد ارائه شده است:

خوشه بندی بر اساس مصرف با استفاده از k-means k=5



۱. وارد کردن کتابخانه های لازم:

– پاندا برای دستکاری و تجزیه و تحلیل داده ها استفاده می شود

– NumPy برای عملیات ریاضی استفاده می شود

– Matplotlib برای تجسم داده ها استفاده می شود

– KMeans و StandardScaler به ترتیب از scikit-learn برای خوشه بندی و پیش پردازش داده ها استفاده می شوند.

۲. خواندن فایل :CSV

– فایل ' 'month.csvبا استفاده از تابع ()read_csvاز کتابخانه Pandas خوانده می شود و در یک شی DataFrame به نام ' 'dfذخیره می شود.

۳. فیلتر کردن ستون های مربوطه:

– ستون های مربوطه برای خوشه بندی با استفاده از تابع []locروی شی 'DataFrame 'dfانتخاب شده و در یک شی DataFrame جدید به نام ' 'dataذخیره می شوند. در اینجا، ستون های ' 'h1تا ' 'h24انتخاب می شوند.

۴. استانداردسازی داده ها:

– تابع ()StandardScalerاز scikit-learn برای استانداردسازی داده ها با مرکزیت و مقیاس گذاری آن استفاده می شود. داده های استاندارد شده در یک شیء DataFrame جدید به نام " "data_scaledذخیره می شوند.

۵. انجام K-means خوشه بندی:

– تابع ()KMeansاز scikit-learn برای انجام خوشه بندی K-means بر روی داده های استاندارد شده ' 'data_scaledبا ۵ خوشه و random_state=42 استفاده می شود. خوشه های به دست آمده در آرایه ای به نام 'خوشه' ذخیره می شوند.

۶. افزودن برچسب های خوشه ای به DataFrame اصلی:

– برچسب‌های خوشه‌ای به‌دست‌آمده از خوشه‌بندی K-means به عنوان یک ستون جدید به نام «خوشه» به شی اصلی DataFrame 'df'اضافه می‌شوند.

۷. رسم خوشه ها:

– خوشه ها با استفاده از تابع ()scatterاز کتابخانه Matplotlib رسم می شوند. محور x نشان دهنده ' 'zone_idو محور y نشان دهنده ' 'total_consumptionاست. هر خوشه با رنگی متفاوت ترسیم می شود و zone_id های مربوطه نمایش داده می شوند.

۸. تنظیم مکان ها و برچسب های تیک محور :x

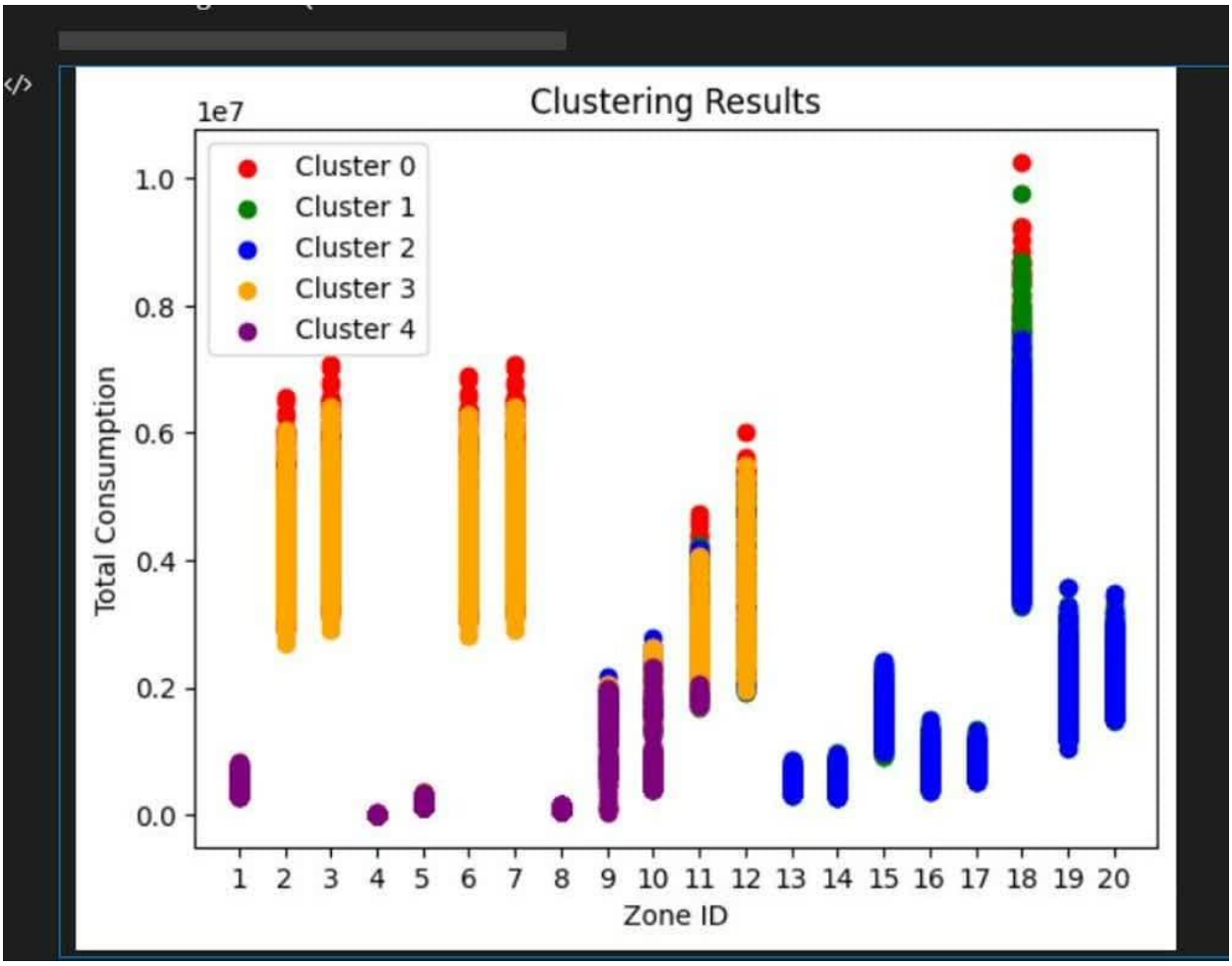
– مکان ها و برچسب های تیک محور x با استفاده از تابع ()xticksاز کتابخانه Matplotlib تنظیم می شوند. در اینجا، np.arange(1، ۲۱) برای تنظیم مکان تیک از ۱ تا ۲۰ استفاده می شود.

۹. نمایش طرح:

– در نهایت نمودار با استفاده از تابع ()showاز کتابخانه Matplotlib نمایش داده می شود.

خوشه بندی بر اساس مصرف و تاریخ

این کد به روز شده، خوشه بندی K-means را روی مجموعه داده ای که از یک فایل CSV به نام « «month.csvخوانده می شود، انجام می دهد. ستون های مربوطه برای خوشه بندی با استفاده از StandardScaler انتخاب و استاندارد می شوند. علاوه بر این، ویژگی‌های ''zone_id، 'year'، 'month' و ' 'dayبه عنوان ویژگی‌های اضافی به داده‌های استاندارد شده الحاق می‌شوند. الگوریتم KMeans با ۵ خوشه اعمال می شود و برچسب های خوشه به DataFrame اصلی اضافه می شوند. در نهایت، خوشه ها با رنگ های مختلف رسم می شوند و zone_id های مربوطه نمایش داده می شوند. مکان ها و برچسب های تیک محور x برای تجسم بهتر تنظیم شده اند. نمودار حاصل نتایج خوشه بندی را با رنگ های مختلف نشان می دهد که خوشه های مختلف را نشان می دهد.



در اینجا توضیح دقیق تری در مورد کد به روز شده آورده شده است:

۱. خواندن فایل :CSV

۲. فیلتر کردن ستون های مربوطه:

۴. استانداردسازی داده ها:

۵. الحاق ویژگی های اضافی:

– ویژگی های ''zone_id'، 'year'، 'month' و ' 'dayاز شی اصلی 'df' DataFrameانتخاب شده و در یک شی DataFrame جدید به نام ' 'additional_featuresذخیره می شوند. سپس این ویژگی ها با استفاده از تابع np.concatenate ()از کتابخانه NumPy به داده های استاندارد شده ' 'data_scaledالحاق می شوند.

۶. انجام K-means خوشه بندی:

۷. افزودن برچسب های خوشه ای به DataFrame اصلی:

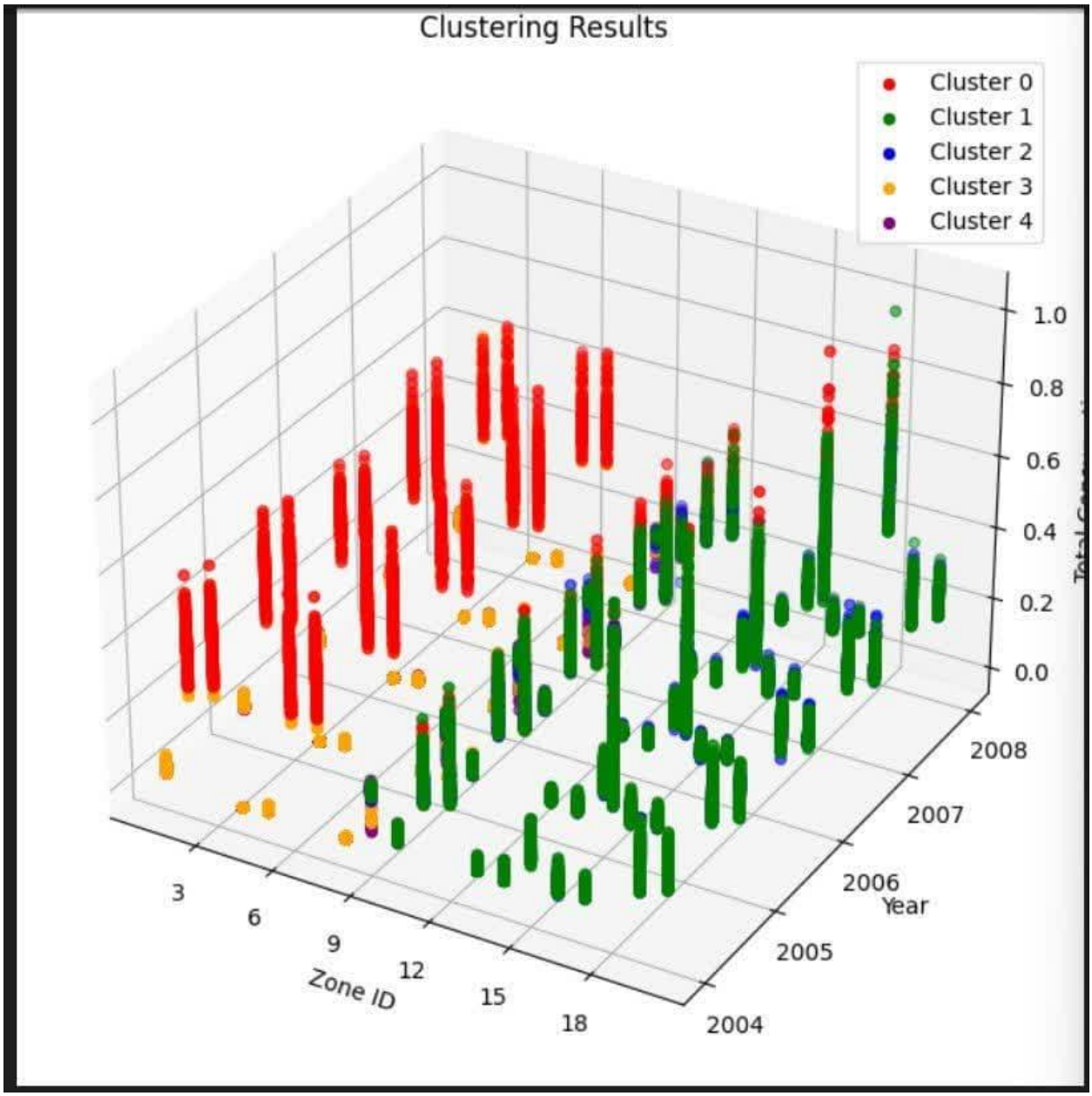
۸. رسم خوشه ها:

۹. تنظیم مکان ها و برچسب های تیک محور :x

۱۰. نمایش طرح:

نمودار سه بعدی

این کد به روز شده، خوشه بندی K-means را روی مجموعه داده ای که از یک فایل CSV به نام « «month.csvخوانده می شود، انجام می دهد. ستون های مربوطه برای خوشه بندی با استفاده از StandardScaler انتخاب و استاندارد می شوند. علاوه بر این، ویژگی‌های ''zone_id'، 'year'، 'month' و ' 'dayبه عنوان ویژگی‌های اضافی به داده‌های استاندارد شده الحاق می‌شوند. الگوریتم KMeans با ۵ خوشه اعمال می شود و برچسب های خوشه به DataFrame اصلی اضافه می شوند. در نهایت، خوشه ها در یک فضای سه بعدی با رنگ های مختلف نشان دهنده خوشه های مختلف رسم می شوند. نمودار حاصل نتایج خوشه بندی را با رنگ های مختلف نشان می دهد که خوشه های مختلف را نشان می دهد.



در اینجا توضیح دقیق تری در مورد کد به روز شده آورده شده است:

۱. وارد کردن کتابخانه های لازم

۲. خواندن فایل :CSV

۳. فیلتر کردن ستون های مربوطه:

۴. استانداردسازی داده ها:

۵. الحاق ویژگی های اضافی:

۶. انجام K-means خوشه بندی:

۷. افزودن برچسب های خوشه ای به DataFrame اصلی

۸. ایجاد طرح سه بعدی:

– یک نمودار سه بعدی بزرگتر با استفاده از تابع figure()از کتابخانه Matplotlib ایجاد می شود. تابع add_subplot()برای افزودن یک طرح سه بعدی به نمودار استفاده می شود.

۹. رسم خوشه ها در فضای سه بعدی:

– خوشه ها در یک فضای سه بعدی با استفاده از تابع scatter()از کتابخانه Matplotlib رسم می شوند. محور x نشان دهنده ''zone_id، محور y نشان دهنده ' 'yearو محور z نشان دهنده ' 'total_consumptionاست. هر خوشه با رنگی متفاوت ترسیم می شود و zone_id های مربوطه نمایش داده می شوند.

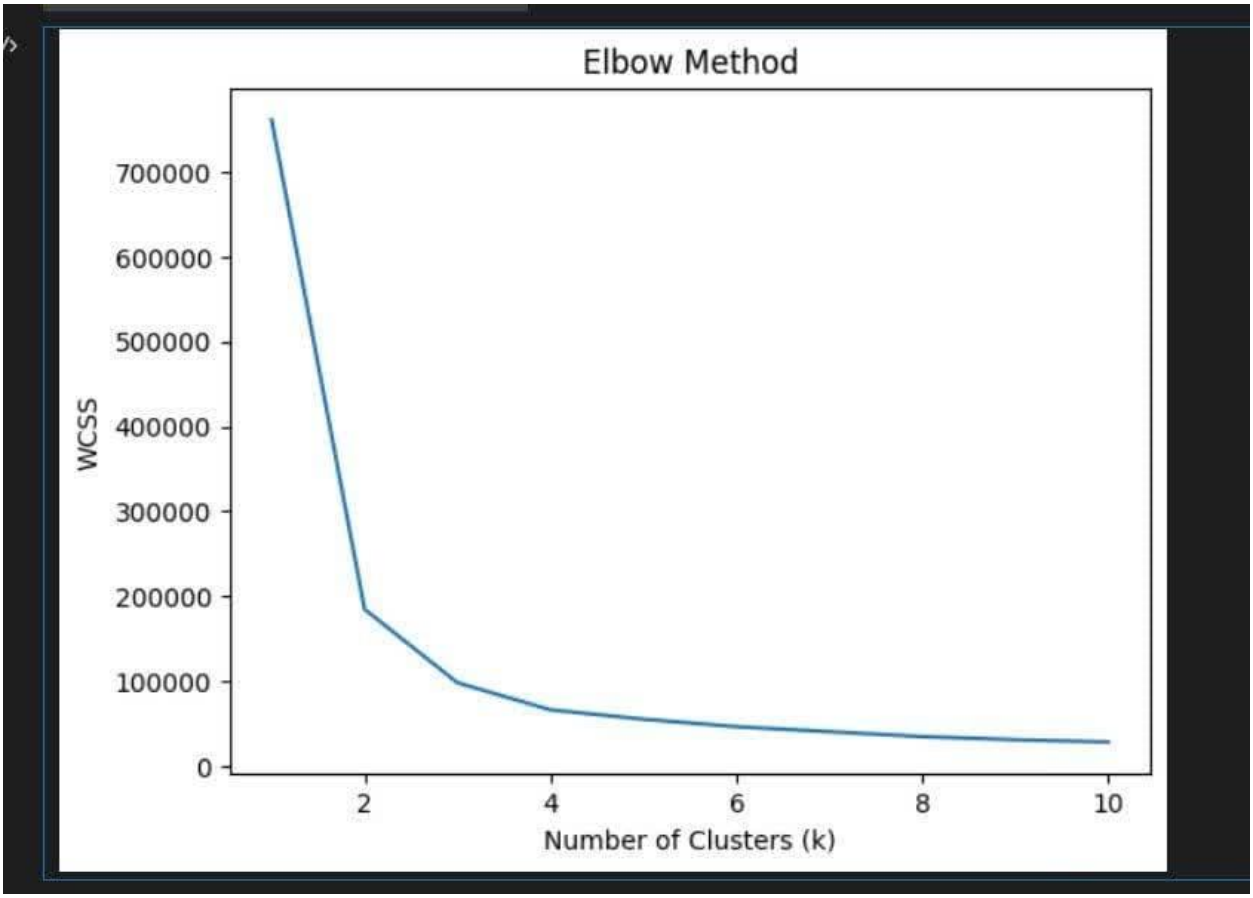
۱۰. تنظیم فرمت کننده های محور x و :y

– فرمت‌کننده‌های محور x و y با استفاده از تابع ()set_major_locator از کتابخانه Matplotlib.ticker تنظیم می‌شوند تا فقط مقادیر صحیح را نمایش دهند.

۱۱. نمایش طرح

Elbow

این کد، خوشه‌بندی K-means را روی مجموعه داده‌ای انجام می‌دهد که از یک فایل CSV به نام « «month.csvخوانده می‌شود. ستون های مربوطه برای خوشه بندی با استفاده از StandardScaler انتخاب و استاندارد می شوند. سپس، خوشه‌بندی K-means برای مقادیر مختلف k از ۱ تا ۱۰ انجام می‌شود. مجموع مربع‌های درون خوشه‌ای ((WCSSبرای هر مقدار k محاسبه می‌شود و در فهرستی به نام « «wcssذخیره می‌شود. در نهایت، مقادیر WCSS در برابر تعداد خوشه‌ها ((kرسم می‌شوند تا مقدار بهینه k با استفاده از روش زانویی تعیین شود.



در اینجا توضیح دقیق تری از کد ارائه شده است:

۱. وارد کردن کتابخانه های لازم:

۲. خواندن فایل :CSV

۳. فیلتر کردن ستون های مربوطه:

۴. استانداردسازی داده ها:

– تابع ()StandardScalerاز scikit-learn برای استانداردسازی داده ها با مرکزیت و مقیاس گذاری آن استفاده می شود. داده های استاندارد شده در یک شیء DataFrame جدید به نام " "data_scaledذخیره می شوند.

۵. انجام K-mean خوشه بندی برای مقادیر مختلف :k

– تابع ()rangeبرای ایجاد محدوده ای از مقادیر برای k از ۱ تا ۱۰ استفاده می شود.

– یک حلقه for برای تکرار بر روی هر مقدار از k و انجام خوشه بندی K-means با استفاده از تابع ()KMeansاز scikit-learn با random_state=42 استفاده می شود.

– مقدار WCSS برای هر مقدار k با استفاده از ویژگی _inertiaشی KMeans محاسبه شده و در لیستی به نام " "wcssذخیره می شود.

۶. رسم مقادیر :WCSS

– مقادیر WCSS بر اساس تعداد خوشه ها ((kبا استفاده از تابع ()plotاز کتابخانه Matplotlib رسم می شوند.

– محور x نشان دهنده تعداد خوشه ها ((kو محور y نشان دهنده مقدار WCSS است.

– طرح با برچسب های محور مناسب و عنوان برچسب گذاری شده است.

۷. نمایش طرح

K-means

این کد پیاده سازی الگوریتم خوشه بندی K-means بر روی یک مجموعه داده است. اولین قدم فیلتر کردن ستون های مربوطه برای خوشه بندی است. در این مورد، ستون های ' 'h1تا ' 'h24از 'df' DataFrameانتخاب می شوند.

سپس داده ها با استفاده از « «()StandardScalerاستاندارد می شوند. سپس داده‌های استاندارد شده با ویژگی‌های اضافی مانند zone_id، سال، ماه و روز ترکیب می‌شوند.

پس از آن، خوشه بندی K-means با استفاده از تابع ()KMeansانجام می شود. در اینجا، تعداد خوشه ها روی ۳ و حالت تصادفی روی ۴۲ تنظیم شده است. تابع « «()fit_predictبرای به دست آوردن برچسب های خوشه استفاده می شود.

سپس برچسب‌های خوشه به عنوان یک ستون جدید به «df «DataFrame اصلی اضافه می‌شوند. در نهایت، خوشه ها با استفاده از « «matplotlibرسم می شوند. هر خوشه منحصر به فرد با رنگ متفاوتی رسم می شود و بر اساس آن برچسب گذاری می شود. مکان‌ها و برچسب‌های تیک محور x نیز برای نشان دادن شناسه‌های منطقه تنظیم شده‌اند.

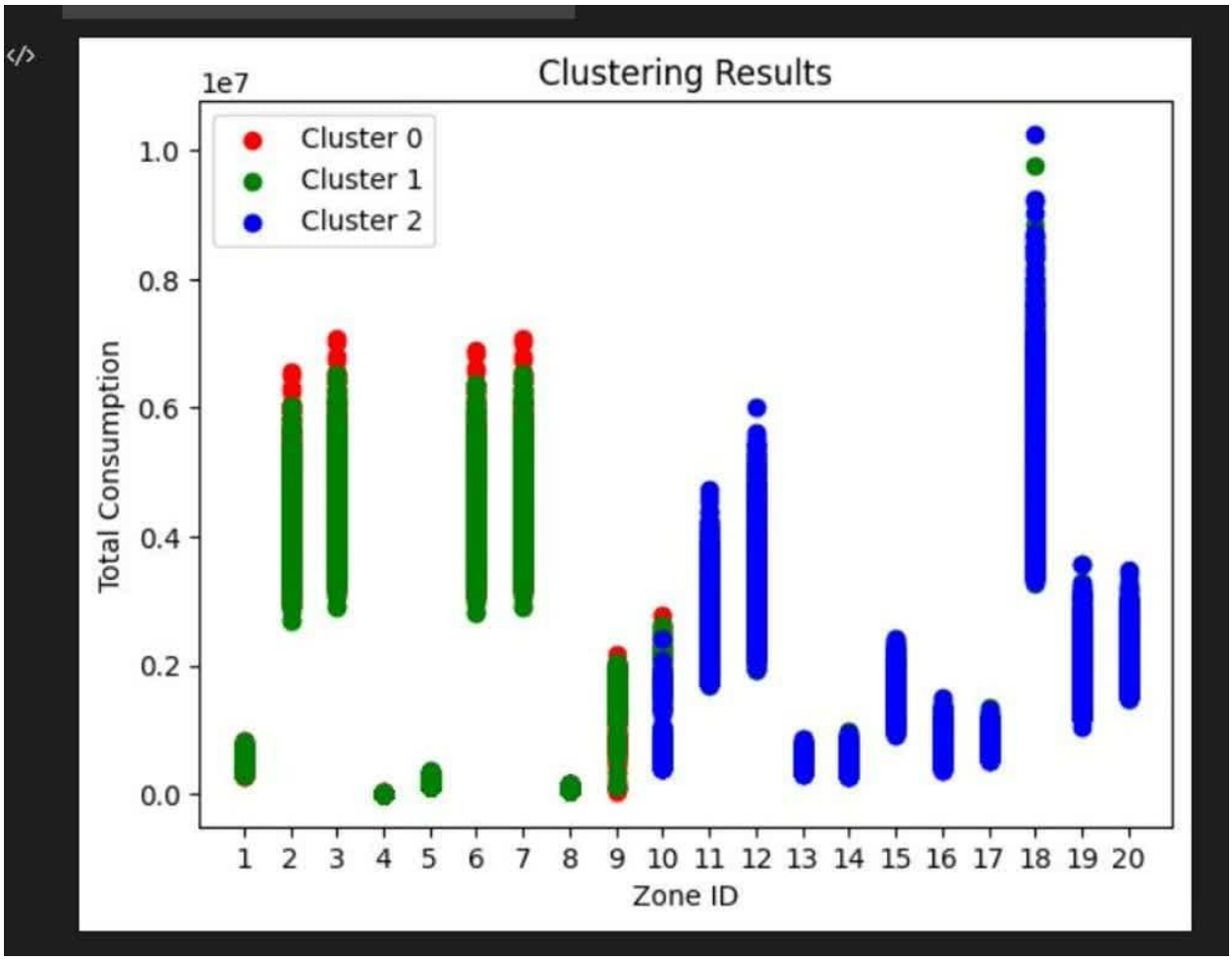
به طور کلی، این کد یک راه ساده و موثر برای انجام خوشه بندی K-means بر روی یک مجموعه داده و تجسم نتایج را ارائه می دهد.

این کد، خوشه‌بندی K-means را روی یک مجموعه داده برای گروه‌بندی نقاط داده مشابه با هم انجام می‌دهد. مجموعه داده ابتدا فیلتر می شود تا فقط ستون های مربوطه را برای خوشه بندی انتخاب کند، که ستون های ' 'h1تا ' 'h24هستند.

سپس، داده ها با استفاده از تابع « «()StandardScalerاستاندارد می شوند تا اطمینان حاصل شود که همه ویژگی ها مقیاس یکسانی دارند. سپس zone_id، سال، ماه و روز به عنوان ویژگی‌های اضافی با استفاده از تابع « «()np.concatenateبه داده‌های استاندارد الحاق می‌شوند.

سپس خوشه بندی K-means بر روی داده های استاندارد شده با استفاده از تابع " "()KMeansانجام می شود. در اینجا، تعداد خوشه ها روی ۳ و حالت تصادفی روی ۴۲ تنظیم شده است. تابع « «()fit_predictبرای به دست آوردن برچسب های خوشه استفاده می شود.

برچسب های خوشه به عنوان یک ستون جدید 'خوشه' به DataFrame اصلی اضافه می شوند. در نهایت، خوشه ها با استفاده از « «matplotlibرسم می شوند. هر خوشه منحصر به فرد با رنگ متفاوتی رسم می شود و بر اساس آن برچسب گذاری می شود. مکان‌ها و برچسب‌های تیک محور x نیز برای نشان دادن شناسه‌های منطقه تنظیم شده‌اند.



به طور کلی، این کد یک راه ساده و موثر برای انجام خوشه بندی K-means بر روی یک مجموعه داده و تجسم نتایج را ارائه می دهد.

DBSCAN

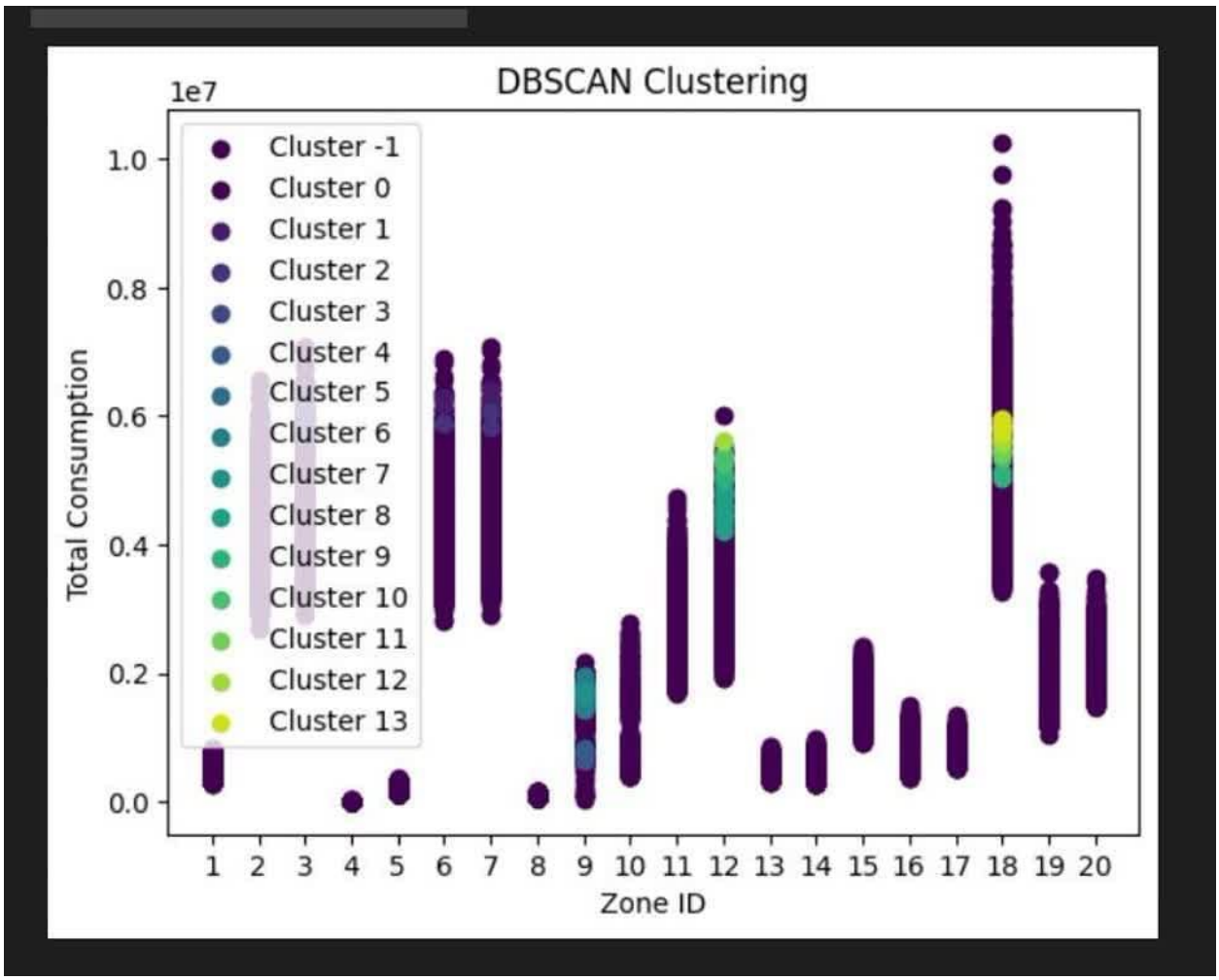
این کد، خوشه بندی DBSCAN را روی یک مجموعه داده انجام می دهد تا نقاط داده مشابه را با هم گروه بندی کند. ستون های مربوطه برای خوشه بندی ابتدا فیلتر می شوند که ستون های ' 'h1تا ' 'h24هستند.

سپس، داده ها با استفاده از تابع « «()StandardScalerاستاندارد می شوند تا اطمینان حاصل شود که همه ویژگی ها مقیاس یکسانی دارند. سپس خوشه‌بندی DBSCAN بر روی داده‌های استاندارد شده با استفاده از تابع « «()DBSCANانجام می‌شود. در اینجا، پارامتر

' 'epsروی ۰٫۵ و پارامتر ' 'min_samplesروی ۵ تنظیم شده است. تابع ' 'fit_predict()'برای به دست آوردن برچسب های خوشه استفاده می شود.

برچسب های خوشه به عنوان یک ستون جدید 'خوشه' به DataFrame اصلی اضافه می شوند. سپس، خوشه‌های منحصربه‌فرد با استفاده از « «np.unique()تنظیم می‌شوند. یک نقشه رنگی با استفاده از " "plt.cm.get_cmap()تعریف می شود.

در نهایت، یک نمودار پراکندگی برای هر خوشه با استفاده از ' 'plt.scatter()ایجاد می شود. هر خوشه منحصر به فرد با رنگ متفاوتی رسم می شود و بر اساس آن برچسب گذاری می شود. مکان‌ها و برچسب‌های تیک محور x نیز برای نشان دادن شناسه‌های منطقه تنظیم شده‌اند.



به طور کلی، این کد یک راه ساده و موثر برای انجام خوشه بندی DBSCAN بر روی یک مجموعه داده و تجسم نتایج را ارائه می دهد.

۴. استخراج الگوی مصرف هر خوشه

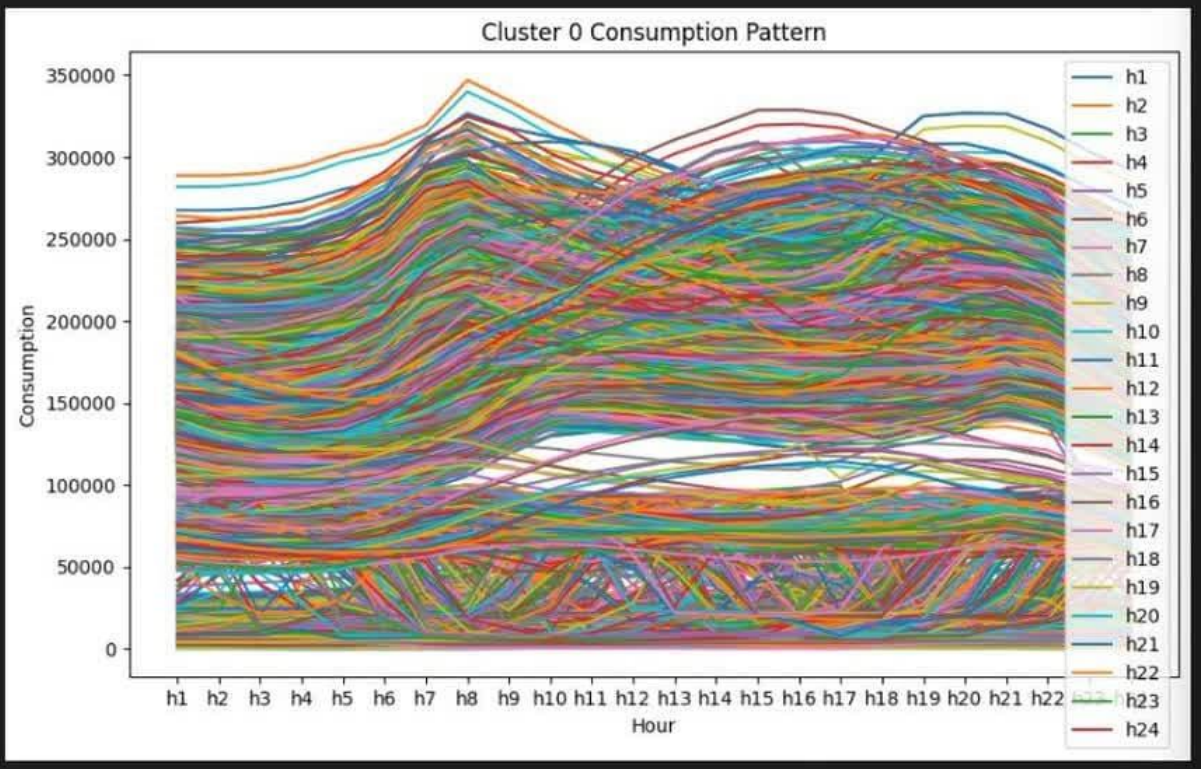
در این کد، تابع « «()groupbyبرای گروه بندی داده ها توسط برچسب های خوشه ای به دست آمده از خوشه بندی K-means استفاده می شود. این یک شی ' 'DataFrameGroupByایجاد می کند که حاوی DataFrame اصلی است که بر اساس برچسب های خوشه به گروه هایی تقسیم شده است. تابع « «()groupbyبه ما این امکان را می دهد که عملیات را روی هر گروه جداگانه انجام دهیم.

در مرحله بعد، یک حلقه « «forبرای تکرار بر روی هر گروه خوشه ای استفاده می شود. برای هر گروه، الگوی مصرف با استفاده از ستون های مربوطه DataFrame استخراج می شود. الگوی مصرف یک سری زمانی از مقادیر مصرف ساعتی برای هر منطقه در خوشه است.

پس از استخراج الگوی مصرف، می توان تحلیل بیشتری روی آن انجام داد. در این کد، مقادیر میانگین، حداکثر و حداقل مصرف برای هر ساعت در تمام مناطق در خوشه محاسبه می شود. این بینشی در مورد الگوهای مصرف معمول برای هر ساعت از روز در هر خوشه ارائه می دهد.

نتایج تجزیه و تحلیل برای هر خوشه با استفاده از " "()printچاپ می شود. نتایج مقادیر میانگین، حداکثر و حداقل مصرف را برای هر ساعت در تمام مناطق در هر خوشه نشان می‌دهد.

پس از انجام این تحلیل بر روی همه خوشه ها، یک خوشه خاص (خوشه ۰) برای تجزیه و تحلیل بیشتر انتخاب می شود. الگوی مصرف برای این خوشه از DataFrame استخراج شده و با استفاده از « «matplotlibرسم شده است. نمودار مقادیر مصرف ساعتی را برای هر منطقه در خوشه نشان می دهد.




```
Cluster 0 Average Consumption:
h1      81590.138278
h2      79351.280855
h3      78245.337959
h4      78239.257399
h5      79803.033526
h6      83895.582586
h7      90973.638462
h8      95963.489377
h9      97144.952352
```

در نهایت تحلیل آماری روی الگوی مصرف انجام می شود. مقادیر میانگین، حداکثر و حداقل مصرف برای کل خوشه در تمام ساعات محاسبه می شود. این بینشی در مورد الگوهای مصرف کلی برای کل خوشه ارائه می دهد.

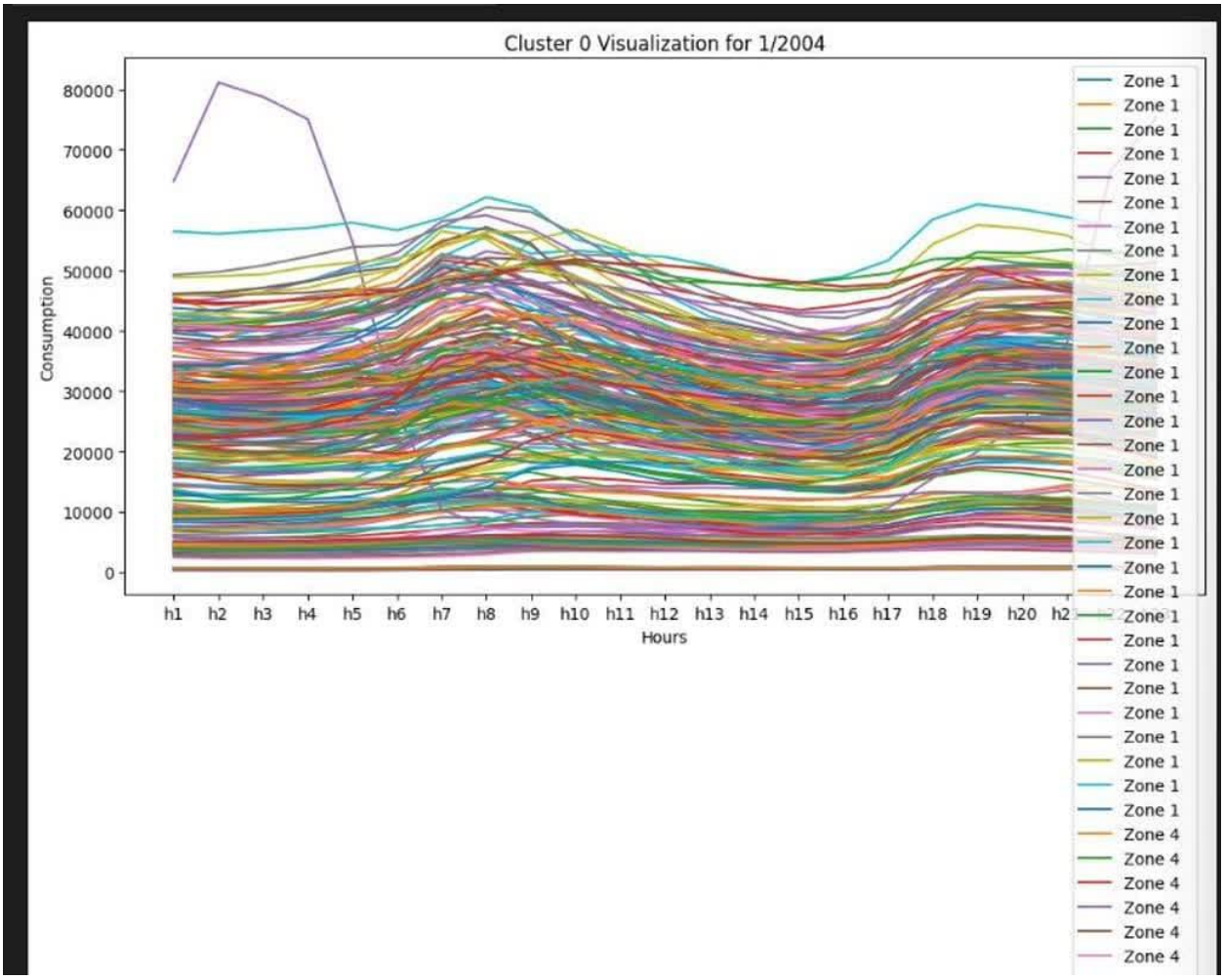
الگوی مصرف و خوشه بندی

خوشه بندی برای ۲۴ ساعت ماه ژانویه سال ۲۰۰۴

این کد داده های مصرف برق را از یک فایل CSV بارگیری می کند و خوشه بندی K-means را روی داده ها انجام می دهد. داده ها ابتدا برای یک سال و ماه خاص فیلتر می شوند. ستون های مربوطه برای خوشه بندی انتخاب می شوند و مقادیر از دست رفته حذف می شوند.

سپس خوشه‌بندی K-means بر روی داده‌های فیلتر شده با استفاده از تابع « «()KMeansاز scikit-learn انجام می‌شود. تعداد خوشه ها روی ۵ تنظیم شده است. تابع « «()fit_predictبرای به دست آوردن برچسب های خوشه برای هر نقطه داده استفاده می شود.

سپس، هر خوشه به طور جداگانه با استفاده از « «matplotlibترسیم می شود. برای هر خوشه، یک نمودار جداگانه ایجاد می شود که مقادیر مصرف ساعتی را برای هر منطقه در خوشه نشان می دهد. محور x ساعات روز و محور y مقادیر مصرف را نشان می دهد. هر ناحیه با رنگی متفاوت ترسیم شده و بر اساس آن برچسب گذاری می شود.



به طور کلی، این کد راهی برای تجسم نتایج خوشه‌بندی K-means بر روی داده‌های مصرف برق فراهم می‌کند. با ترسیم هر خوشه به طور جداگانه، می توانیم بینشی در مورد الگوهای مصرف مختلف در خوشه های مختلف به دست آوریم.

خوشه بندی برای ماه ژانویه سال ۲۰۰۴

این کد داده های مصرف برق را از یک فایل CSV بارگیری می کند و خوشه بندی K-means را روی داده ها انجام می دهد. داده ها ابتدا برای یک سال و ماه خاص فیلتر می شوند. ستون های مربوطه برای خوشه بندی انتخاب می شوند و مقادیر از دست رفته حذف می شوند.

سپس خوشه‌بندی K-means بر روی داده‌های فیلتر شده با استفاده از تابع « «()KMeansاز scikit-learn انجام می‌شود. تعداد خوشه ها روی ۵ تنظیم شده است. تابع « «()fit_predictبرای به دست آوردن برچسب های خوشه برای هر نقطه داده استفاده می شود.

سپس، هر خوشه به طور جداگانه با استفاده از « «matplotlibترسیم می شود. برای هر خوشه، یک نمودار جداگانه ایجاد می شود که مقادیر مصرف روزانه را برای هر منطقه در خوشه نشان می دهد. محور x روزهای ماه را نشان می دهد و محور y مقادیر مصرف را نشان می دهد. هر ناحیه با رنگی متفاوت ترسیم شده و بر اساس آن برچسب گذاری می شود.

به طور کلی، این کد راهی برای تجسم نتایج خوشه‌بندی K-means بر روی داده‌های مصرف برق برای یک ماه فراهم می‌کند. با ترسیم هر خوشه به طور جداگانه، می توانیم بینشی در مورد الگوهای مصرف مختلف در خوشه های مختلف به دست آوریم.
خوشه بندی برای سال ۲۰۰۴

این کد داده های مصرف برق را از یک فایل CSV بارگیری می کند و خوشه بندی K-means را روی داده ها انجام می دهد. ستون های مربوطه برای خوشه بندی انتخاب شده و مقادیر گم شده حذف می شوند. سپس داده ها برای یک سال خاص فیلتر می شوند.

خوشه بندی K-means بر روی داده های فیلتر شده با استفاده از تابع " "()KMeansاز scikit-learn انجام می شود. تعداد خوشه ها روی ۵ تنظیم شده است. تابع « «()fit_predictبرای به دست آوردن برچسب های خوشه برای هر نقطه داده استفاده می شود.

برچسب های خوشه با استفاده از روش " "()assignبه DataFrame اصلی اضافه می شوند. به هر نقطه داده یک برچسب خوشه ای بر اساس عضویت در یکی از خوشه ها اختصاص داده می شود.

سپس، هر خوشه به طور جداگانه با استفاده از « «matplotlibترسیم می شود. برای هر خوشه، یک نمودار جداگانه ایجاد می شود که مقادیر مصرف روزانه را برای هر منطقه در خوشه نشان می دهد. محور x روزهای ماه را نشان می دهد و محور y مقادیر مصرف را نشان می دهد. هر ناحیه با رنگی متفاوت ترسیم شده و بر اساس آن برچسب گذاری می شود.

به طور کلی، این کد راهی برای تجسم نتایج خوشه بندی K-means بر روی داده های مصرف برق برای یک سال خاص ارائه می دهد. با ترسیم هر خوشه به طور جداگانه، می توانیم بینشی در مورد الگوهای مصرف مختلف در خوشه های مختلف به دست آوریم.

خوشه
این کد داده های مصرف برق را از یک فایل CSV بارگیری می کند و خوشه بندی K-means را روی داده ها انجام می دهد. ستون های مربوطه برای خوشه بندی انتخاب می شوند و مقادیر از دست رفته حذف می شوند.

خوشه‌بندی K-means روی داده‌های فیلتر شده با استفاده از تابع «()KMeansاز scikit-learn انجام می‌شود. تعداد خوشه ها روی ۵ تنظیم شده است. تابع « «()fit_predictبرای به دست آوردن برچسب های خوشه برای هر نقطه داده استفاده می شود.

برچسب‌های خوشه با استفاده از روش « «()assignبه DataFrame اصلی اضافه می‌شوند. به هر نقطه داده یک برچسب خوشه ای بر اساس عضویت در یکی از خوشه ها اختصاص داده می شود.

سپس، هر خوشه به طور جداگانه با استفاده از « «matplotlibترسیم می شود. برای هر خوشه، یک نمودار جداگانه ایجاد می شود که مقادیر مصرف روزانه را برای هر منطقه در خوشه نشان می دهد. محور x روزهای ماه را نشان می دهد و محور y مقادیر مصرف را نشان می دهد. هر ناحیه با رنگی متفاوت ترسیم شده و بر اساس آن برچسب گذاری می شود.

به طور کلی، این کد راهی برای تجسم نتایج خوشه‌بندی K-means بر روی داده‌های مصرف برق برای همه مناطق فراهم می‌کند. با ترسیم هر خوشه به طور جداگانه، می توانیم بینشی در مورد الگوهای مصرف مختلف در خوشه های مختلف به دست آوریم.

الگوی مصرف بر اساس شناسه منطقه

این کد داده های مصرف برق را از یک فایل CSV بارگیری می کند و کل مصرف را برای هر ID منطقه و روز محاسبه می کند. ستون های مربوطه برای خوشه بندی انتخاب می شوند.

روش « «()groupbyبرای گروه بندی داده ها بر اساس شناسه منطقه و جمع مقادیر مصرف برای هر روز استفاده می شود.
DataFrame به دست آمده دارای شناسه منطقه به عنوان ردیف و روز به عنوان ستون است.

سپس DataFrame انتقال یافته با استفاده از « «matplotlibرسم می شود. برای هر شناسه منطقه، یک نمودار جداگانه ایجاد می شود که مقادیر کل مصرف را برای هر روز نشان می دهد. محور x روزهای ماه را نشان می دهد و محور y مقادیر مصرف را نشان می دهد. هر ناحیه با رنگی متفاوت ترسیم شده و بر اساس آن برچسب گذاری می شود.

به طور کلی، این کد راهی برای تجسم الگوی مصرف برای هر شناسه منطقه در ماه معین ارائه می دهد. با ترسیم داده ها برای هر منطقه به طور جداگانه، می توانیم بینشی در مورد الگوهای مصرف مختلف در مناطق مختلف بدست آوریم.

الگوی مصرف برای هر خوشه
این کد داده های مصرف برق را از یک فایل CSV بارگیری می کند و خوشه بندی K-means را روی داده ها انجام می دهد. ستون های مربوطه برای خوشه بندی انتخاب می شوند و مقادیر از دست رفته حذف می شوند.

خوشه‌بندی K-means روی داده‌های فیلتر شده با استفاده از تابع «()KMeansاز scikit-learn انجام می‌شود. تعداد خوشه ها روی ۵ تنظیم شده است. تابع « «()fit_predictبرای به دست آوردن برچسب های خوشه برای هر نقطه داده استفاده می شود.

برچسب‌های خوشه با استفاده از روش « «()assignبه DataFrame اصلی اضافه می‌شوند. به هر نقطه داده یک برچسب خوشه ای بر اساس عضویت در یکی از خوشه ها اختصاص داده می شود.

الگوی مصرف متوسط برای هر خوشه با استفاده از روش " "()meanاستخراج می شود. الگوهای به دست آمده در یک فرهنگ لغت با برچسب خوشه ای به عنوان کلید ذخیره می شوند.

سپس، الگوی مصرف هر خوشه به طور جداگانه با استفاده از « «matplotlibترسیم می‌شود. برای هر خوشه، نمودار جداگانه ای ایجاد می شود که مقادیر میانگین مصرف را برای هر روز نشان می دهد. محور x روزهای ماه را نشان می دهد و محور y مقادیر مصرف را نشان می دهد.

شناسه‌های منطقه‌ای که الگوی مصرف مشابهی با خوشه فعلی دارند برای هر خوشه چاپ می‌شوند.

به طور کلی، این کد راهی برای تجسم الگوی مصرف متوسط برای هر خوشه و شناسایی اینکه کدام منطقه شناسه دارای الگوهای مصرف مشابه است، ارائه می دهد. با تجزیه و تحلیل این الگوها، ما می توانیم بینشی در مورد رفتارهای مصرف مختلف در خوشه ها و مناطق مختلف به دست آوریم.

۵. سیستم اعلان(notification) هوشمند به کاربران

از طریق email :

این کد یک سیستم اطلاع رسانی هوشمند را برای ارسال ایمیل به کاربران بر اساس میزان مصرف آنها پیاده سازی می کند. مراحل زیر را انجام می دهد:

داده های مصرف و داده های ایمیل را از فایل های CSV بارگیری کنید.
ستون های مربوطه را برای خوشه بندی فیلتر کنید.
سطرهایی را با مقادیر از دست رفته رها کنید.
داده های سال و ماه مورد نظر را فیلتر کنید.
انجام خوشه بندی با استفاده از K-means برای شناسایی الگوهای مصرف.
داده های مصرف را با داده های ایمیل بر اساس zone_id ادغام کنید.

میانگین مصرف برای هر روز را محاسبه کنید.
تنظیمات ایمیل از جمله آدرس ایمیل و رمز عبور فرستنده را پیکربندی کنید.
تکرار بر روی هر کاربر و داده های مصرف آنها.
میزان مصرف (بالا، متوسط یا کم) را برای هر روز محاسبه کنید.
یک محتوای ایمیل با میزان مصرف برای هر روز تولید کنید.
بر اساس تعداد روزهای کم مصرف، متوسط یا زیاد، پیام‌های اضافی اضافه کنید.
یک پیام ایمیل با استفاده از MIMEText ایجاد کنید و فرستنده، گیرنده و موضوع را تنظیم کنید.
به سرور SMTP متصل شوید، فرستنده را احراز هویت کنید و ایمیل را ارسال کنید.
اگر ایمیل با موفقیت ارسال شد، پیام موفقیت آمیز را چاپ کنید.
در صورت بروز مشکل در ارسال ایمیل، موارد استثنا را مدیریت کنید و پیام خطا را چاپ کنید.
کد را می توان با تغییر مسیرهای فایل، تنظیمات ایمیل، آستانه مصرف و پیام های ارسال شده به کاربران بر اساس الگوی مصرف آنها سفارشی کرد.

در اینجا توضیحی در مورد نحوه عملکرد کد آورده شده است:
۱. **بارگیری داده ها:** کد با بارگیری داده های مصرف از یک فایل CSV در یک Pandas DataFrame ('df') شروع می شود. همچنین داده های ایمیل را از یک فایل CSV دیگر در یک DataFrame جداگانه (`email_df`) بارگیری می کند.

۲. **پیش پردازش داده ها:** ستون های مربوطه برای خوشه بندی فیلتر شده و در لیست "ستون_معتبر" ذخیره می شوند. ردیف‌هایی که مقادیر گمشده در داده‌های مصرفی دارند با استفاده از تابع «dropna()» حذف می‌شوند تا کیفیت داده‌ها تضمین شود.

۳. **فیلتر کردن داده ها:** کد داده های مصرف را برای یک سال و ماه خاص با مقایسه ستون های 'year' و 'month' در DataFrame ('df') با مقادیر مورد نظر ('year' و ') فیلتر می کند. متغیرهای ماه).

۴. **خوشه بندی:** داده های مصرف از DataFrame فیلتر شده ('df') استخراج شده و در آرایه 'X' ذخیره می شود. سپس کد با استفاده از کلاس «KMeans» از ماژول «sklearn.cluster» خوشه بندی K-means را انجام می دهد. تعداد خوشه‌ها روی «تعداد_خوشه‌ها» تنظیم شده است که به صورت ۵ تعریف می‌شود. برچسب‌های خوشه به هر نقطه داده اختصاص داده می‌شوند و در متغیر «برچسب_خوشه‌ای» ذخیره می‌شوند.

۵. **افزودن برچسب های خوشه:** برچسب های خوشه به عنوان یک ستون جدید ('cluster') به DataFrame ('df') اضافه می شوند.

۶. **ادغام داده ها:** داده مصرفی DataFrame ('df') با داده های ایمیل DataFrame ('email_df') بر اساس ستون 'zone_id' با استفاده از تابع 'merge()' ادغام می شود. DataFrame ادغام شده (`merged_df`) حاوی داده های مصرف و آدرس های ایمیل مربوطه است.

۷. **محاسبه میانگین مصرف:** میانگین مصرف برای هر روز با گرفتن میانگین ستون های داده مصرف ("valid_columns") با استفاده از تابع "mean()" محاسبه می شود. مقادیر متوسط مصرف در متغیر «متوسط_مصرف» ذخیره می شود.

۸. **پیکربندی ایمیل:** کد آدرس فرستنده ایمیل ("email_sender")، رمز عبور ایمیل ("email_password") و آستانه مصرف ("consumption_threshold") را بر اساس مقادیر مورد نظر تنظیم می کند.

۹. **ارسال ایمیل:** کد روی هر ردیف در DataFrame ادغام شده (`merged_df`) با استفاده از تابع `iterrows`()` تکرار می شود. برای هر کاربر، آدرس ایمیل، شناسه منطقه و داده های مصرف را بازیابی می کند.

۱۰. **محاسبه سطح مصرف:** این کد سطح مصرف (بالا، متوسط یا کم) را برای هر روز از ماه با مقایسه مقادیر واقعی مصرف با مقادیر آستانه ("آستانه_مصرف * متوسط_مصرف" و "(۱) محاسبه می کند. / آستانه_مصرف) *متوسط_مصرف`). همچنین تعداد روزهای با مصرف کم، متوسط و زیاد را پیگیری می کند.

۱۱. **تولید محتوای ایمیل:** کد یک پیام ایمیل ایجاد می کند که شامل شناسه منطقه کاربر و سطح مصرف برای هر روز از ماه است. روی سطوح مصرف تکرار می شود و روز و سطح مربوطه را به پیام اضافه می کند.

۱۲. **پیام های اضافی:** بر اساس تعداد روزهای مصرف کم، متوسط و زیاد، کد پیام های اضافی را به محتوای ایمیل اضافه می کند. اگر تعداد روزهای کم مصرف بیش از ۱۵ روز باشد، به کاربر پیشنهاد می‌کند تا میزان مصرف خود را بررسی کند. اگر تعداد روزهای مصرف متوسط بیش از ۱۵ روز باشد، بهینه سازی استفاده برای کارایی را پیشنهاد می کند. اگر تعداد روزهای پرمصرف بیش از ۱۵ روز باشد، نظارت بر استفاده را برای صرفه جویی احتمالی توصیه می کند.

۱۳. **ارسال ایمیل:** کد با استفاده از کلاس 'MIMEText' از ماژول 'email.mime.text' یک پیام ایمیل ایجاد می کند. فرستنده، گیرنده، موضوع و محتوای ایمیل را تعیین می کند. سپس، با استفاده از کلاس «smtplib.SMTP» به سرور SMTP متصل می شود، یک اتصال امن TLS را راه اندازی می کند، به حساب ایمیل فرستنده وارد می شود و ایمیل را با استفاده از روش «send_message()» ارسال می کند.

۱۴. **چاپ و رسیدگی به خطا:** اگر ایمیل با موفقیت ارسال شود، کد یک پیام موفقیت آمیز چاپ می کند. اگر در طول فرآیند ارسال ایمیل استثنا وجود داشته باشد، یک پیام خطا همراه با جزئیات استثنا چاپ می کند.

کد را می توان با تغییر مسیرهای فایل، تنظیمات ایمیل، آستانه مصرف و پیام های ارسال شده به کاربران بر اساس الگوی مصرف آنها سفارشی کرد.

از طریق sms :

به طور خلاصه، Twilio در کد استفاده می‌شود تا از قابلیت‌های پیامک خود استفاده کند، و به سیستم اجازه می‌دهد تا هشدارهای پیامکی را به طور کارآمد و قابل اعتماد برای کاربران در زمانی که معیارهای مصرف 25 روز متوالی یا بیشتر مصرف بالا را برآورده می‌کنند، ارسال کند

توضیحی در مورد نحوه عملکرد کد آورده شده است:

۱. کد با وارد کردن ماژول‌ها و کتابخانه‌های لازم، مانند «زمان» برای تأخیر و «twilio.rest» برای ارسال پیام‌های SMS با استفاده از Twilio شروع می‌شود.

۲. اعتبارنامه های مورد نیاز حساب Twilio، از جمله SID حساب، رمز تأیید، و شماره تلفن Twilio را تنظیم می کند.

۳. سپس کد یک حلقه را بر روی شناسه‌های منطقه منحصربه‌فرد موجود در «merged_df» DataFrame آغاز می‌کند.

۴. برای هر Zone ID، داده‌ها را از «merged_df» فیلتر می‌کند تا رکوردهای مربوط به آن Zone ID را به دست آورد و آنها را در «zone_data» DataFrame ذخیره می‌کند.

۵. در داخل حلقه، کد یک متغیر شمارنده، 'continuous_high_consumption_days' را مقداردهی اولیه می کند تا تعداد روزهای متوالی با مصرف بالا را پیگیری کند.

۶. سپس روی ردیف‌های «zone_data» حلقه می‌زند تا داده‌های هر کاربر را جداگانه پردازش کند.

۷. در داخل حلقه، کد آدرس ایمیل، اطلاعات مصرف و شماره تلفن کاربر را بازیابی می کند.

۸. سطح مصرف را برای هر روز بر اساس داده های مصرف محاسبه می کند و تعداد روزهای مصرف کم، متوسط و زیاد را پیگیری می کند.

۹. در صورت وجود روزهای مصرف بالا، کد شمارنده «روزهای_مصرف_بالا_مستمر» را افزایش می دهد.

۱۰. اگر شمارنده «continuous_high_consumption_days» به 25 یا بیشتر برسد، کد با استفاده از API Twilio یک هشدار پیامکی برای کاربر ارسال می کند.

۱۱. با استفاده از SID حساب Twilio و توکن تأیید، یک شی «مشتری» ایجاد می کند.

۱۲. کد پیام کوتاه شامل شناسه منطقه و یک پیام از پیش تعریف شده که مصرف بالا را نشان می دهد، می سازد.

۱۳. پیام SMS را با استفاده از روش «create()» از ویژگی «پیام‌ها» شی «مشتری» ارسال می‌کند و شماره تلفن Twilio و شماره تلفن کاربر را مشخص می‌کند.

۱۴. کد با استفاده از "time.sleep()" تاخیر ۳ ثانیه ای را برای کنترل سرعت ارسال پیامک معرفی می کند.

۱۵. پس از ارسال پیامک به کاربر، از حلقه خارج می شود، زیرا برای هر شناسه منطقه ای که معیارهای مصرف بالا را دارد، فقط یک پیامک ارسال می شود.

به طور خلاصه، کد بر روی داده‌ها تکرار می‌شود، کاربرانی را که مصرف بالایی دارند برای شناسه هر منطقه شناسایی می‌کند و هشدارهای پیامکی را با تاخیر بین هر پیام به آن کاربران ارسال می‌کند تا از نرخ کنترل شده تحویل اطمینان حاصل شود.

Twilio یک پلت فرم ارتباطات ابری است که APIها و خدمات مختلفی را برای ارسال و دریافت پیام‌های متنی (SMS)، برقراری تماس‌های تلفنی و انجام سایر وظایف مرتبط با ارتباط ارائه می‌کند. در کد، Twilio برای ارسال پیامک هشدار به کاربرانی که 25 روز یا بیشتر متوالی پرمصرف را تجربه کرده اند استفاده می شود.

در اینجا توضیح مختصری درباره دلیل استفاده از Twilio در کد آورده شده است:

۱. یکپارچه سازی API Twilio: کد با REST API Twilio برای ارسال پیامک ادغام می شود. از ماژول "twilio.rest" برای ایجاد یک شی "Client" با SID حساب Twilio و نشانه تایید استفاده می کند که تعامل با سرویس های Twilio را امکان پذیر می کند.

۲. ارسال پیام‌های پیام کوتاه: هنگامی که شی «مشتری» ایجاد شد، کد از روش «create()» ویژگی «پیام‌ها» برای ارسال پیام کوتاه استفاده می‌کند. شماره تلفن فرستنده (شماره تلفن Twilio) و شماره تلفن گیرنده (شماره تلفن کاربر) را برای تحویل پیام مشخص می کند.

۳. سهولت ارتباط: Twilio فرآیند ارسال هشدارهای پیامکی به کاربران را ساده می کند. با استفاده از API Twilio، توسعه‌دهندگان می‌توانند بدون نیاز به مدیریت پیچیدگی‌های اتصالات و پروتکل‌های اپراتور، عملکرد SMS را در برنامه‌های خود ادغام کنند.

۴. مقیاس پذیری و قابلیت اطمینان: Twilio یک پلت فرم مقیاس پذیر و قابل اعتماد برای ارسال پیام های SMS ارائه می دهد. این زیرساخت زیرساخت و اتصالات شبکه را کنترل می کند و اطمینان می دهد که پیام ها به سرعت و قابل اعتماد به گیرندگان تحویل داده می شوند.

۶. رتبه بندی کاربران براساس مصرف کل، مصرف درپیک، کاهش مصرف و .

رتبه بندی بر اساس مصرف کل. برای اینکار مصرف کل کاربر را جمع میزنیم سپس کاربران را بترتیب از بیشترین به کمترین مصرف رتبه بندی میکنیم.

```
zone_id
18    8129663659
3     7136036706
7     7136036706
6     6909147471
2     6613552350
12    5060428249
11    4104436948
20    3374361428
19    2979479364
9     2573968584
15    2369808762
17    1251232837
10    1236774440
16    1112488750
14     794009894
13     748541231
1      709560216
5      295624223
8      143645257
4       18986468
Name: total_consumption, dtype: int64
```

رتبه بندی بر اساس مصرف در پیک. برای اینکار ابتدا ساعات پیک مصرف را پیدا کردیم.
که نتیجه ساعت ۱۹ تا ۲۱ شد.

سپس بر اساس مصرف در این ساعات کاربران را رتبه بندی کردیم.

```
zone_id
18    1196665142
3      993494239
7      993494239
6      967492283
2      920752838
12     771064057
11     600103618
20     477178639
19     440609723
15     330994701
9      313378128
17     176741047
10     175903159
16     162948751
14     119983838
13     106318180
1      105314233
5       46739501
8       21317418
4        2807960
Name: pick_hour_consumption, dtype: int64
```

در این رتبه بندی هم مانند رتبه بندی قبلی کاربر ۱۸ رتبه اول شد.

اما مشاهده میکنیم که رتبه کاربران ۹ و ۱۵ در این دو رتبه بندی متفاوت هستند.

رتبه بندی بر اساس کاهش مصرف. برای این کار ابتدا مصرف کل هر کاربر را در سال ۲۰۰۴ و همچنین در سال ۲۰۰۸ محاسبه کردیم. با توجه به میزان کاهش در مصرف کاربران را رتبه بندی کردیم.

zone_id	Percentage Decrease	init Consum	Final Consum	Decrease
5	-50.315032	65643549	32614976	-33028573
15	-49.771140	532510964	267474188	-265036776
9	-49.348764	587631650	297642692	-289988958
14	-47.864959	174558120	91005948	-83552172
16	-47.643106	243523715	127501453	-116022262
13	-47.640781	166089035	86962922	-79126113
1	-46.984773	156305631	82865785	-73439846
6	-46.835800	1520997068	808625923	-712371145
4	-46.776375	4176897	2223096	-1953801
3	-46.678475	1570335514	837326846	-733008668
7	-46.678475	1570335514	837326846	-733008668
2	-46.678473	1455359080	776019690	-679339390
19	-45.956791	648949744	350713265	-298236479
17	-45.760879	272697296	147908617	-124788679
11	-45.561512	888648953	483767052	-404881901
12	-45.322889	1085583013	593565432	-492017581
20	-44.934546	737477454	406095309	-331382145
18	-44.647475	1749231222	968243657	-780987565
8	-43.114879	31071566	17675098	-13396468
10	37.549861	237425845	326578920	89153075

مقالات مربوطه

۱. «تحلیل سری های زمانی: پیش بینی و کنترل» نوشته جورج ای پی باکس، گویلیم ام. جنکینز، گرگوری سی راینسل و گرتا ام. لیونگ

این کتاب درک عمیقی از تکنیک های تحلیل سری های زمانی و پیش بینی ارائه می دهد. این مفاهیم مدل‌های ARIMA، هموارسازی نمایی، مدل‌های فضای حالت و موارد دیگر را پوشش می‌دهد. این کتاب همچنین شامل مثال‌های عملی و مطالعات موردی است تا به خوانندگان کمک کند تا این مفاهیم را در مسائل دنیای واقعی به کار ببرند.

۲. «پیش بینی بار برق: مروری» نوشته علی محمدی و علی عباسپور-تهرانی فرد.

این نامه مروری بر تکنیک های پیش بینی بار الکتریکی، از جمله روش های پیش بینی کوتاه مدت و بلند مدت ارائه می دهد. همچنین چالش های مرتبط با پیش بینی بار و اهمیت پیش بینی های دقیق برای مدیریت انرژی را مورد بحث قرار می دهد.

۳. "خوشه بندی داده ها: یک بررسی" توسط .A.Kجین، م.ن. مورتی و پی جی فلین

این مقاله مروری بر تکنیک‌های خوشه‌بندی، از جمله خوشه‌بندی سلسله مراتبی، خوشه‌بندی k-means و خوشه‌بندی مبتنی بر چگالی ارائه می‌دهد. همچنین کاربردهای خوشه بندی در زمینه های مختلف مانند پردازش تصویر، بیوانفورماتیک و داده کاوی را مورد بحث قرار می دهد.

۴. «الگوهای متداول استخراج، ارتباط و همبستگی ها: از جریان داده تا برنامه های کاربردی در دنیای واقعی» نوشته چارو سی. آگاروال

این کتاب مفاهیم الگوکاوی، از جمله الگوبرداری مکرر و قانون کاوی را پوشش می دهد. همچنین کاربردهای الگو کاوی در حوزه های مختلف مانند خرده فروشی، مراقبت های بهداشتی و مالی را مورد بحث قرار می دهد.

۵. «سیستم‌های اعلان هوشمند: یک مرور کلی» توسط Alok K. Shukla، Saurabh Pal، و Ramesh C. Joshi

این مقاله مروری بر سیستم‌های اعلان هوشمند، از جمله معماری، اجزا و برنامه‌های کاربردی آن‌ها ارائه می‌کند. همچنین چالش های مرتبط با طراحی سیستم های اطلاع رسانی موثر و اهمیت شخصی سازی در اعلان ها را مورد بحث قرار می دهد.

۶. «تحلیل مصرف برق: مروری» نوشته محمد آصف حنیف و محمد علی عمران.

این مقاله مروری بر تکنیک‌های آنالیز مصرف برق، از جمله پروفایل بار، تحلیل اوج تقاضا، و تفکیک انرژی ارائه می‌کند. همچنین کاربردهای تجزیه و تحلیل مصرف در حوزه های مختلف مانند خانه های هوشمند، شبکه های هوشمند و سیستم های مدیریت انرژی را مورد بحث قرار می دهد.