به نام خداوند

گزارش پروژهٔ پایان ترم

اقتصادسنجی پیشرفته

علیرضا جمالی

[alireza.jamalie@ut.ac.ir](mailto:alireza.jamalie@ut.ac.ir)

\* گزارش توضیح کدهای هر قسمت از پروژه، در پوشهٔ مربوط به همان قسمت وجود دارد.

\* تمامی کدهای تمرین‌ها و پروژهٔ درس اقتصادسنجی به همراه تاریخچه آن‌ها در [این مخزن گیت](https://github.com/alrzJml/advanced_econ) قابل مشاهده است.

دانشکده مدیریت دانشگاه تهران

زمستان ۱۴۰۱

فهرست

[پروژه ۱ – انتقال کد معاملات جفتی از متلب به پایتون 3](#_Toc124725900)

[راهنمای فایل‌ها 3](#_Toc124725901)

[دادهٔ ورودی 3](#_Toc124725902)

[روش اجرا 3](#_Toc124725903)

[خروجی 4](#_Toc124725904)

[پروژه ۲ – مقایسه VECM با شبکه‌های عصبی 6](#_Toc124725905)

[راهنمای فایل‌ها 6](#_Toc124725906)

[دادهٔ ورودی 6](#_Toc124725907)

[روش اجرا 6](#_Toc124725908)

[محدودیت‌ها 7](#_Toc124725909)

[خروجی 7](#_Toc124725910)

[پروژه ۳ – آزمون خطی بودن ارزهای دیجیتال 9](#_Toc124725911)

[راهنمای فایل‌ها 9](#_Toc124725912)

[ورودی‌ها 9](#_Toc124725913)

[روش اجرا 9](#_Toc124725914)

[خروجی‌ها 9](#_Toc124725915)

[پروژه ۴ – یافتن توزیع Kernel Density مدل ECM 11](#_Toc124725916)

[راهنمای فایل‌ها 11](#_Toc124725917)

[ورودی‌ها 11](#_Toc124725918)

[روش اجرا 11](#_Toc124725919)

[خروجی‌ها 12](#_Toc124725920)

# پروژه ۱ – انتقال کد معاملات جفتی از متلب به پایتون

در این پروژه، کد mohPairesStocks.m به زبان پایتون نوشته می‌شود.

## راهنمای فایل‌ها

|  |  |
| --- | --- |
| توضیح | آدرس فایل |
| گزارش توضیح کدها به زبان انگلیسی | ./01\_pair\_trading/02\_paired\_all.pdf |
| فایل ژوپیتر نوت‌بوک نهایی همراه با توضیحات | ./01\_pair\_trading/02\_paired\_all.ipynb |
| فایل پایتون نهایی همراه با توضیحات | ./01\_pair\_trading/02\_paired\_all.py |
| پوشهٔ حاوی نمودارهای سهام هم‌انباشته | ./01\_pair\_trading/plots/{inex\_name} |
| فایل اکسل لیست سهام هم‌انباشته | ./01\_pair\_trading/pairs\_2023-01-15.xlsx |

## دادهٔ ورودی

1. ۳۰ سهم موجود در شاخص Dow Jones (منبع: یاهوفایننس)
2. ۴۰ سهم موجود در شاخص Dax (منبع: یاهوفایننس)
3. ۴۰ سهم موجود در شاخص CAC 40 (منبع: یاهوفایننس)
4. ۵۰ سهم موجود در شاخص پنجاه شرکت بزرگ بورس تهران (منبع: TSE)

* گام سری‌های قیمتی روزانه است.
* بازهٔ زمانی برای سری‌های قیمتی، ۵۲۱ روز است که ۱۲۶ روز آخر به عنوان بازهٔ آزمون انتخاب شده است.
* از لگاریتم قیمت‌ها برای محاسبات مربوط به تست‌های هم‌انباشتگی و رسم نمودارها استفاده می‌شود.

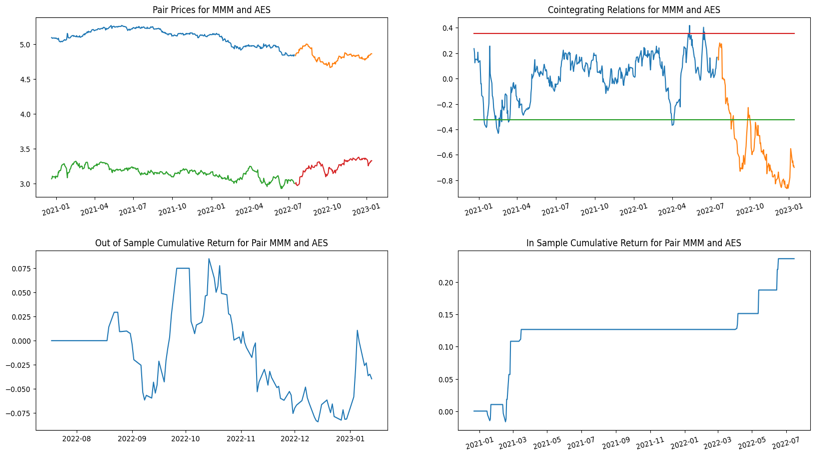
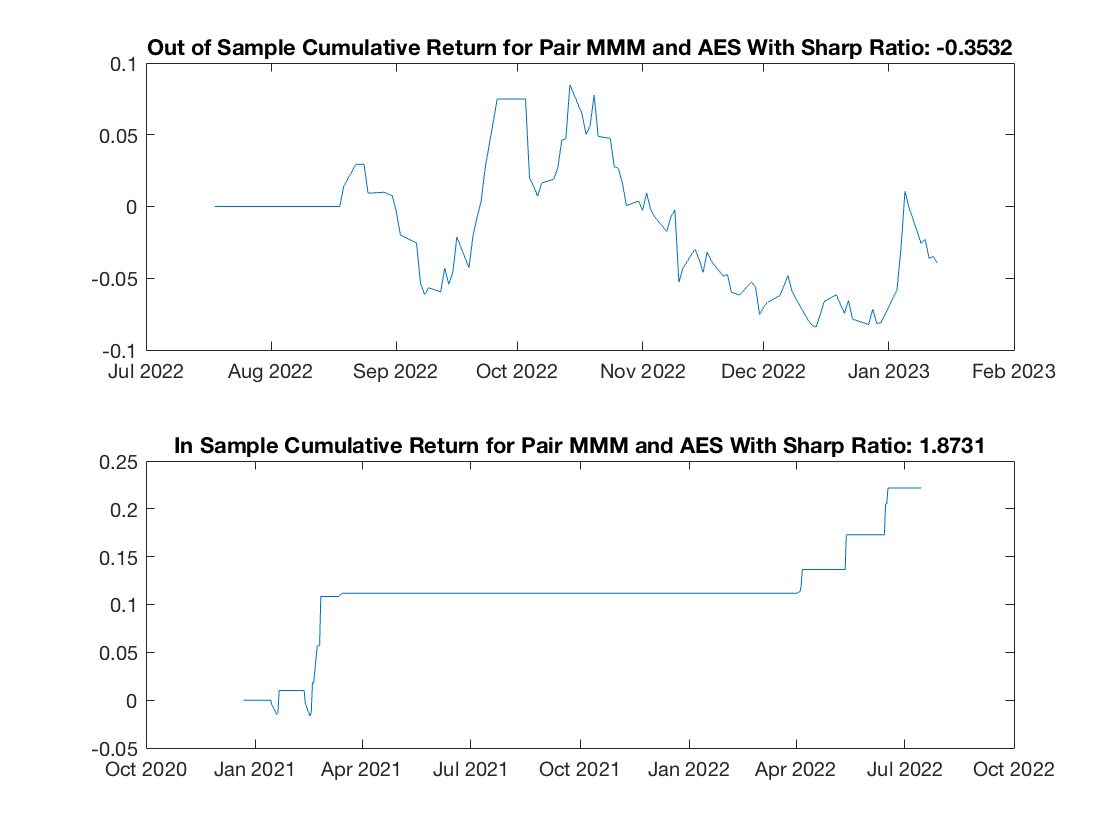
## روش اجرا

به ازای هر جفت سهم موجود در هر شاخص، آزمون هم‌انباشتگی یوهانسون اجرا می‌شود. در صورتی که دو سهم هم‌انباشته باشند، الگوریتم معاملات جفتی اجرا می‌شود و پس از محاسبه بازده این الگوریتم، نمودار مربوطه رسم و در نهایت ذخیره می‌شود.

* تعداد تست‌های هم‌انباشتگی انجام شده (مجموع تمام ترکیب‌های دوتایی سهام موجود در هر شاخص):
* برخی از توابع و کلاس‌های استفاده شدهٔ مهم:

|  |  |
| --- | --- |
| توضیح | تابع یا کلاس |
| تابعی است که برای دریافت لیست سهام موجود در هر شاخص نوشته‌ام. برای سه شاخص خارجی، از پکیج PyTickerSymbols استفاده می‌شود و برای لیست سهام ایرانی، از وبسایت فیپیران مستقیما خوانده می‌شود. | ./01\_pair\_trading/helper\_functions.stock\_list |
| تابعی است که با دریافت لیستی از سهام، دیتافریمی از قیمت تعدیل‌شدهٔ آن‌ها را به عنوان خروجی می‌دهد. | ./01\_pair\_trading/helper\_functions.stock\_prices |
| این تابع، تعداد وقفهٔ مناسب را برای مدل vecm پیشنهاد می‌دهد. | statsmodels.tsa.vector\_ar.vecm.select\_order |
| این تابع، مرتبهٔ هم‌انباشتگی را برای دو سری قیمتی محاسبه می‌کند. در درون این تابع، تست یوهانسون انجام می‌گیرد. از جمله موارد خروجی، آماره و مقادیر بحرانی تست یوهانسون است. | statsmodels.tsa.vector\_ar.vecm.select\_coint\_rank |
| این کلاس با دریافت دو سری زمانی هم‌انباشته، یک آبجکت VECM به عنوان خروجی می‌دهد. با فراخوانی متد fit می‌توان پارامترهای مدل VECM را محاسبه کرد. | statsmodels.tsa.vector\_ar.vecm.VECM |

## خروجی

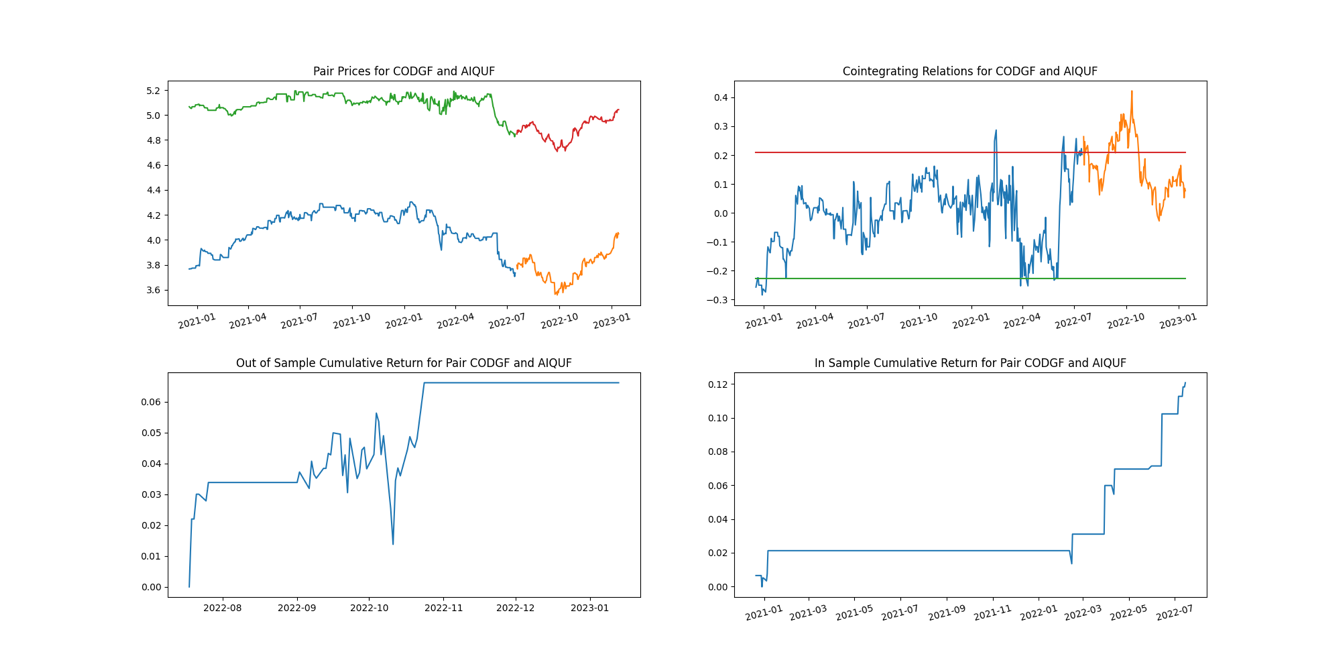
خروجی کد پایتون بسیار به کد متلب شبیه است. برای مثال، در زیر نمودارهای نهایی دو سهم هم‌انباشته در خروجی متلب (چپ) و خروجی پایتون (راست) آورده شده است.

شکل 1 بازدهٔ الگوریتم معاملات جفتی محاسبه‌شده توسط کد پایتون

شکل 2 بازدهٔ الگوریتم معاملات جفتی محاسبه‌شده توسط کد متلب

* تعداد جفت سهام هم‌انباشته:

|  |  |
| --- | --- |
| تعداد | شاخص |
| ۶۶ | Dow Jones |
| ۱۴۵ | CAC 40 |
| ۱۳۴ | Dax |
| ۱۶۴ | Teh50 |
| **۵۰۹** | **جمع** |

* ۱۶درصد تست‌های هم‌انباشتگی این پروژه مثبت شدند. در تمرین آخر کلاسی، تست هم‌انباشتگی بر روی ۱۰۰۰ سری گام تصادفی با ۱۰۰ مشاهده انجام شد که فقط ۵ درصد این سری‌ها با هم‌انباشته بودند.
* لیست سهام هم‌انباشتهٔ هر شاخص در فایل اکسل ./01\_pair\_trading/pairs\_2023-01-15.xlsx در شیت‌های مجزا موجود است.
* **به ازای هر جفت سهام هم‌انباشته، یک تصویر حاوی چهار نمودار در پوشه ./01\_pair\_trading/plots/{index} ذخیره شده است.**
* از آن‌جایی‌که تعداد تصاویر و نمودارها بسیار زیاد است (۵۰۹ تا)، در زیر، دو تصویر از میان آن‌ها برای نمونه انتخاب شده است.
* نمودار CODGF و AIQUF در شاخص CAC 40:
* نمودار فخاس و فملی در شاخص Teh50:

# پروژه ۲ – مقایسه VECM با شبکه‌های عصبی

در این پروژه، خطای پیش‌بینی مدل VECM با مدل MPLRegression مقایسه می‌شود.

## راهنمای فایل‌ها

|  |  |
| --- | --- |
| توضیح | آدرس فایل |
| گزارش توضیح کد پیش‌بینی مدل VECM به زبان انگلیسی | ./02\_vecm\_vs\_neurals/01\_vecm.pdf |
| گزارش توضیح کد پیش‌بینی مدل شبکه‌های عصبی به زبان انگلیسی | ./02\_vecm\_vs\_neurals/02\_nn\_multioutput.pdf |
| فایل ژوپیتر نوت‌بوک کد پیش‌بینی مدل VECM همراه با توضیحات | ./02\_vecm\_vs\_neurals/01\_vecm.ipynb |
| فایل ژوپیتر نوت‌بوک کد پیش‌بینی مدل شبکه‌های عصبی همراه با توضیحات | ./02\_vecm\_vs\_neurals/02\_nn\_multioutput.ipynb |
| فایل پایتون کد پیش‌بینی مدل VECM همراه با توضیحات | ./02\_vecm\_vs\_neurals/01\_vecm.py |
| فایل پایتون کد پیش‌بینی مدل شبکه‌های عصبی همراه با توضیحات | ./02\_vecm\_vs\_neurals/02\_nn\_multioutput.py |
| فایل اکسل خطاهای پیش‌بینی مدل VECM | ./02\_vecm\_vs\_neurals/vecm\_mape.xlsx |
| فایل اکسل خطاهای پیش‌بینی مدل شبکه‌های عصبی | ./02\_vecm\_vs\_neurals/nn\_mape.xlsx |
| پوشهٔ حاوی نمودارهای پیش‌بینی مدل VECM | ./02\_vecm\_vs\_neurals/preds\_vecm/{index\_name} |
| پوشهٔ حاوی نمودارهای پیش‌بینی مدل شبکه‌های عصبی | ./02\_vecm\_vs\_neurals/preds\_nn/{index\_name} |

## دادهٔ ورودی

* ۵۰۹ سهمی که در تمرین قبل هم‌انباشته تشخیص داده شدند.
* گام سری‌های قیمتی روزانه است.
* بازه زمانی برای سری‌های قیمتی، ۵۲۱ روز است که ۱۲۶ روز آخر به عنوان بازه آزمون انتخاب شده است.
* از لگاریتم قیمت‌ها برای محاسبات مربوط به تست‌های هم‌انباشتگی و رسم نمودارها استفاده می‌شود.

## روش اجرا

ابتدا پیش‌بینی تک‌گام و چندگام مدل VECM برای همهٔ جفت سهام هم‌انباشته صورت می‌گیرد (۵۰۹ جفت سهم) و خطای پیش‌بینی در فایل اکسل ذخیره می‌شود. سپس همین‌کار برای پیش‌بینی‌های مدل MLPRegressor نیز انجام می‌شود. در نهایت، با رسم هیستوگرام خطاهای هر دو مدل، به مشاهده تفاوت خطاها می‌پردازیم.

* برای پیش‌بینی‌ها از مدل‌های multioutput استفاده می‌شود.
* پیش‌بینی‌های چندگام، همگی دینامیک هستند.
* از شاخص MAPE برای مقایسه کارایی این دو مدل استفاده شده است. اصلی‌ترین دلیل این انتخاب، مستقل بودن از مقیاس این شاخص است. زیرا می‌خواهیم از طریق رسم هیستوگرام، خطای ۵۰۹ جفت سری زمانی را که مقادیر مختلفی دارند، با هم مقایسه کنیم و برای این کار، معیارمان باید مستقل از مقیاس باشد.
* ۵۰۹ جفت سهم هم‌انباشته از تمرین قبل داریم. به ازای هر جفت، چهار پیش‌بینی صورت می‌گیرد: تک‌گام سهم اول، چندگام سهم اول، تک‌گام سهم دوم، چندگام سهم دوم. بنابراین تعداد مقادیر خطای محاسبه شده باید 509\*4 = 2036 باشد.
* برخی از توابع و کلاس‌های استفاده شدهٔ مهم:

|  |  |
| --- | --- |
| توضیح | تابع یا کلاس |
| تابعی است که برای دریافت لیست سهام موجود در هر شاخص نوشته‌ام. برای سه شاخص خارجی، از پکیج PyTickerSymbols استفاده می‌شود و برای لیست سهام ایرانی، از وبسایت فیپیران مستقیما خوانده می‌شود. | ./02\_vecm\_vs\_neurals/helper\_functions.stock\_list |
| تابعی است که با دریافت لیستی از سهام، دیتافریمی از قیمت تعدیل‌شدهٔ آن‌ها را به عنوان خروجی می‌دهد. | ./02\_vecm\_vs\_neurals/helper\_functions.stock\_prices |
| این کلاس با دریافت دو سری زمانی هم‌انباشته، یک آبجکت VECM به عنوان خروجی می‌دهد. با فراخوانی متد fit می‌توان پارامترهای مدل VECM را محاسبه کرد. | statsmodels.tsa.vector\_ar.vecm.VECM |
| این تابع، مدل‌های پکیج sklearn را تبدیل به مدل‌های چندخروجی می‌کند. | sklearn.multioutput.MultiOutputRegressor |
| این تابع، از مدل MLP برای رگرسیون سری‌ زمانی استفاده می‌کند. | sklearn.neural\_network.MLPRegressor |

## محدودیت‌ها

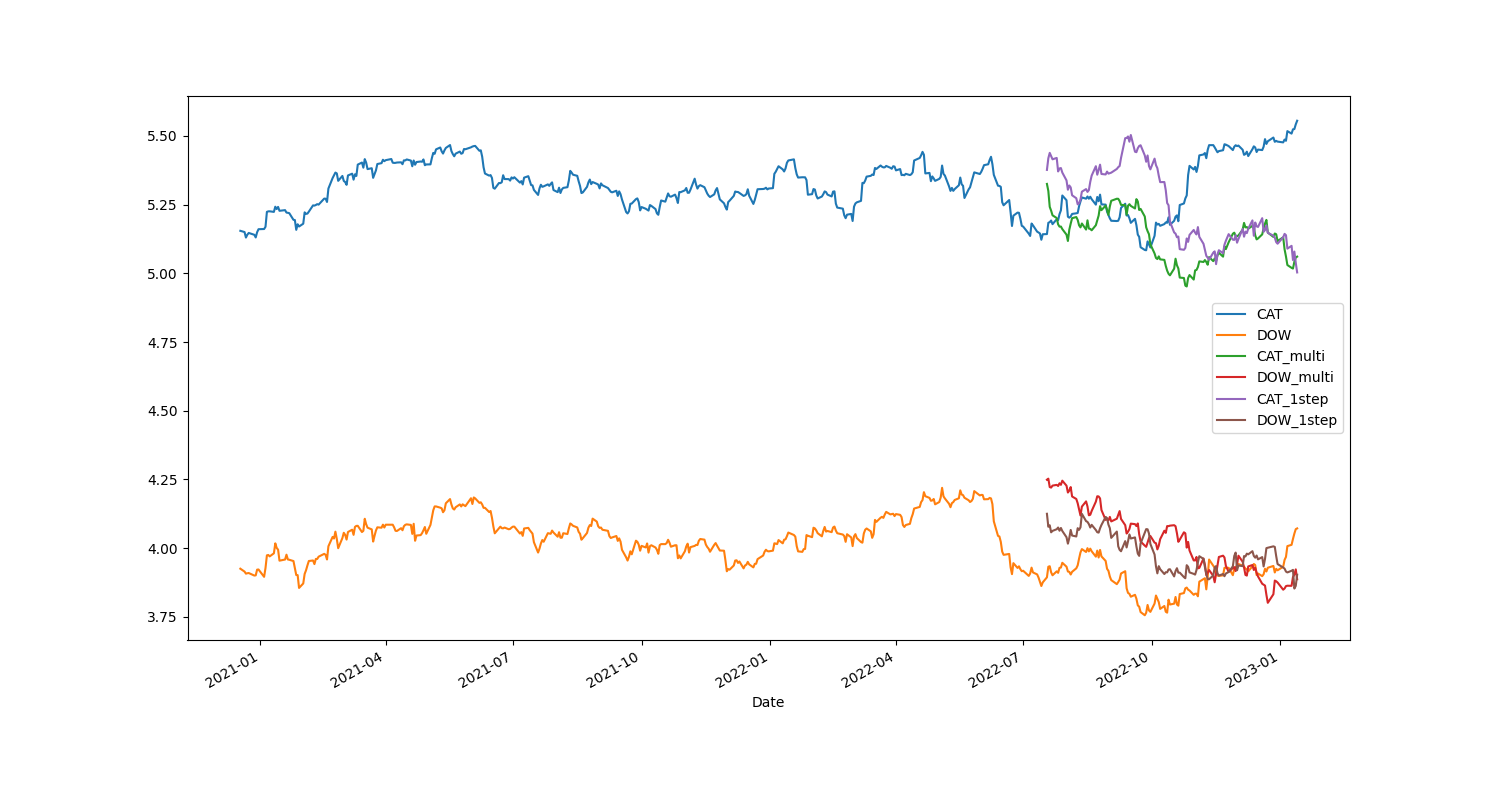
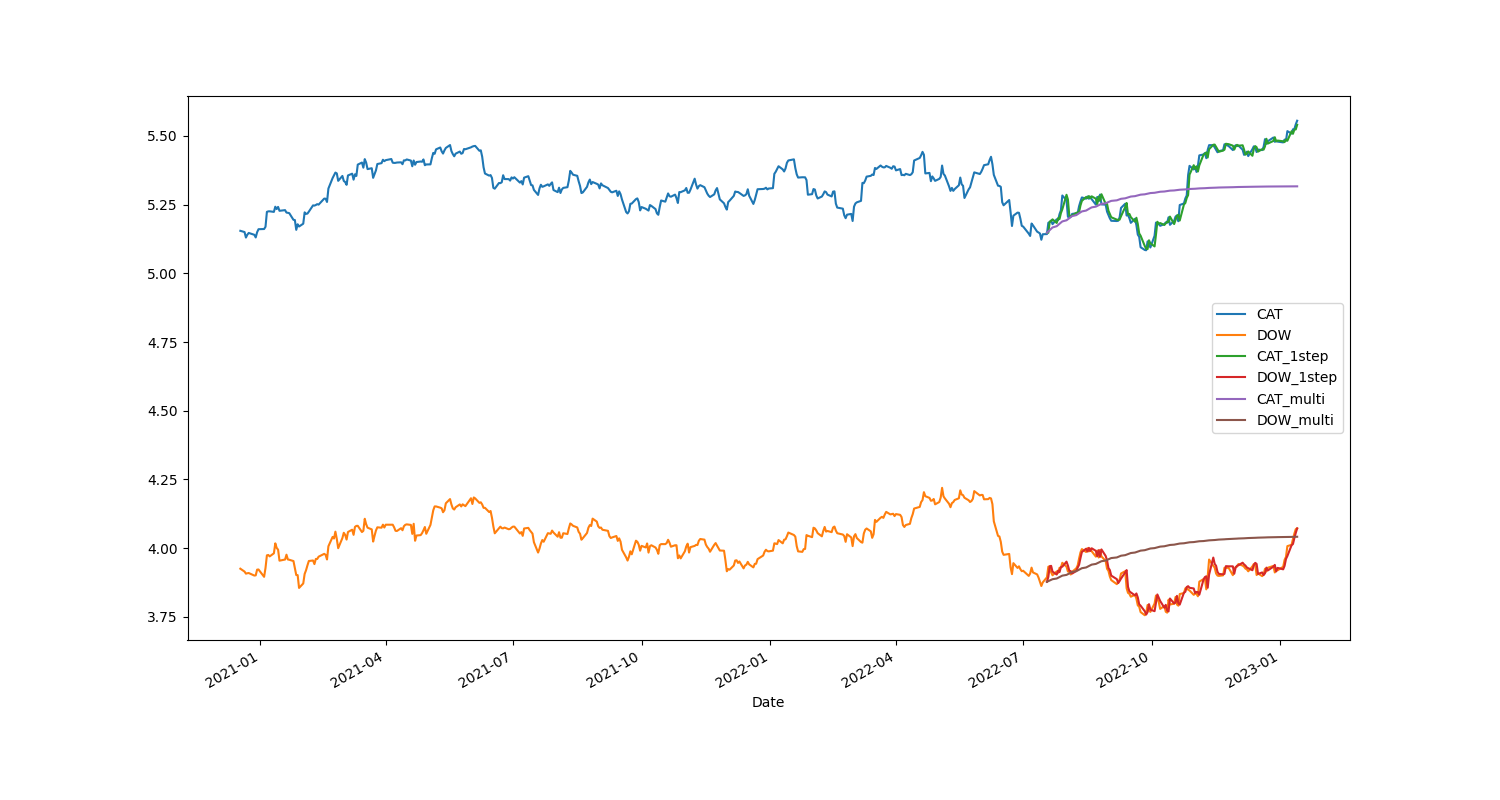
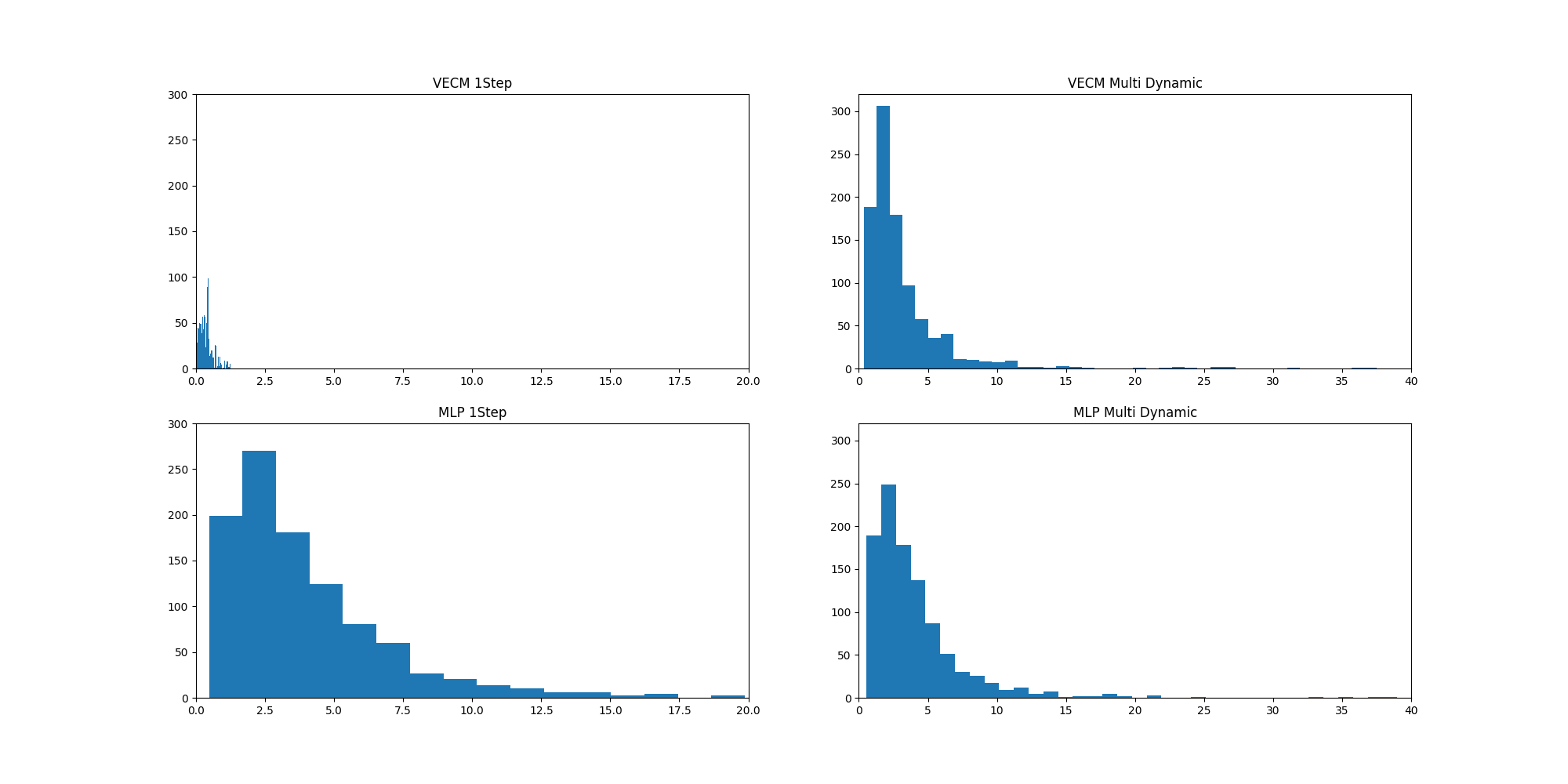
* در بخشی از پیش‌بینی‌های تک‌گام، هنگامی که دیتای جدید به دادهٔ آموزش اضافه می‌شود، دیگر سری‌ها هم‌انباشته نیستند. خطای این جفت سهام، برای مدل VECM در نظر گرفته نشده‌اند.
* همچنین در برخی از پیش‌بینی‌های تک‌گام، هنگامی که دادهٔ جدید به دادهٔ آموزش اضافه می‌شود، ممکن است دادهٔ جدید مقدار null داشته باشد. مدل MLPRegressor این مشکل را خودبه‌خود حل می‌کند اما مدل VECM خطا می‌دهد. بنابراین خطای این جفت سهام، برای مدل VECM در نظر گرفته نشده‌اند.
* آموزش مدل MLPRegressor بسیار سخت بود. در واقع یافتن پارامترهای بهینه که در مجموع ۲۰۳۶ پیش‌بینی، بهترین نتیجه را ارائه دهد طاقت‌فرسا بود. خروجی‌های موجود این مدل هم‌چنان نیاز به بهبود دارند و راضی‌کننده نیستند.

## خروجی

در مجموع ۲۰۳۶ پیش‌بینی انجام شده است (چندخروجی به صورت تک‌گام و چندگام). به ازای هر جفت سهام هم‌انباشته، یک نمودار ذخیره شده است. نمودارهای مربوط به پیش‌بینی‌های VECM در پوشهٔ ./02\_vecm\_vs\_neurals/preds\_vecm و نمودارهای مدل شبکه‌های عصبی در پوشهٔ ./02\_vecm\_vs\_neurals/preds\_nn ذخیره شده است.

هم‌چنین خطای MAPE پیش‌بینی‌ها در فایل اکسل ./02\_vecm\_vs\_neurals/nn\_mape.xlsx ذخیره شده‌اند.

در زیر، دو نمونه از نمودارهای پیش‌بینی، یکی برای مدل VECM و دیگری برای مدل شبکه‌های عصبی آورده شده است.

در زیر، نمودار هیستوگرام خطای MAPE برای همهٔ جفت سهام هم‌انباشته، نمایش داده شده است. همان‌طور که مشخص است، مدل VECM هم در پیش‌بینی تک‌گام و هم پیش‌بینی چندگام، به طور میانگین خطای کم‌تری دارد.

شکل 3 هیستوگرام خطای دو مدل VECM و MLP در دو نوع تک‌گام و چندگام

شکل ۲ نمودار پیش‌بینی مدل MLPRegressor برای دو سهم CAT و DOW

شکل ۱ نمودار پیش‌بینی مدل VECM برای دو سهم CAT و DOW

# پروژه ۳ – آزمون خطی بودن ارزهای دیجیتال

در این پروژه، تست خطی بودن برای سری‌های قیمتی ارزهای دیجیتال در بازه‌های زمانی متفاوت انجام می‌شود.

## راهنمای فایل‌ها

|  |  |
| --- | --- |
| توضیح | آدرس فایل |
| گزارش توضیح کد | ./03\_crypto\_nlntest/01\_cypto\_nlntest.pdf |
| فایل ژوپیتر نوت‌بوک کد همراه با توضیحات | ./03\_crypto\_nlntest/01\_cypto\_nlntest.ipynb |
| فایل پایتون کد همراه با توضیحات | ./03\_crypto\_nlntest/01\_cypto\_nlntest.py |
| پوشهٔ حاوی نمودارها و جداول نهایی | ./03\_crypto\_nlntest/output |

## ورودی‌ها

* قیمت‌های تاریخی ده ارز دیجیتال
* بازهٔ زمانی، از روز اول معاملهٔ هر ارز تا به امروز است.
* گام زمانی سری‌ها، روزانه است.

## روش اجرا

ابتدا سری‌های زمانی هر ارز دیجیتال، به تکه‌های یک‌ساله، دو ساله، سه ساله و پنج ساله تقسیم شدند. تست خطی بودن توسط پکیج nlntest بر روی هر کدام از این تکه‌ها اجرا شد. خروجی تابع nlntestuniv نتیجه چهار تست است. اگر حداقل دو تست، خطی بودن را رد کند، رای به غیرخطی بودن آن تکه از سری زمانی می‌دهیم.

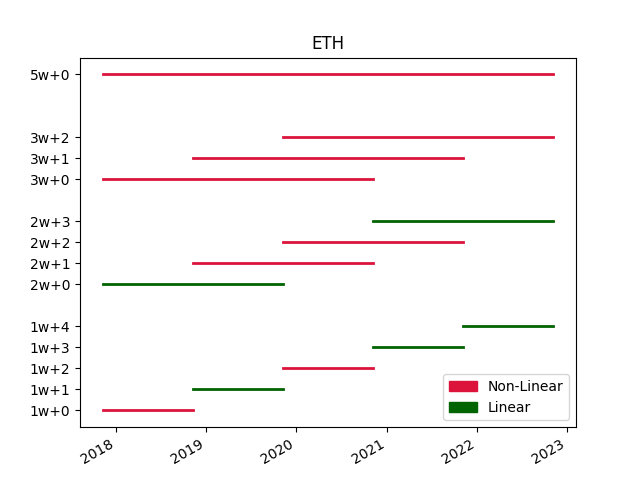
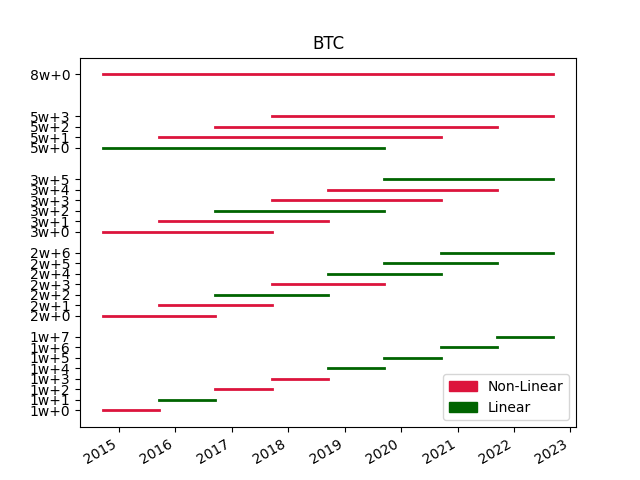
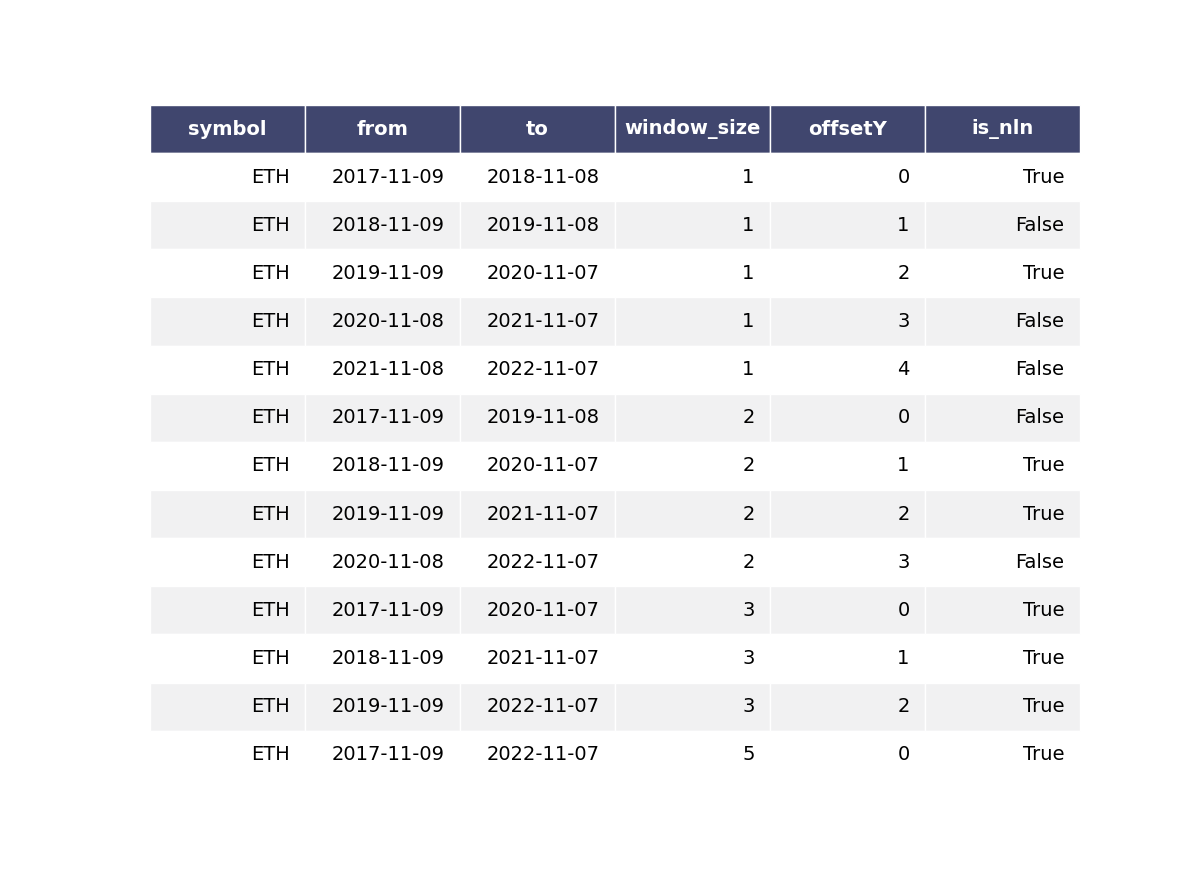
* برخی از توابع و کلاس‌های استفاده شدهٔ مهم:

|  |  |
| --- | --- |
| توضیح | تابع یا کلاس |
| این تابع با دریافت یک سری زمانی، چهار تست خطی بودن را روی آن اجرا می‌کند و p-value هر تست را به عنوان خروجی می‌دهد. | nlntest.nlntstuniv |

## خروجی‌ها

به ازای هر کدام از ارزهای دیجیتال، یک نمودار و یک جدول به صورت تصویر در پوشهٔ ./03\_crypto\_nlntest/output ذخیره شده‌اند. در زیر، دو نمونه از خروجی‌ها آورده شده است.

می‌توان نتیجه گرفت که خطی/غیرخطی بودن، وابسته به زمان است.

در نمودارهای زیر، محور افقی زمان است و در محور عمودی، تکه‌های از سری زمانی در پنجره‌ها و زمان‌های مختلف است. در بازه‌هایی که سری زمانی غیرخطی است، از رنگ قرمز و بازه‌هایی که خطی است از رنگ سبز برای نمایش استفاده شده است. در زیر، نمودار بیت‌کوین و اتریوم به همراه جداول مربوطه برای نمونه آورده شده است.

شکل 4 نتایج تست خطی بودن بیت‌کوین (جدول معادل با نمودار بالا)

شکل 5 نتایج تست خطی بودن بیت‌کوین (جدول معادل با نمودار بالا)

شکل 6 نتیجهٔ آزمون خطی بودن بیت‌کوین در تکه‌زمان‌های متفاوت

شکل 7 تست خطی بودن اتریوم در تکه‌زمان‌های متفاوت

# پروژه ۴ – یافتن توزیع Kernel Density مدل ECM

در این پروژه، سعی می‌شود توزیع Kernel Density مدل ECM برای جفت سهام هم‌انباشتهٔ پروژه شماره ۱ تخمین زده شود.

## راهنمای فایل‌ها

|  |  |
| --- | --- |
| توضیح | آدرس فایل |
| گزارش توضیح کد | ./04\_kernel\_dist/04\_dist.pdf |
| فایل ژوپیتر نوت‌بوک کد همراه با توضیحات | ./04\_kernel\_dist/04\_dist.ipynb |
| فایل پایتون کد همراه با توضیحات | ./04\_kernel\_dist/04\_dist.py |
| فایل اکسل حاوی نتایج تست‌ها | ./04\_kernel\_dist/ecm\_dists.xlsx |
| پوشهٔ حاوی نمودارها به تفکیک شاخص‌ | ./04\_kernel\_dist/plots/{index\_name} |

## ورودی‌ها

* ۵۰۹ جفت سهم هم‌انباشته از تمرین اول.
* بازهٔ زمانی در این تمرین (برخلاف تمرین‌های گذشته)، ۷۲۰ روز در نظر گرفته شده است. زیرا برای انجام آزمون‌های فرض، بهتر است که تعداد داده‌ها بیشتر باشد.
* بنابراین نیاز است که پیش از محاسبه مدل ECM، دوباره آزمون هم‌انباشتگی اجرا شود تا مشخص شود که آیا این سهام در این بازهٔ زمانی نیز هم‌انباشته هستند یا نه.

## روش اجرا

به ازای هر جفت سهم هم‌انباشته، چهار تست نرمال بودن شامل:‌jarque\_bera، anderson، cramervonmises و lilliefors انجام می‌گیرد. نتایج این چهار تست در نهایت در فایل اکسل ذخیره می‌شود.

هم‌چنین تست Kolmogorov-Smirnov در این تمرین به گونه‌ای پیاده‌سازی شده که بتواند علاوه بر نرمال بودن، توزیع‌های دیگر را شامل: lognorm, chi2, t, beta, gamma, weibull\_min آزمون کند. توضیحات مربوط به کدها، در گزارش کدها آورده شده است.

مراحل اجرای آزمون کولموگروف-اسمیرنوف به این شکل است:

* با توجه به kernel density مدل ECM، بهترین پارامترهای هر توزیع را می‌یابیم.
* به کمک تابع kstest در پکیج scipy تست می‌کنیم که آیا سری زمانی، با هر کدام از توزیع‌های ذکر شده در بالا (با پارامترهایی به دست آمده از مرحله قبل) چقدر شباهت دارد. فرض صفر این آزمون، شباهت این دو توزیع به هم است. اگر فرض صفر رد شود، می‌توان نتیجه گرفت که دو توزیع به هم شباهت ندارند. در این تمرین، هر جا فرض صفر رد نشود، نتیجه گرفته شده است که دو توزیع همانند هم هستند (که زیاد دقیق نیست اما برای رسم نمودارها و خروجی اکسل، این فرض مورد نیاز است).
* اگر توزیع kernel density با هر کدام از توزیع‌های ذکر شده شباهت داشته باشد، علاوه بر ذخیره کردن نام آن توزیع در فایل اکسل، نمودار kernel density به همراه آن توزیع رسم و ذخیره می‌شود.
* برخی از توابع و کلاس‌های استفاده شدهٔ مهم:

|  |  |
| --- | --- |
| توضیح | تابع یا کلاس |
| چهار تست نرمال بودن توزیع از پکیج scipy. فرض صفر همهٔ این تست‌ها، نرمال بودن توزیع دیتای ورودی به این توابع است. | scipy.stats.jarque\_bera  scipy.stats.anderson  scipy.stats.cramervonmises  scipy.stats.lilliefors |
| تست کولموگروف-اسمیرنوف برای شباهت دو توزیع | scipy.stats.kstest |

## خروجی‌ها

تعداد ۳۵۰ نمودار در پوشهٔ 04\_kernel\_dist/plots ذخیره شده‌اند. این نمودارها حاوی kernel density مدل ECM و توزیع‌های فیت‌شده بر آن هستند. هم‌چنین فایل اکسل ./04\_kernel\_dist/ecm\_dist.xlsx حاوی نتایج همهٔ تست‌های انجام شده است.

* از مجموع ۳۵۰ جفت سهم، ۱۶۶تای آن‌ها از هیچ‌کدام از توزیع‌های بالا پیروی نمی‌کنند.
* در جدول زیر، تعداد جفت سهامی که هر کدام از توزیع‌های ذکرشده بر kernel density آن‌ها به خوبی برازش شده‌اند، آورده شده است: (دقت شود که به ازای هر جفت سهم‌، ممکن است چند توزیع برازش شده باشد).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| weibull\_min | gamma | beta | t | chi2 | lognorma | normal |
| ۸۵ | ۴۰ | ۸۰ | ۱۳۵ | ۲۱ | ۵۰ | ۳۷ |

در زیر، برخی از نمودارهای خروجی به عنوان نمونه آورده شده است.

|  |  |
| --- | --- |
| نمودار | توزیع‌هایی که آزمون کولموگروف آن‌ها را رد نکرده‌اند |
|  | توزیع بتا |
|  | همه توزیع‌ها |
|  | توزیع تی-استیودنت  (سهام اخابر و رانفور) |
|  | هیچ‌کدام از توزیع‌ها |