دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی
مهندسی نرم‌افزار

پیش بینی تغییرات روند ارزهای رمزنگاری‌شده

نگارش

امیرحسین علی محمدی

استاد راهنما

جناب آقای دکتر احسان الدین عسگری

تیر ۱۴۰۲


بسم الله الرحمن الرحیم

سپاس

از استاد بزرگوارم که با کمک‌ها و راهنمایی‌های بی‌دریغشان، مرا در به سرانجام رساندن این پایان‌نامه یاری داده‌اند، تشکر و قدردانی می‌کنم. هم‌چنین از همکاران عزیزی که با راهنمایی‌های خود در بهبود نگارش این نوشتار سهیم بوده‌اند، صمیمانه سپاسگزارم.

چکیده

ارزهای دیجیتال در سال‌های اخیر به یک دارایی مهم تبدیل شده‌اند و توانایی پیش‌بینی قیمت آن‌ها مورد توجه سرمایه‌گذاران و معامله‌گران است. این مقاله به بررسی استفاده از مدل‌های یادگیری ماشین برای پیش‌بینی ارز دیجیتال می‌پردازد.

در این پایان‌نامه به معرفی و پیاده‌سازی کتابخانه CryptoPredictions می‌پردازیم که بستری را برای پیاده‌سازی و ارزیابی مدل‌های مختلف یادگیری ماشین برای پیش‌بینی قیمت ارزهای دیجیتال فراهم می‌کند. سپس مدل‌های مختلفی را که در کتابخانه پیاده‌سازی کردیم، از جمله جنگل تصمیم‌گیری تصادفی، حافظه کوتاه‌مدت، اوربیت، آریما، ساریمکس، XGBoost و پرافت را شرح می‌دهیم. طی آزمایش‌های گوناگون به این نتیجه رسیدیم که بهترین عملکرد مربوط به مدل پرافت می‌باشد.

بخش دیگر این پایان‌نامه استفاده از روش رگرسیون چندکی همدیس جهت پیش‌بینی تغییر روند بازار می‌باشد. این روش بازه‌های پیش‌بینی قیمت را تعریف می‌کند و به هر روز یا ساعت یک امتیاز سازگاری اختصاص می‌دهد که نشان می‌دهد بازه‌ی پیش‌بینی قیمت ما چگونه باید محاسبه شود. با استفاده از این بازه‌های پیش‌بینی می‌توان نقاطی را که روند در آن‌ها تغییر می‌کند شناسایی کرد.

پس از آن آزمایش‌هایی را انجام می‌دهیم تا اثربخشی روش جدیدمان را بسنجیم. طی این آزمایشات به این نتیجه می‌رسیم که بر خلاف فرض اولیه که احتمالا کاهش دقت مدل در هنگام تغییر روند را داریم، دقت مدل در نقاطی که در آن‌ها تغییر روند صورت می‌گیرد افزایش می‌یابد.

کلیدواژه‌ها: ارز دیجیتال، رمزارز، یادگیری ماشین، پیش‌بینی، پیش‌بینی روند

فهرست مطالب

فهرست جدول‌ها

فهرست شکل‌ها

فصل ۱

مقدمه

رمزارزها شکلی از ارز دیجیتال هستند که تولید واحدهای ارزی را تنظیم می‌کند و با استفاده از تکنیک‌های رمزنگاری، انتقال وجوه را احراز هویت می‌کند. قابل ذکر است که ارزهای دیجیتال توسط یک مقام مرکزی اداره نمی‌شوند و بر اساس ساختار غیرمتمرکز عمل می‌کنند. از زمان راه اندازی بیت کوین در سال ۲۰۰۹، ارزهای رمزنگاری شده روش انتقال پول توسط مردم را متحول کردند. ارز رمزنگاری شده برای اولین بار در سال ۱۹۹۸ توسط یک دانشمند کامپیوتر به نام وی دای پیشنهاد شد که یک سیستم مبتنی بر رمزنگاری ایجاد کرد که می‌تواند برای تسهیل پرداخت‌ها بین طرفین استفاده شود. این سیستم که "پول ب" [۱] نام دارد، راه را برای ارزهای رمزپایه آینده هموار کرد.

مشخصات ساختاری سیستماتیک بیت کوین [۱] در نوامبر ۲۰۰۸ توسط یک فرد یا گروه ناشناس با نام مستعار ساتوشی ناکاموتو منتشر شد. بیت کوین اولین ارز دیجیتالی بود که غیرمتمرکز شد. از زمان معرفی بیت کوین در سال ۲۰۰۹، ارزهای دیجیتال نحوه ارسال و دریافت پول را تغییر داده اند. با وجود ایجاد هزاران ارز دیجیتال دیگر و چندین نوسان قیمت از آن زمان، بیت کوین همچنان محبوب ترین و با ارزش ترین ارز دیجیتال در جهان است. در زمان نگارش این مقاله، ارزش بازار بیت کوین از ۴۷۵ میلیارد دلار فراتر رفته است. علاوه بر این، ارزش بازار تمام ارزهای دیجیتال فعال، از جمله بیت کوین، به ۱۷.۱ تریلیون دلار آمریکا می‌رسد [۲].

به دلیل ماهیت غیرمتمرکز اکثر ارزهای دیجیتال، قیمت آنها تحت تأثیر نرخ بهره، نرخ تورم یا سیاست‌های پولی نیست، بلکه بیشتر تحت تأثیر ادراک کاربران بر اساس اخبار، وب سایت‌ها و سایر عناصر غیر اساسی است [۳]. بازارهای سهام تحت تأثیر عوامل مختلفی هستند که باعث ایجاد عدم اطمینان می‌شوند، از جمله مسائل سیاسی و اقتصادی که تأثیر محلی یا جهانی دارند. درک کلیدهای موفقیت یا عواملی که

[۱] b-money

پیش‌بینی‌های دقیق را ارائه می‌دهند، کار دشواری است. ما می‌توانیم با استفاده از هر تکنیکی از جمله شاخص‌های فنی، نوسانات قیمت و تحلیل تکنیکال بازار، بازار را بررسی کنیم. بنابراین نیاز به ابزارهای پیش‌بینی خودکار برای کمک به سرمایه‌گذاران در تصمیم‌گیری برای سرمایه‌گذاری در بیت‌کوین یا سایر ارزهای دیجیتال وجود دارد. پیش‌بینی‌های مدرن بازار سهام معمولاً شامل فناوری‌های اتوماسیون می‌شوند و ما می‌توانیم همان رویکرد و استراتژی را در این حوزه از ارزهای دیجیتال اعمال کنیم.

یادگیری ماشین یک انتخاب قدرتمند و موثر برای استراتژی‌های معاملاتی [۴] است. توانایی آن در کشف روابط پنهان داده‌ای که ممکن است از مشاهدات انسانی گریزان باشد، آن را در پیش‌بینی خروجی‌های عددی مانند قیمت یا حجم و شناسایی خروجی‌های طبقه‌بندی شده مانند روندها ارزشمند می‌کند. با ارائه مدل با داده‌های ورودی اکتشافی، معامله گران می‌توانند از طیف گسترده ای از مدل‌های یادگیری ماشین برای به دست آوردن بینش و تصمیم گیری آگاهانه در معاملات استفاده کنند.

چندین مدل یادگیری ماشینی در تجارت موفق بوده اند. مدل‌های رگرسیون، از جمله رگرسیون خطی [۵] و پشتیبان رگرسیون برداری [۲] [۶]، تخمین دقیق حرکت قیمت را بر اساس داده‌های تاریخی ارائه می‌دهند. مدل‌های طبقه‌بندی مانند درخت‌های تصمیم [۷] و جنگل‌های تصادفی [۸] در شناسایی روندهای بازار و پیش‌بینی‌های طبقه‌بندی برتری دارند. شبکه‌های عصبی، مانند مدل‌های یادگیری عمیق [۹]، در ثبت الگوهای پیچیده در داده‌های مالی مهارت بالایی دارند.

تحقیقات گسترده کارآمدی یادگیری ماشین را در معاملات نشان داده است، با مطالعاتی که عملکرد برتر و بازدهی بالاتری را در مقایسه با استراتژی‌های سنتی [۱۰] [۱۱] نشان می‌دهد. علاوه بر این، تکنیک‌های یادگیری ماشین برای تجزیه و تحلیل منابع داده جایگزین مانند احساسات در رسانه‌های اجتماعی [۱۲] و مقالات خبری [۱۳] برای به دست آوردن مزیت رقابتی در بازار استفاده شده است.

یادگیری ماشین مجموعه متنوعی از مدل‌ها و تکنیک‌ها را در اختیار معامله‌گران قرار می‌دهد که استراتژی‌های معاملاتی را بهبود می‌بخشد. همانطور که فناوری به پیشرفت خود ادامه می‌دهد و داده‌های بیشتری در دسترس قرار می‌گیرد، انتظار می‌رود نقش یادگیری ماشینی در بازارهای مالی به طور قابل توجهی رشد کند.

۱ـ۱ تعریف مسئله

در این پروژه ما با دو مسئله مواجه شدیم. مسئله‌ی اول بررسی مدل‌های موجود بر روی داده‌های رمزارزهای گوناگون و ارزیابی روش‌ها می‌باشد. مسئله‌ی دوم پیشبینی تغییرات روند بازار است که برای این منظور باید تعریفی برای روند در بازار پیدا می‌کردیم. در فصل روش‌شناسی تعاریف موجود برای روند در بازارهای

[۲] Support Vector Machine

مالی آمده است اما روش انتخابی ما استفاده از رگرسیون چندکی همدیس بود که بازه‌های پیش‌بینی برای خروجی مدل ما تعریف می‌کند و ما تغییر روند را خروج قیمت واقعی ارز از آن بازه تعریف می‌کنیم. در فصل آزمایش‌های مربوط به پیش‌بینی روند بازار به ارزیابی کیفیت تعریف ارائه‌شده و نتایج آزمایش‌های مربوط به روش رگرسیون چند همدیسی خواهیم پرداهت.

۱ـ۲ اهمیت موضوع

دو مورد از اهمیت‌های قابل توجه پیش‌بینی تغییرات روند در بازار ارزهای دیجیتال عبارت است از:

۱. **برای کمک به سرمایه گذاران در تصمیم گیری بهتر**
با پیش‌بینی روند قیمت در آینده، سرمایه گذاران می‌توانند تصمیمات آگاهانه تری در مورد زمان خرید، فروش یا نگهداری ارزهای دیجیتال بگیرند. این می‌تواند به آنها کمک کند تا زیان خود را به حداقل برسانند و سود خود را به حداکثر برسانند.

۲. **برای توسعه استراتژی‌های تجاری جدید**
یادگیری ماشینی می‌تواند برای توسعه استراتژی‌های معاملاتی جدید که بر اساس داده‌های قیمت تاریخی و سایر عوامل است، استفاده شود. این استراتژی‌ها می‌تواند به معامله گران کمک کند تا معاملات سودآورتری داشته باشند.

به صورت کل استفاده از یادگیری ماشین مزایای خاص زیر را دارد که می‌تواند نشان‌دهنده اهمیت استفاده از آن در پیش‌بینی تغییرات روند در رمزارزها باشد.

- **دقت** : الگوریتم‌های یادگیری ماشینی را می‌توان بر روی داده‌های تاریخی آموزش داد تا الگوهای افزایش قیمت ارزهای دیجیتال را بیاموزند. این می‌تواند به بهبود دقت پیش‌بینی‌ها کمک کند.
- **مقیاس‌پذیری** : الگوریتم‌های یادگیری ماشینی را می‌توان برای مدیریت حجم زیادی از داده‌ها مقیاس بندی کرد. این برای بازارهای ارزهای دیجیتال که دائما در حال تولید داده‌های جدید هستند، مهم است.
- **سفارشی سازی** : الگوریتم‌های یادگیری ماشین را می‌توان برای رفع نیازهای خاص سرمایه‌گذاران یا معامله‌گران سفارشی کرد. این به آن‌ها اجازه می‌دهد تا استراتژی‌هایی را توسعه دهند که برای تحمل ریسک و اهداف سرمایه‌گذاری فردی آنها طراحی شده است.

۱ـ۳ اهداف پژوهش

پروژه شامل دو بخش است که در بخش اول هدف بررسی میزان پیش‌بینی پذیر بودن داده‌های اقتصادی و ارائه بهترین مدل‌های موجود برای پیش‌بینی چنین داده‌هایی است. دربخش دوم توجه به بررسی تغییر حالات بازار و تاثیرات آن در پیش‌بینی داده‌های اقتصادی معطوف است. با توجه به پیچیدگی‌های موجود در بازارها خصوصا در بازار رمز ارزهای دیجیتال، بخش دوم پروژه با مدل سازی بر روی داده‌های بیت‌کوین صورت گرفته است. لازم به ذکر است که ماهیت چنین مسائلی بسیار پیچیده تر از آن است که بتوان به یک مدل پیش‌بینی قابل اطمینان برای چنین داده‌های اقتصادی رسید، که اگر غیر از این بود شرکت‌های فعال مطرح در حوزه خدمات اقتصادی دیگر دغدغه‌ای نداشتند. لذا توجه به این نکته بسیار مهم است که ماحصل این پروژه تحقیقاتی نه لزوما ارائه مدل پیش‌بینی قابل اعتماد و دقیق برای پیش‌بینی بازار و تغییر حالات آن، که صرفا مطالعه ای بر اعمال فنون یادگیری ماشین موجود بر داده‌های اقتصادی بوده، به امید آنکه مدل‌های به دست آمده از کیفیت پیش‌بینی مطلوب نیز برخوردار باشند.

۱ـ۴ ساختار پایان‌نامه

این پایان‌نامه در ۶ فصل ارائه شده است. در فصل ۲ به مفاهیم اولیه برای درک روش رگرسیون چندکی همدیس می‌پردازیم و سپس متریک‌هایی که برای ارزیابی مدل‌ها استفاده می‌شوند را معرفی می‌کنیم. در فصل ۳ به مدل‌های معرفی شده در زمینه هوش مصنوعی که برای پیش‌بینی قیمت ارزهای دیجیتال استفاده می‌شوند و در کتابخانه‌مان هم پیاده سازی شده است می‌پردازیم. در فصل ۴ به جزئیات کتابخانه‌ای که آن را پیاده‌سازی کرده‌ایم می‌پردازیم و سپس روش انتخابی برای پیش‌بینی تغییرات روند را با جزئیات کافی شرح می‌دهیم. در فصل ۵ به ارزیابی مدل‌های معرفی‌شده بر روی رمزارزهای گوناگون می‌پردازیم. و سپس در فصل ۶ نتایج آزمایشات مربوط به پیش‌بینی تغییرات روند بازار با استفاده از رگرسیون چندکی همدیس را مورد بررسی قرار می‌دهیم.

فصل ۲

مفاهیم اولیه

در این بخش به معرفی پیش‌نیازهای درک روش رگرسیون چندکی همدیس اشاره می‌کنیم که روش انتخابی برای پیش‌بینی تغییرات روند در قیمت رمزارزها استفاده می‌شود. سپس متریک‌هایی که برای ارزیابی مدل‌های مختلف هوش مصنوعی به صورت گسترده استفاده می‌شوند را معرفی می‌کنیم.

۲ـ۱ رگرسیون چندک

هدف از رگرسیون چندک شرطی [۱۴] [۱] تخمین یک چندک معین مانند میانه Y مشروط بر X است. به یاد بیاورید که تابع توزیع شرطی Y با توجه به X = x است.

$$F(y|X = x) := P\{Y \leqslant y|X = x\},$$

و این که α امین تابع چندک شرطی

$$q_\alpha(x) := inf\{y \in R : F(y|X = x) \geqslant \alpha\}$$

چندک پایین و بالا را به ترتیب برابر با $q_{\alpha_{lo}} = \alpha/۲$ و $q_\alpha = ۱ - \alpha/۲$ قرار می‌دهیم. با داشتن جفت $q_{\alpha_{lo}}$ و $q_{\alpha_{hi}}$ به عنوان تابع چندک شرطی می‌توانیم بازه پیشبینی شرطی Y را به شرط $X = x$ با پوشش نادرست α با

$$C(x) := [q_{\alpha_{lo}}(X),\ q_{\alpha_{hi}}(X)]$$

---

[۱] conditional quantile regression

به دست آورد.

با تعریف این تابع، شرط

$$C(x) := P\{Y \in C(X)|X = x\} \geqslant ۱ - \alpha$$

ارضا می‌شود. همچنین باید به این مسئله توجه شود که طول بازه‌ی $C(x)$ می‌تواند تا حد زیادی به مقدار x بستگی داشته باشد. عدم قطعیت در پیش‌بینی Y به طور طبیعی در طول بازه منعکس می‌شود.

تخمین چندک از داده

تحلیل رگرسیون کلاسیک میانگین شرطی پاسخ آزمون $Y_{n+۱}$ را با توجه به ویژگی‌های $Xn+۱=x$ با به حداقل رساندن مجموع مجذور باقیمانده در n نقطه آموزشی تخمین می‌زند:

$$\hat{\mu} = \mu(x; \hat{\theta}), \qquad \hat{\theta} = argmin_\theta \frac{۱}{n} \sum_{i=۱}^{n} (Y_i - \mu(X_i; \theta))^۲ + R(\theta)$$

در اینجا θ پارامتر مدل رگرسیون است و $\mu(x; \theta)$ تابع رگرسیون است و همچنین R یک تنظیم کننده بالقوه [۲] است.

به طور مشابه رگرسیون چندک یک تابع چند شرطی q_α از $Y_{n+۱}$ به شرط $X_{n+۱} = x$ را تخمین می‌زند.

$$\hat{q}_\alpha(x) = f(x, \hat{\theta}), \qquad \hat{\theta} = argmin_\theta \frac{۱}{n} \sum_{i=۱}^{n} p_\alpha(Y_i, f(X_i; \theta)) + R(\theta)$$

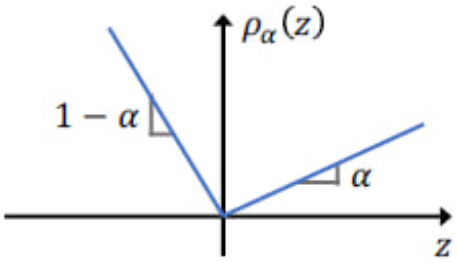
به گونه‌ای که $f(x, \hat{\theta})$ تابع رگرسیون چندک است و p_α تابع ضرر "تابع چک" [۳] [۱۴] [۱۵] است که توسط

$$p_\alpha(y, \hat{y}) := \begin{cases} \alpha(y - \hat{y}), & \text{اگر } y - \hat{y} > ۰, \\ (۱ - \alpha)(\hat{y} - y), & \text{در غیر این صورت} \end{cases}$$

تعریف می‌شود و در شکل ۲ ـ ۱ نمایش داده شده است. سادگی و عمومیت این فرمول باعث می‌شود که رگرسیون چندک به طور گسترده ای قابل اجرا باشد. مشابه رگرسیون کلاسیک، می‌توان از طیف وسیعی از روش‌های یادگیری ماشین برای طراحی و یادگیری $\hat{q}_\alpha$ استفاده کرد [۱۶] [۱۷] [۱۸] [۱۹] [۲۰].

همه موارد بالا یک استراتژی واضح را برای ایجاد یک باند پیش‌بینی با نرخ پوشش نامی نشان می‌دهد: ابتدا $\hat{q_{\alpha_{lo}}}(x)$ و $\hat{q_{\alpha_{hi}}}(x)$ را با رگرسیون چند تخمین می‌زنیم، سپس $\hat{C}(X_{n+۱}) = [q_{\alpha_{lo}}(\hat{X}_{n+۱}), q_{\alpha_{hi}}(\hat{X}_{n+۱})]$ را به عنوان تخمین خروجی محاسبه می‌کنیم. این رویکرد به طور گسترده قابل اجرا است و اغلب در عمل به خوبی کار می‌کند و فواصل زمانی را به وجود می‌آورد که با ناهمسانی سازگار است. با این حال، زمانی

[۲] potential regularizer
[۳] check function



شکل ۲ـ۱: تصویر تابع چک هنگامی که $z = y - \hat{y}$

که $C(X)$ با فاصله تخمینی $\hat{C}(X_{n+1})$ جایگزین شود، تضمینی برای برآورده کردن فرض پوشش [۴] وجود ندارد. در واقع، عدم وجود هر گونه تضمین برای تعداد نمونه محدود گاهی اوقات می‌تواند مشکل‌ساز باشد. این نگرانی توسط آزمایش‌های گوناگون تأیید می‌شود، که نشان می‌دهد فواصل ایجاد شده توسط شبکه‌های عصبی می‌توانند به طور قابل‌توجهی پنهان شوند.

تحت شرایط منظم خاص و برای مدل‌های خاص، تخمین توابع چندک شرطی از طریق تابع چک به طور مجانبی ثابت است [۱۹] [۱۵] . روش‌های مرتبطی که از تابع چک را به حداقل نمی‌رسانند، مانند جنگل‌های تصادفی چندکی [۵] [۲۱] ، نیز مجانبی سازگار هستند. اما برای به دست آوردن پوشش معتبر در نمونه‌های محدود، باید از مجموعه ای متفاوت از ایده‌ها استفاده کنیم، از پیش‌بینی همدیس.

۲ـ۲ پیش‌بینی همدیس

پیش‌بینی همدیس [۲۲] [۲۳] فواصل پیش‌بینی را به گونه‌ای ایجاد می‌کند که ضمانت پوشش در نمونه‌های محمدود رعایت شود. برای این که رویه اصلی یا کامل به صورت دقیق اجرا شود باید الگوریتم رگرسیون بینهایت بار فراخوانی شود. در مقابل، روش تقسیم، یا استقرایی، پیش‌بینی همدیس [۲۴] [۲۵] از این مشکل، به قیمت تقسیم داده‌ها جلوگیری می‌کند.

بر اساس مفروضات ارائه شده در [۲۶] ، روش تقسیم همدیس [۶] با تقسیم داده‌های آموزشی به دو زیرمجموعه مجزا آغاز می‌شود: یک مجموعه آموزشی مناسب $\{(X_i, Y_i) : i \in I_1\}$ و مجموعه کالیبراسیون $\{(X_i, Y_i) : i \in I_2\}$. سپس، با توجه به هر الگوریتم رگرسیون A [۷] ،یک مدل رگرسیون برای مجموعه

[۴] coverage
[۵] quantile random forests
[۶] split conformal method
[۷] در پیش‌بینی کاملاً همدیس، الگوریتم رگرسیون باید داده‌ها را به صورت مبادله‌ای رفتار کند، اما چنین محدودیتی برای پیش‌بینی همدیس تقسیم‌بندی اعمال نمی‌شود.

آموزشی برازش می‌شود:

$$\hat{\mu}(x) \Leftarrow A(\{(X_i, Y_i) : i \in I_1\})$$

سپس، باقیمانده مطلق بر روی مجموعه کالیبراسیون به شرح زیر محاسبه می‌شود

$$R_i = |Y_i - \hat{\mu}(X_i)|, \qquad i \in I_2.$$

برای یک سطح معین α، سپس چندک از توزیع تجربی باقیمانده‌های مطلق را محاسبه می‌کنیم

$$Q_{1-\alpha}(R, I_2) := (1 - \alpha)(1 + 1/|I_2|).$$

در آخر بازه‌ی پیش‌بینی برای نقطه جدید X_{n+1} توسط

$$C(X_{n+1}) = [\hat{\mu}(X_{n+1}) - Q_{1-\alpha}(R, I_2), \hat{\mu}(X_{n+1}) + Q_{1-\alpha}(R, I_2)]$$

به دست می‌آید.

با نگاهی دقیق تر به فاصله پیش‌بینی فرمول بالا متوجه یک محدودیت عمده در این روش می‌شویم: طول $C(X_{n+1})$ ثابت و برابر با $2Q_{1-\alpha}(R, I_2)$ مستقل از X_{n+1} است. لی و همکاران [۱۵] مشاهده کردند که فواصل تولید شده توسط روش کاملاً همدیس نیز فقط کمی با X_{n+1} متفاوت است، مشروط بر اینکه الگوریتم رگرسیون نسبتاً پایدار باشد. این ما را به این نتیجه می‌رساند که از رویکردی اصولی برای ساخت فواصل پیش‌بینی همدیس با عرض متغیر استفاده کنیم.

۲ـ۳ رگرسیون چندکی همدیس

در این بخش روش رگرسیون چندکی همدیس [۸] بررسی می‌شود. این روش با تقسیم داده به دو بخش مجموعه آموزشی I_1 و مجموعه کالیبراسیون I_2 شروع می‌شود. با داشتن الگوریتم رگرسیون چندک A، دو تابع چندک شرطی $\hat{q_{\alpha_{lo}}}$ و $\hat{q_{\alpha_{hi}}}$ را روی مجموعه آموزشی برازش می‌کنیم.

$$\{\hat{q}_{\alpha_{lo}}, \hat{q}_{\alpha_{hi}}\} \Leftarrow A(\{(X_i, Y_i) : i \in I_1\})$$

در گام بعد، ما نمرات انطباق [۹] که وظیفه تعیین کمیت خطای ایجاد شده توسط پیش‌بینی افزوده‌شده را دارد برای هر $i \in I_2$ حساب می‌کنیم.

$$E_i := max\{\hat{q}_{\alpha_{lo}}(X_i) - Y_i, Y_i - \hat{q}_{\alpha_{hi}}(X_i)\}$$

[۸] Conformalized Quantile Regression (CQR)
[۹] conformity scores

نمره انطباق E_i تفسیر زیر را دارد. اگر Y_i زیر نقطه پایینی پایین بازه باشد آنگاه $Y_i < \hat{q}_{\alpha_{lo}}(X_i)$ که در نتیجه $E_i = |Y_i - \hat{q}_{\alpha_{lo}}(X_i)|$ اندازه خطای رخ‌داده می‌باشد.

به طور مشابه، اگر Y_i بالاتر از نقطه بالایی بالای بازه باشد آنگاه $Y_i > \hat{q}_{\alpha_{hi}}(X_i)$ و $E_i = |Y_i - \hat{q}_{\alpha_{hi}}(X_i)|$. در آخر اگر Y_i عضو بازه‌ی $[\hat{q}_{\alpha_{lo}}(X_i), \hat{q}_{\alpha_{hi}}(X_i)]$ باشد، آنگاه E_i عدد بزرگ‌تر در مجموعه‌ی دو عدد نامنفی $[Y_i - \hat{q}_{\alpha_{lo}}(X_i), \hat{q}_{\alpha_{hi}}(X_i) - Y_i]$ که در نتیجه خودش هم نا منفی است. در نهایت، با توجه به داده‌های ورودی جدید X_{n+1}، فاصله پیش‌بینی برای Y_{n+1} را به شکل زیر می‌سازیم

$$C(X_{n+1}) = [\hat{q}_{\alpha_{lo}}(X_{n+1}) - Q_{1-\alpha}(E, I_2), \hat{q}_{\alpha_{hi}}(X_{n+1}) + Q_{1-\alpha}(E, I_2)]$$

به طوری که $Q_{1-\alpha}(E, I_2)$ برابر است با $(1-\alpha)(1+1/|I_2|)$ امین چندک تجربی مجموعه $\{E_i : i \in I_2\}$ است.

قضیه‌ی ۲ـ۱ *اگر (X_i, Y_i) و $i = 1, ..., n$ قابل تعویض باشند، آنگاه بازه پیش‌بینی $C(X_{n+1})$ به‌دست آمده توسط الگوریتم رگرسیون چندکی همدیس شرط زیر را ارضا می‌کند.*

$$P\{Y_{n+1} \in C(X_{n+1}) \geqslant 1 - \alpha\}$$

همچنین اگر نمرات انطباق E_i مجزا باشند، آنگاه فاصله پیش‌بینی تقریباً کاملاً کالیبره شده است یعنی:

$$P\{Y_{n+1} \in C(X_{n+1}) \leqslant 1 - \alpha + \frac{1}{|I_2| + 1}\}.$$

اثبات:

فرض کنید E_{n+1} نمره انطباق نقطه (X_{n+1}, Y_{n+1}) در مجموعه تست باشد. با ساختن بازه پیش‌بینی داریم:

$$Y_{n+1} \in C(X_{n+1}) \quad \text{اگر و تنها اگر} \quad E_{n+1} \leqslant Q_{1-\alpha}(E, I_2),$$

و به صورت مشخص

$$P\{Y_{n+1} \in C(X_{n+1}) | (X_i, Y_i) : i \in I_1\} = P\{E_{n+1} \leqslant Q_{1-\alpha}(E, I_2) | (X_i, Y_i) : i \in I_1\}.$$

حال چون جفت‌های (X_i, Y_i) قابل تعویض هستند در نتیجه E_i برای $i \in I_2$ و $i = n + 1$. در نتیجه طبق لم ۲ بر روی چندهای تجربی (در پیوست A مقاله [۲۶])

$$P\{E_{n+1} \leqslant Q_{1-\alpha}(E, I_2) | (X_i, Y_i) : i \in I_1\} \geqslant 1 - \alpha$$

و تحت فرض اضافه این که E_i ها متمایز هستند

$$P\{E_{n+۱} \leqslant Q_{۱-\alpha}(E, I_۲)|(X_i, Y_i) : i \in I_۱\} \leqslant ۱ - \alpha + \frac{۱}{|I_۲| + ۱}$$

قضیه‌ی ۲ـ۲ *با تعریف بازه‌ی پیش‌بینی*

$$C(X_{n+۱}) = [\hat{q}_{\alpha_{lo}}(X_{n+۱}) - Q_{۱-\alpha}(E, I_۲), \hat{q}_{\alpha_{hi}}(X_{n+۱}) + Q_{۱-\alpha}(E, I_۲)]$$

به نحوی که $Q_{۱-\alpha}(E_{lo}, I_۲)$ برابر است با $(۱ - \alpha_{lo})$ مین چندک تجربی $\{\hat{q}_{\alpha_{lo}}(X_i) - Y_i : i \in I_۲\}$ و $Q_{۱-\alpha}(E_{hi}, I_۲)$ برابر است با $(۱ - \alpha_{hi})$ مین چندک تجربی $\{\hat{q}_{\alpha_{hi}}(X_i) - Y_i : i \in I_۲\}$. اگر نمونه‌ها (X_i, Y_i) و $i = ۱, ..., n$ قابل تعویض باشند، آنگاه داریم

$$P\{Y_{n+۱} \geqslant \hat{q}_{\alpha_{lo}}(X_{n+۱}) - Q_{۱-\alpha_{lo}}(E_{lo}, I_۲)\} \geqslant ۱ - \alpha_{lo}$$

و

$$P\{Y_{n+۱} \leqslant \hat{q}_{\alpha_{lo}}(X_{n+۱}) + Q_{۱-\alpha_{hi}}(E_{lo}, I_۲)\} \geqslant ۱ - \alpha_{hi}$$

در نتیجه با فرض $\alpha = \alpha_{lo} + \alpha_{hi}$ داریم که $p\{Y_{n+۱} \in C(X_{n+۱})\} \geqslant ۱ - \alpha$

اثبات:

دو فرمول بالا به ترتیب معادل $\hat{q}_{\alpha_{lo}} - Y_{n+۱} \leqslant Q_{۱-\alpha_{lo}}(E_{lo}, I_۲)$ و $\hat{q}_{\alpha_{hi}} - Y_{n+۱} \leqslant Q_{۱-\alpha_{lo}}(E_{hi}, I_۲)$ هستند. بنابراین می‌توانیم لم ۲ را دو بار، به همان روشی که در اثبات قضیه ۱ انجام دادیم، اعمال کنیم.

ملاحظات عملی

رگرسیون چندک همدیس می‌تواند طیف وسیعی از روش‌های رگرسیون چندک [۱۴] [۱۶] [۱۷] [۱۸] [۲۱] [۱۹] [۲۷] [۲۰] را برای تخمین توابع چندک شرطی، q_{lo} و q_{hi} در خود جای دهد. برآوردگرها حتی می‌توانند مجموعه ای از الگوریتم‌های رگرسیون چندکی مختلف باشند.

از آنجا که الگوریتم رگرسیون چندک زیربنایی ممکن است مجموعه آموزشی مناسب را به روش‌های دلخواه پردازش کند، چارچوب این روش انعطاف‌پذیری گسترده‌ای را در تنظیم فراپارامتر فراهم می‌کند. به عنوان مثال تنظیم فراپارامترهای معمولی شبکه‌های عصبی، مانند اندازه دسته، نرخ یادگیری و تعداد دوره‌ها. ابرپارامترها ممکن است، طبق معمول، با اعتبارسنجی متقاطع [۱۰] انتخاب شوند، جایی که ما میانگین طول بازه را بر روی چین‌ها [۱۱] به حداقل می‌رسانیم. در این راستا، دو جزئیات پیاده سازی خاص بیان می‌شود.

[۱۰] cross-validation
[۱۱] folds

۱. رگرسیون چندکی گاهی بیش از حد محافظه کارانه است که منجر به فواصل پیش‌بینی غیر ضروری می‌شود. در تجربه ما، جنگل‌های رگرسیون چندک [۲۱] اغلب بیش از حد محافظه‌کار هستند و شبکه‌های عصبی چندک [۱۷] گاهی اوقات چنین هستند. ما می‌توانیم این مشکل را با تنظیم چندک‌های اسمی به‌عنوان فراپارامترهای اضافی در اعتبارسنجی متقاطع کاهش دهیم. قابل ذکر است، این تنظیم ضمانت پوشش را باطل نمی‌کند، اما ممکن است فواصل کوتاه تری را به همراه داشته باشد، همانطور که بعضی آزمایشات تأیید می‌شود.

۲. برای کاهش هزینه محاسباتی، به جای برازش دو شبکه عصبی مجزا برای تخمین توابع چندک پایین و بالایی، می‌توانیم تخمین تک بعدی استاندارد پاسخ مجهول را با یک تخمین دو بعدی از چندک‌های پایین و بالایی جایگزین کنیم. به این ترتیب، بیشتر پارامترهای شبکه بین دو تخمین‌گر کمیت به اشتراک گذاشته می‌شوند.

۲ـ۴ متریک‌ها

پس از به دست آوردن پیش‌بینی‌های نهایی مدل، اعتبار سنجی داده‌ها معمولاً با محاسبه متریک‌های زیر انجام می‌شود:

میانگین خطای مطلق این میانگین قدر مطلق تفاوت بین قیمت پیش‌بینی شده و ارزش واقعی است. تفسیر آن آسان است و با ارائه خطا در واحدهای داده‌ها و پیش‌بینی به شما سود می‌رساند. با این حال، موارد پرت را جریمه نمی‌کند (که می‌تواند در پیش‌بینی قیمت خیلی مهم نباشد). مهم‌ترین اشکال این معیار این است که به مقیاس وابسته است، بنابراین نمی‌توانیم ارزهای دیجیتال مختلف را با واحدهای مختلف مقایسه کنیم.

$$MAE = \frac{۱}{n}\sum_{i=۱}^{n}|A_i - F_i|$$

میانگین مربعات خطا این روش میانگین مجذور تفاوت‌های قیمت پیش‌بینی‌شده و ارزش واقعی است. در این معیار، داده‌های پرت به شدت مجازات می‌شوند. از سوی دیگر، از آنجایی که خطا در واحدهای اصلی داده‌ها و پیش‌بینی نیست، تفسیر آن دشوارتر است. همچنین وابسته به مقیاس است، بنابراین مشکل مشابهی مانند میانگین خطای مطلق داریم.

$$MSE = \frac{۱}{n}\sum_{i=۱}^{n}(A_i - F_i)^۲$$

ریشه میانگین مربعات خطا این همان خطای میانگین مربعات است، جدا از اینکه در پایان نتیجه را جذر می‌دهیم. در این متریک، نقاط پرت به شدت مانند خطای میانگین مربعات مجازات می‌شوند و نقطه قوت آن قرار گرفتن در واحدهای داده و پیش‌بینی است. این به نوعی بهترین دنیای خطای میانگین مربعات و میانگین درصد مطلق خطا است. اما، از آنجایی که خطا را مربع می‌کنید، باز هم می‌تواند کمتر قابل تفسیر باشد. علاوه بر این، وابسته به مقیاس است.

$$RMSE = \sqrt{\frac{۱}{n}\sum_{i=۱}^{n}(A_i - F_i)^۲}$$

میانگین درصد مطلق خطا

میانگین درصد مطلق خطا میانگین درصد اختلاف بین مقدار واقعی و پیش‌بینی است. این اغلب به عنوان معیار پایه برای اندازه‌گیری بیشتر مدل‌های پیش‌بینی استفاده می‌شود. میانگین درصد مطلق خطا نه

تنها به راحتی قابل تفسیر است، بلکه مستقل از مقیاس است که به ما امکان می‌دهد ارزهای دیجیتال مختلف را با هم مقایسه کنیم. با این وجود، در رمزارزهای با ارزش واقعی نزدیک به صفر، ممکن است خطای بی نهایت داشته باشیم. در این متریک، پیش‌بینی‌های پایین‌تر به خطای ۱۰۰ درصد محدود می‌شوند، اما پیش‌بینی‌های بالاتر می‌تواند به خطای بی‌نهایت افزایش یابد، بنابراین، به پیش‌بینی کمتر سوگیری دارد.

$$MAPE = \frac{۱}{n}\sum_{i=۱}^{n}(۱۰۰ \times \frac{|A_i - F_i|}{A_i})$$

خطای درصد مطلق میانگین متقارن

این یک توسعه میانگین درصد مطلق خطا است. این میانگین ۲۰۰ برابر اختلاف بین مقدار واقعی و پیش‌بینی تقسیم بر مجموع مقادیر مطلق آنها است. این دیگر به نفع پیش‌بینی‌های پایین‌تر نیست. اکنون کاملاً بین ۰% و ۱۰۰% محدود شده است. از آنجایی که مخرج می‌تواند هنوز در حدود صفر باشد، هنوز احتمال مقادیر بی نهایت وجود دارد. علاوه بر این، تفسیر یک متریک بین ۰% و ۱۰۰% ممکن است دشوار باشد. یکی از مشکلات احتمالی خطای درصد مطلق میانگین متقارن این است که متقارن نیست زیرا پیش‌بینی‌های بیش از حد و کمتر به طور یکسان در نظر گرفته نمی‌شوند. این نکته با مثال زیر با استفاده از فرمول خطای درصد مطلق میانگین متقارن نشان داده شده است:

- پیش‌بینی بالاتر از قیمت واقعی: $A_t = ۱۰۰$ و $F_t = ۱۱۰$ نتیجه می‌دهد $SMAPE = ۹/۰۹٪$
- پیش‌بینی کمتر از قیمت واقعی: $A_t = ۱۰۰$ و $F_t = ۹۰$ نتیجه می‌دهد $SMAPE = ۱۰/۵۲٪$

$$SMAPE = \frac{۱}{n}\sum_{i=۱}^{n}(۲۰۰ \times \frac{|A_i - F_i|}{|A_i| + |F_i|})$$

میانگین خطای مقیاس مطلق دو حالت دارد. در حالتی که فصلی بودن را در نظر نگیریم:

$$MASE = \frac{\frac{۱}{n}\sum_{i=۱}^{n} A_i - F_i}{\frac{۱}{T-۱}\sum_{t=۲}^{T} A_t - A_{t-۱}} = \frac{MAE}{\frac{۱}{T-۱}\sum_{t=۲}^{T} A_t - A_{t-۱}}$$

در حالتی که فصلی بودن را در فرمول اعمال کنیم.

$$MASE = \frac{\frac{۱}{n}\sum_{i=۱}^{n} A_i - F_i}{\frac{۱}{T-m}\sum_{t=m+۱}^{T} A_t - A_{t-m}} = \frac{MAE}{\frac{۱}{T-m}\sum_{t=m+۱}^{T} A_t - A_{t-m}}$$

این متریک میانگین خطای مطلق مقیاس برای سری‌های زمانی فصلی و غیرفصلی است و احتمالاً بهترین و منصفانه‌ترین معیار برای استفاده است. این متریک خروجی را با پیش‌بینی ساده مقایسه می‌کند.

پیش‌بینی‌های ساده مقرون‌به‌صرفه‌ترین مدل پیش‌بینی هستند و معیاری را ارائه می‌دهند که می‌توان با آن مدل‌های پیچیده‌تر را مقایسه کرد. این روش پیش‌بینی فقط برای داده‌های سری زمانی مناسب است. با استفاده از رویکرد ساده لوحانه، پیش‌بینی‌هایی تولید می‌شود که برابر با آخرین مقدار مشاهده شده است. این روش برای سری‌های زمانی اقتصادی و مالی، که اغلب دارای الگوهایی هستند که پیش‌بینی دقیق و قابل اعتماد آن‌ها دشوار است، بسیار خوب عمل می‌کند. اگر اعتقاد بر این است که سری زمانی فصلی دارد، رویکرد ساده‌لوحانه فصلی ممکن است در جایی که پیش‌بینی‌ها برابر با مقدار فصل گذشته باشد، مناسب‌تر باشد.

در نماد سری زمانی: $\hat{y}_{T+h|T} = y_t$

در میانگین خطای مقیاس مطلق اگر خطا کمتر از یک باشد، می‌توان نتیجه گرفت که پیش‌بینی بهتر از یک پیش‌بینی ساده متوسط است. برعکس، اگر بیش از یک مورد وجود داشته باشد، پیش‌بینی بدتر از یک پیش‌بینی ساده و متوسط است. مزایای این معیار، استقلال مقیاس و جریمه کردن مقدار زیر پیش‌بینی و بیش از پیش‌بینی به طور مساوی است.

میانگین مربعات خطای لگاریتمی این روش نسبت یا تفاوت نسبی بین مقادیر واقعی و پیش‌بینی شده را با محاسبه میانگین مجذور لگاریتم مقدار واقعی به اضافه یک تقسیم بر مقدار پیش‌بینی شده به اضافه یک اندازه‌گیری می‌کند. این سنجه، پیش‌بینی‌های کمتر را بیشتر از پیش‌بینی بیش‌ازحد مجازات می‌کند. با این حال، تفسیر آن آسان نیست.

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(log(\frac{A_i+1}{F_i+1}))^2$$

فصل ۳

کارهای پیشین

در این فصل مدل‌های پیاده‌سازی شده در کتاب‌خانه که در فصل ۵ مورد آزمایش فرار می‌گیرند را معرفی می‌کنیم و ریاضیات هر مدل را به دقت مورد بررسی قرار می‌دهیم.

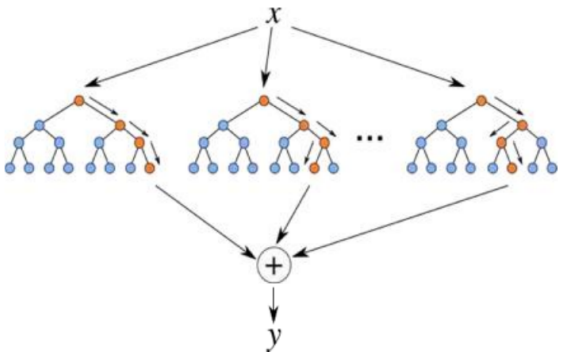
۳ـ۱ جنگل‌های تصمیم تصادفی

مسئله‌ی **جنگل‌های تصمیم تصادفی** [۱] یک تکنیک یادگیری ماشین است که از **درخت‌های تصمیم** [۲] استفاده می‌کند تا یک مدل پیشبینی ایجاد کند[۲۸]. درخت‌های تصمیم همچنین در پیشبینی قیمت بیتکوین هم استفاده شده اند [۲۹] . در ۲۰۱۶ Khaidem و بقیه برای پیشبینی روند ارزش سهام شرکت‌هایی مثل اپل، سامسونگ، الکترونیکس و جنرال الکتریک در بازار نزدک[۳] از درخت‌های تصمیم استفاده کرده اند.

به طور خاص جنگل‌های تصمیم تصادفی یک تکنیک مجموعه ای [۴] است که از درخت‌های تصمیم و بسته بندی [۵] تشکیل شده است [۳۰] . این باعث می‌شود نمونه داده‌های گوناگون برای آموزش هر درخت برای یک مسئله یکسان استفاده شوند. درختان مختلف هنگام استفاده از بسته بندی، بخش‌های متمایز داده را مشاهده می‌کنند. هیچ درختی تمام داده‌های آموزشی را نمی‌بیند که سبب می‌شود تا در هنگام ادغام نتایج، برخی از بی دقتی‌های درخت‌ها جبران شود و پیشبینی قابل تعمیم تری داشته باشیم. [۳۱]

بیش‌برازش [۶] یکی از مشکلات اصلی است که می‌تواند اثرات مخربی بر نتایج مدل داشته باشد. با

[۱] Random Forest
[۲] Decision Trees
[۳] NASDAQ
[۴] ensemble method
[۵] bagging
[۶] Overfitting



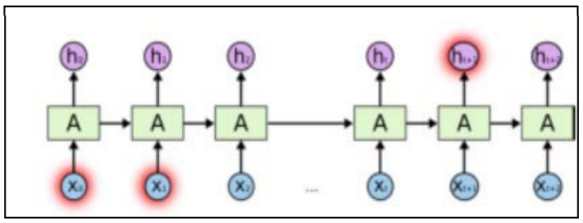
شکل ۳ـ۱: ساختار یک جنگل تصمیم تصادفی

این وجود، اگر درختان تصمیم تصادفی به اندازه کافی وجود داشته باشد، مشکل بیش‌برازش رخ نمی‌دهد. همچنین مقادیر از دست رفته را می‌توان توسط جنگل‌های تصمیم تصادفی مدیریت کرد. همانطور که قبلا اشاره شد، جنگل‌های تصمیم تصادفی بر اساس تکنیک متفاوتی به نام بسته بندی می‌باشد که مستلزم اجرای یک بوت استرپ [۷] بر روی داده‌های آزمایش که بر روی داده‌ی ورودی نمونه گیری شده اند و دارای تعداد ثابتی متغیر هستند.

یک نمونه تصادفی از N مورد از مجموعه آموزشی با جایگزینی گرفته می‌شود. این نمونه مجموعه آموزشی برای ساخت درخت i را تشکیل می‌دهد. با توجه به اینکه تعداد متغیرهای ورودی M است، تعداد متغیرهای انتخاب شده برای هر گره m است $(m < M)$ که در طول تولید بلوک ثابت می‌ماند. سپس بلوک با تقسیم آن و استفاده از تقسیم بهینه خواص m با گره مرتبط می‌شود. تعداد پیش‌بینی‌کننده‌های ارزیابی‌شده در هر بخش تقریباً برابر است با جذر تعداد کل پیش‌بینی‌کننده‌ها. موثرترین روش برای تعیین مقدار مناسب آنالیز خطای میانگین مربعات خارج از کیسه [۸] برای مقادیر مختلف m است. به طور کلی، اگر متغیرهای انتخاب شده در هر گره بسیار به هم متصل باشند، مقادیر کوچک m منجر به نتایج مطلوب می‌شود. در نتیجه، در هر گره، m مشاهده برای آموزش و M-m مشاهده برای آزمایش استفاده می‌شود. به صورت کلی، با توجه به مجموعه آموزشی D با اندازه n می‌توان m مجموعه آموزشی جدید $D_{i=1 \dots m}$ با اندازه n′ از m نمونه با جایگزینی ایجاد کرد. پس از آن، هر درخت تصمیم با استفاده از مجموعه داده D_i آموزش داده می‌شود. علاوه بر این، همانطور که در شکل ۳ـ۱ نشان داده شده است، به گره‌های هر درخت وارد می‌شود تا زمانی که مشاهده جدیدی داشته باشد، پیش‌بینی کند.

---

[۷] Bootstrap
[۸] out-ot-bag



شکل ۳ـ۲: ساختار گسترش یافته یک شبکه عصبی بازگشتی

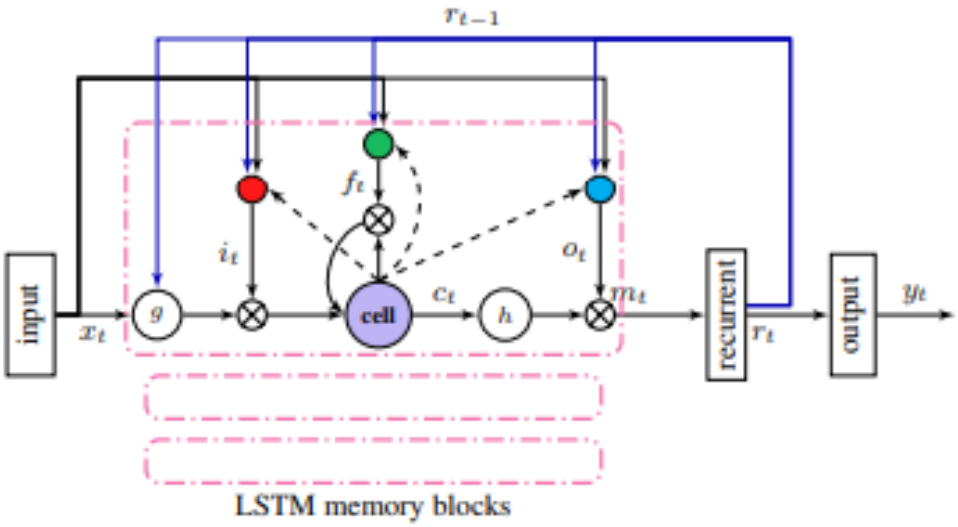
۳ـ۲ حافظه طولانی کوتاه مدت

حافظه طولانی کوتاه مدت [۹] شکل دیگری از شبکه عصبی بازگشتی [۱۰] می‌باشد. حافظه طولانی کوتاه مدت توسط Hochreiter و Schmidhuber در سال ۱۹۹۷ طراحی شد [۳۲]. این مدل بعدها توسط چندین محقق توسعه پیدا کرد و رایج شد. این شبکه از ماژول‌هایی با سازگاری بازگشتی مشابه شبکه‌های عصبی بازگشتی تشکیل شده است. تمایز بین حافظه طولانی کوتاه مدت و شبکه بازگشتی عصبی در ارتباط بین لایه‌های پنهان شبکه بازگشتی عصبی می‌باشد. ساختار شبکه عصبی بازگشتی در شکل ۳ـ۲ نشان داده شده است. تنها تمایز این دو مدل در حافظه‌ی سلول لایه پنهان می‌باشد. همچنین طراحی سه دروازه منحصر به فرد به طور موثر مشکل محو شدن گرادیان را حل می‌کند. شکل ۳ـ۳ ساختار حافظه طولانی کوتاه مدت را نشان می‌دهد [۳۳] [۳۴] .

شکل ؟؟ توضیح می‌دهد که شبکه عصبی بازگشتی دارای نقص‌هایی است که ممکن است در ورودی مشاهده شود. این مشکل توسط Bengio و همکاران در ۱۹۹۴ کشف شد. [۳۵]. X_0، X_1 دارای گستره بسیار وسیعی از اطلاعات X_t، X_{t+1} هستند، به طوری که وقتی h_{t+1} به اطلاعات نیاز دارد، اطلاعات مربوط به X_0، X_1 در شبکه عصبی بازگشتی قادر به یادگیری پیوند دادن اطلاعات نیست. زیرا حافظه قدیمی که ذخیره می‌شود با گذشت زمان به طور فزاینده ای بی فایده می‌شود با توجه به اینکه حافظه جدید بازنویسی می‌شود یا جایگزین می‌شود.

همانطور که در شکل ۳ـ۳ نشان داده شده است، واحدهای ویژه حافظه طولانی کوتاه مدت (لایه‌های پنهان مکرر) حاوی بلوک‌های حافظه هستند. علاوه بر سلول‌های حافظه با اتصالات خود که وضعیت زمانی شبکه را ذخیره می‌کنند، بلوک‌های حافظه همچنین دارای واحدهای ضربی به نام دروازه هستند که جریان اطلاعات را تنظیم می‌کنند. در معماری اصلی، هر بلوک حافظه شامل یک دروازه ورودی و یک دروازه خروجی بود. کنترل جریان فعال سازی ورودی به سلول حافظه، دروازه ورودی است. دروازه خروجی جریان فعال سازی سلول از سلول به بقیه شبکه را تنظیم می‌کند. متعاقباً، بلوک حافظه دروازه فراموشی

[۹] Long Short-Term Memory(LSTM)
[۱۰] Recurrent Neural Network(RNN)

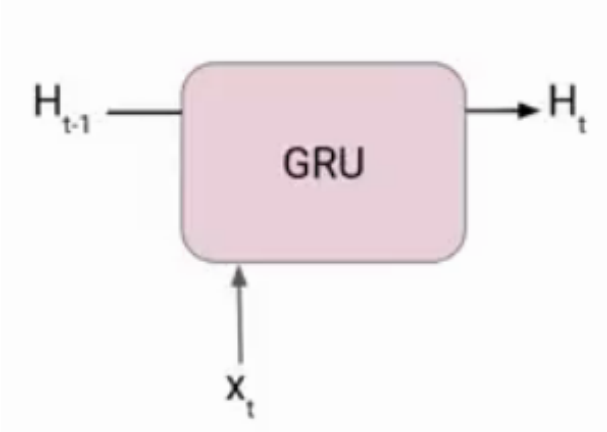


شکل ۳ـ۳: معماری حافظه طولانی کوتاه مدت: یک بلوک حافظه واحد برای وضوح نشان داده شده است.

[۳۶] را دریافت کرد. این نقص مدل‌های حافظه طولانی کوتاه مدت را برطرف می‌کرد که مانع از پردازش جریان‌های ورودی پیوسته که به دنباله‌های بعدی تقسیم نمی‌شدند، می‌شد. دروازه فراموشی، وضعیت داخلی سلول را قبل از افزودن آن به عنوان ورودی به سلول از طریق پیوند خود بازگشتی، مقیاس می‌کند، در نتیجه حافظه سلول را به روشی تطبیقی فراموش یا تنظیم مجدد می‌کند. علاوه بر این، معماری حافظه طولانی کوتاه مدت معاصر شامل اتصالات چشمی از سلول‌های داخلی خود به دروازه‌های همان سلول است تا زمان‌بندی دقیق خروجی را یاد بگیرد [۳۷].

یک نگاشت از یک دنباله ورودی $x = (X_0, X_1, ..., X_t)$ به دنباله خروجی $y = (Y_0, Y_1, ..., Y_t)$ توسط یک حافظه طولانی کوتاه مدت به وسیله محاسبه واحد فعال‌سازی شبکه به صورت تکرارپذیر برای i=۱ تا t به دست می‌آید.

در این فرمول‌ها σ ، i ، f ، o و c به ترتیب تابع سیگموئید لجستیک، دروازه ورودی، دروازه فراموشی، دروازه خروجی و بردار فعال سازی سلول هستند که همگی به اندازه بردار فعال سازی خروجی سلول هستند. متر W به عنوان ماتریس وزن نشان داده می‌شود (به عنوان مثال W_{ix} ماتریس وزن‌ها از دروازه ورودی به ورودی است، W_{ic} ، W_{fc} ، W_{pc} ماتریس‌های وزن مورب هستند. برای اتصالات چشمی)، عبارت b بردارهای بایاس را نشان می‌دهد (به عنوان مثال b_i بردار بایاس دروازه ورودی است). علاوه بر این، g و h به ترتیب توابع فعال‌سازی ورودی و خروجی سلول و tanh و توابع فعال‌سازی خروجی شبکه هستند و softmax وجود دارد.



شکل ۳ـ۴: معماری واحد بازگشتی دروازه‌ای

۳ـ۳ واحد بازگشتی دروازه‌ای

واحد بازگشتی دروازه‌ای [۱۱] ارائه شد [۳۸] تا هر واحد تکراری بتواند وابستگی‌های تطبیق‌پذیر را در چندین مقیاس زمانی ثبت کند. مشابه حافظه طولانی کوتاه مدت، واحد بازگشتی دروازه‌ای دارای واحدهای دروازه‌ای است که بر جریان اطلاعات در داخل واحد تاثیر می‌گذارد، اما بدون سلول‌های حافظه.

می‌توان معماری واحد بازگشتی دروازه‌ای را در شکل ۳ـ۴ مشاهده کرد.

مدل در هر برچسب زمانی t ، یک ورودی X_t و حالت پنهان برچسب زمانی قبلی H_{t-1} دریافت می‌کند. متعاقباً، یک حالت مخفی جدید H_t را خروجی می‌دهد، که در نتیجه به مهر زمانی بعدی ارسال می‌شود. در حال حاضر یک سلول واحد بازگشتی دروازه‌ای از ۲ سلول تشکیل شده است.(بر خلاف حافظه طولانی کوتاه مدت که از ۳ سلول تشکیل شده است.) یکی از دروازه‌های اولیه تنظیم مجدد و دیگری به روز رسانی می‌باشد [۳۹].

دروازه تنظیم مجدد (حافظه کوتاه مدت)

دروازه تنظیم مجدد [۱۲] مسئول حافظه کوتاه مدت شبکه است که که همان حالت پنهان H_t است.

$$r_t = \sigma(x_t * U_r + H_{t-1} * W_r)$$

این معادله مشابه معادله دروازه حافظه طولانی کوتاه مدت است. تابع سیگموید r_t را به ضریبی بین ۰ و ۱ از ماتریس‌های U_r و W_r محدود می‌کند.

دروازه به روز رسانی (حافظه بلند مدت)

[۱۱] Gated Recurrent Unit(GRU)
[۱۲] Reset Gate

به طور مشابه، ما یک دروازه به روز رسانی [۱۳] برای حافظه بلند مدت داریم و معادله دروازه در زیر نشان داده شده است.

$$u_t = \sigma(x_t * U_u + H_{t-۱} * W_u)$$

تنها تفاوت در وزن ماتریس‌ها است که همان U_u و W_u هستند.

دروازه حافظه فعلی

معمولا این دروازه در حین توضیح واحد بازگشتی دروازه‌ای نادیده گرفته می‌شود. یک جزء از دروازه تنظیم مجدد است، همان طور که دروازه مدولاسیون ورودی زیرمجموعه دروازه ورودی است و برای معرفی غیرخطی بودن ورودی و صفر کردن میانگین آن استفاده می‌شود. این دروازه برای معرفی غیرخطی بودن ورودی و صفر کردن میانگین آن استفاده می‌شود. دلیل دیگری گنجاندن آن به عنوان بخش فرعی دروازه تنظیم مجدد، کاهش تاثیر اطلاعات گذشته بر اطلاعاتی است که به آینده منتقل می‌شود.معادله آن به شرح زیر می‌باشد.

$$\hat{H}_t = tanh(x_t * U_g + (r_t \circ H_{t-۱}) * W_g)$$

۳ـ۴ اوربیت

اوربیت طراحی شده توسط شرکت اوبر یک بسته منبع باز است که برای سهولت استنتاج و پیش‌بینی سری‌های زمانی با استفاده از مدل‌های سری زمانی ساختاری بیزی [۱۴] برای برنامه‌های کاربردی دنیای واقعی و مطالعات علمی طراحی شده است [۴۰]. این بسته از زبان‌های برنامه‌نویسی احتمالی مانند Stan [۴۱] و Pyro استفاده می‌کند. [۴۲] در حالی که یک رابط آشنا و شهودی برای بارهای کاری [۱۵] سری‌های زمانی ارائه می‌دهد.

اوربیت مجموعه‌ای از مدل‌های هموارسازی نمایی بیزی اصلاح‌شده [۱۶] با طیف وسیعی از اولویت‌ها، مشخصات نوع مدل و گزینه‌های توزیع نویز را معرفی می‌کند. این مدل شامل یک روند سراسری جدید است که برای سری‌های زمانی کوتاه‌مدت مؤثر است. مهمتر از همه، این شامل یک نرم افزار/بسته محاسباتی پایتون به نام Orbit (سری زمانی بیزی شی گرا [۱۷]) است. فرآیند نمونه برداری و بهینه سازی زیربنایی با

[۱۳] Update Gate
[۱۴] structural Bayesian time series models
[۱۵] workload
[۱۶] refined Bayesian exponential smoothing models
[۱۷] Object-oriented Bayesian Time Series

استفاده از زبان‌های برنامه نویسی احتمالی Stan و Pryo انجام می‌شود. Pyro که توسط محققان اوبر ایجاد شده است، یک زبان برنامه نویسی احتمالی جهانی (PPL) است که در پایتون ساخته شده است و توسط PyTorch و JAX پشتیبان آن است. اوربیت در حال حاضر دارای زیرمجموعه‌ای از الگوریتم‌های پیش‌بینی و نمونه‌برداری موجود برای تخمین Pyro است.

۳ـ۵ آریما

روش میانگین متحرک یکپارچه اتورگرسیو (آریما) [۱۸] در سال ۱۹۷۰ توسط جورج باکس و گویلین جنکینز توسعه یافت [۴۳]. روش آریما به طور کامل متغیرهای مستقل را در حین پیش‌بینی نادیده می‌گیرد، و آن را برای داده‌های آماری به هم پیوسته (وابسته) مناسب می‌سازد و به برخی مفروضات مانند همبستگی خودکار [۱۹]، روند یا فصلی بودن داده نیاز دارد. روش آریما می‌تواند داده‌های تاریخی را با تأثیر داده‌های غیرقابل درک پیش‌بینی کند، از دقت بالایی در پیش‌بینی کوتاه‌مدت برخوردار است و می‌تواند با تغییرات داده‌های فصلی مقابله کند. روش آریما به چهار دسته خود رگرسیون، میانگین متحرک، میانگین متحرک خود رگرسیون و میانگین متحرک یکپارچه اتورگرسیو (ARIMA) طبقه‌بندی می‌شود [۴۴][۴۵].

۱. **خود رگرسیون**

این مدل توسط Yule در سال ۱۹۲۶ معرفی شد و توسط واکر در سال ۱۹۳۲ گسترش یافت. این مدل فرض می‌کند که داده‌های دوره‌های قبل در حال حاضر بر داده‌های فعلی تأثیر می‌گذارند. از آنجایی که در این مدل بر اساس مقادیر قبلی متغیر بازسازی شده است، خود رگرسیون نامیده می‌شود. روش خود رگرسیو برای محاسبه مقدار ترتیب ضریب p استفاده می‌شود که نشان دهنده وابستگی یک مقدار به نزدیکترین مقدار قبلی آن است [۴۶].

$$X_t = \mu + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + e_t$$

۲. **میانگین متحرک**

برای اولین بار توسط Slutzky در سال ۱۹۷۳ ارائه شد. رویکرد میانگین متحرک برای محاسبه ضریب ترتیب q استفاده می‌شود، که برای حرکت متغیر مقدار باقیمانده قبلی استفاده می‌شود [۴۷]. شکل زیر فرم عمومی مدل میانگین عمومی با مرتبه $q(MA(q))$ یا مدل آریما $(\circ\circ q)$ است.

$$X_t = e_t - \phi_1 e_{t-1} - \phi_2 e_{t-2} - \ldots - \phi_q X_{t-q}$$

[۱۸] The Autoregressive Integrated Moving Average (ARIMA)
[۱۹] autocorrelation

۳. **میانگین متحرک خود رگرسیون (آرما)**

این مدل مدل‌های خود رگرسیون و میانگین متحرک را ترکیب می‌کند. فرض کنید که داده‌های دوره جاری تحت تأثیر داده‌های دوره قبل و ارزش اجباری دوره قبل است. در زیر اشکال رایج مدل‌های فرآیند خود رگرسیون و میانگین متحرک یا آرما هستند $(p, ۰, q)$.

$$X_t = \mu + \phi_۱ X_{t-۱} + \phi_۲ X_{t-۲} + ... + \phi_p X_{t-p} + e_t -$$
$$\phi_۱ e_{t-۱} - \phi_۲ e_{t-۲} - ... - \phi_q X_{t-q}$$

۴. **میانگین متحرک یکپارچه اتورگرسیو (آریما)**

مدل آریما فرض می‌کند که داده‌های مورد استفاده باید ثابت باشند، به این معنی که میانگین تغییرات داده‌ها ثابت است. داده‌های غیر ثابت ابتدا باید با استفاده از روش تفاضل به داده‌های ثابت تبدیل شوند. تکنیک آریما یک روش دیدگاه آماری است که با سه پارامتر نشان داده می‌شود، که اولین مورد فرآیند خود رگرسیون داده دوره قبل است که بعداً در فرآیند یکپارچه گرفته شده و حفظ می‌شود و پیش‌بینی داده‌ها را آسان‌تر می‌کند. نسخه رایج مدل آریما (p, d, q) به شرح زیر است:

$$X_t = \mu + X_{t-۱} + X_{t-d} + ... + \phi_۱ X_{t-۱} + \phi_۲ X_{t-۲} + ... +$$
$$\phi_p X_{t-p} + e_t - \phi_۱ e_{t-۱} - \phi_۲ e_{t-۲} - ... - \phi_q e_{t-q}$$

۳ـ۶ ساریمکس

آرما ترکیبی از مدل‌های خود رگرسیون و میانگین متحرک است. همان طور که در بخش آریما گفته شد افزودن یک اپراتور یکپارچه سازی به یک مدل آرما یک مدل آرما تولید می‌کند. یک مدل ساریمکس شامل متغیرهای برون زا ارزیابی شده در زمان t است که بر مقدار داده‌های ورودی در زمان t و ضرب کننده‌های عدد صحیح فصلی تأثیر می‌گذارد [۴۸]. پارامترهای مورد نیاز برای تعریف مدل ساریمکس در ۳ـ۵ آمده است.

فرمول مدل خود رگرسیون AR(p) به شرح زیر است:

$$y_t = \Theta(L)^p * y_t + \epsilon_t$$

مدل میانگین متحرک MD(d) به شکل زیر نمایش داده می‌شود:

$$y_t = \phi(L)^q * \epsilon_t + \epsilon_t$$

Symbol	Remark	Symbol	Remark
p	Number of time lags to regress on.	θ_i	Parameters of Moving Average.
β	Constant (Measured as deviations from its mean).	$\Theta(L)^p$	An order p polynomial function of L.
θ_i	Parameters of Moving Average.	d	Order of differencing used.
L	Lag Operator	y_t	Prediction Value
$\phi(L)^q$	An order q polynomial function of L.	n	Number of exogenous variables.
Δ^d	Integration Operator.	β_n	Coefficients of exogenous variables.
x_t^i	Exogenous variables defined at each time step t. For in	Δ_s^D	Differencing operator.
q	Number of time lags of the error term to regress on	$\phi(L^s)^Q$	An order Q polynomial function with seasonality of L.
ϵ_t	Gaussian White noise at time t. (Zero mean)		

شکل ۳ـ۵: فهرست نمادها و پارامترهای مدل ساریمکس

مدل میانگین متحرک خودرگرسیون ARMA(p،q) را می‌توان به صورت زیر نوشت:

$$y_t = \Theta(L)^p * y_t + \phi(L)^q * \epsilon_t + \epsilon_t$$

و مدل میانگین متحرک یکپارچه اتورگرسیو ARIMA(p،d،q) به صورت زیر بیان می‌شود:

$$y_t^{[d]} = \Delta^d * y_t = y_t^{[d-1]} - y_{t-1}^{[d-1]}$$

$$\Delta^d * y_t = \Theta(L)^p * \Delta^d * y_t + \phi(L)^q \Delta^d * \epsilon_t + \Delta^d * \epsilon_t$$

در نهایت مدل میانگین متحرک تلفیقی خودرگرسیون فصلی با متغیر توضیحی $SARIMAX((p,d,q) * (P,D,Q)$ [۲۰] به صورت زیر بیان می‌شود:

$$\Theta(L)^p * \theta(L^s)^p * \Delta_s^D * \Delta^d * y_t = \phi(L)^q * \phi(L^s)^Q * \Delta^d * \Delta_s^D * \epsilon_t + \sum_{i=1}^{n} \beta_i * x_t^i$$

۳ـ۷ پرافت

پرافت [۲۱] روشی برای پیش‌بینی داده‌های سری زمانی بر اساس یک مدل افزایشی است که در آن روندهای غیرخطی با فصلی سالانه، هفتگی و روزانه، علاوه بر اثرات تعطیلات، برازش داده می‌شوند. زمانی که برای سری‌های زمانی با اثرات فصلی قابل‌توجه و چندین فصل از داده‌های تاریخی استفاده شود، بیشترین تأثیر را دارد. پرافت نسبت به داده‌های از دست رفته و نوسانات در روند مقاوم است و معمولاً به خوبی با اطلاعات پرت برخورد می‌کند [۴۹].

$$p(y_t, y_{t-1}, ..., y_1) =$$

$$\aleph(y_t|m(t), \sigma^2).\aleph(y_{t-1}|m(t-1), \sigma^2)...\aleph(y_1|m(1), \sigma^2)$$

[۲۰] Seasonal Autoregressive Integrated Moving Average Model with Explicative Variable (SARIMAX)

[۲۱] Prophet

تا زمانی که پرافت به طور دقیق میانگین و واریانس شرطی را نشان می‌دهد، باید به نتایج قابل قبولی برسد. از نظر ریاضی این فرمول را داریم:

$$m_{prophet}(t+h) \approx \mathbb{E}[y_{t+h}|y_t, ..., y_1]$$

$$v_{prophet}(t+h) = \sigma^2 \approx \mathbb{V}ar[y_{t+h}|y_t, ..., y_1]$$

$$For\ all\ forecast\ periods\ t+h$$

این می‌تواند در صورتی اتفاق بیفتد که سیستم زیربنایی در شرایط تعادل باشد، مانند زمانی که اقتصاد با ثبات است. بنابراین، هنگامی که یک شوک قابل توجه رخ می‌دهد، معیار واریانس به طور قطع نقض می‌شود. این دقیقاً همان چیزی است که در مثال سری زمانی قبلی مشاهده کردیم.

XGBoost ۸_۳

درختان افزایش گرادیان یادگیری بدون نظارت را انجام می‌دهند که در آن از داده‌ها بدون مدل مشخصی یاد می‌گیرند. XGBoost یک کتابخانه محبوب افزایش گرادیان است. می‌توان از آن برای آموزش مدل با استفاده از واحد پردازش گرافیکی، محاسبات توزیع شده و موازی سازی استفاده کرد. دقیق، قابل انطباق با انواع داده‌ها و موقعیت‌ها، به خوبی مستند و بسیار کاربرپسند است.

XGBoost مخفف عبارت افزایش گرادیان شدید [۲۲] است. این یک نسخه موازی شده و بهینه شده از تکنیک تقویت گرادیان است. موازی کردن کل فرآیند تقویت، زمان تمرین را به شدت کاهش می‌دهد.

آنها به‌جای آموزش بهترین مدل ممکن بر روی داده‌ها (همانطور که در مورد رویکردهای مرسوم است)، صدها مدل را بر روی زیر مجموعه‌های مختلف مجموعه داده آموزشی آموزش دادند و سپس برای تعیین مدل با بهترین عملکرد رأی‌گیری کردند.

در بسیاری از موقعیت‌ها، XGBoost نسبت به روش‌های مرسوم افزایش گرادیان برتری دارد. پیاده سازی Python دسترسی به مجموعه عظیمی از پارامترهای داخلی را فراهم می‌کند که می‌توانند برای بهبود دقت و صحت اصلاح شوند.

موازی سازی، منظم سازی، غیر خطی بودن، اعتبارسنجی متقابل و مقیاس پذیری از ضروری ترین ویژگی‌های XGBoost هستند.

الگوریتم XGBOOST به گونه ای کار می‌کند که یک تابع را در نظر گرفته یا تخمین می‌زند. برای شروع، ما یک دنباله بر اساس گرادیان تابع تولید می‌کنیم. معادله زیر نوع خاصی از نزول گرادیان را مدل

[۲۲] Extreme Gradient Boosting

می‌کند. جهت کاهش تابع را مشخص می‌کند، زیرا تابع ضرر را برای به حداقل رساندن نشان می‌دهد. مربوط به نرخ یادگیری در نزول گرادیان است و نرخ تغییر متناسب با تابع ضرر است. پیش‌بینی می‌شود که رفتار زیان را به اندازه کافی تکرار کند.

$$F_{x_t+1} = F_{x_t} + \epsilon_{x_t} \frac{\partial F}{\partial x}(x_t)$$

برای تکرار روی مدل و تعیین فرمول بهینه آن، باید کل فرمول را به عنوان یک دنباله توصیف کنیم و تابعی را شناسایی کنیم که به کمترین مقدار تابع همگرا شود. این تابع به عنوان یک معیار خطا به ما کمک می‌کند تا خطا را به حداقل برسانیم و عملکرد را در طول زمان حفظ کنیم و همجنین سری به حداقل مقدار تابع نزدیک شود. این نماد خاص نشان دهنده تابع خطایی است که هنگام ارزیابی یک رگرسیون تقویت کننده گرادیان اعمال می‌شود [۵۰].

$$f(x, \theta) = \sum l(F((X_i, \theta), y_i))$$

فصل ۴

روش‌شناسی

در ابتدا به معرفی کتابخانه خودمان می‌پردازیم که پلتفرمی برای پیاده‌سازی تمام آزمایش‌ها در شرایط منصفانه برای ما فراهم می‌کند. سپس به شرح روشی که برای پیش‌بینی تغییر روند استفاده کردیم می‌پردازیم.

۴ـ۱ کتابخانه

به منظور فراهم کردن بستری برای جامعه که در آن مدل‌ها و ارزهای دیجیتال مختلف در دسترس هستند، کتابخانه‌ای به نام CryptoPredictions [۱] طراحی کرده‌ایم. مقالات پیش‌بینی قیمت ارزهای دیجیتال قبلی از معیارهای مختلف و تنظیمات داده استفاده می‌کردند که باعث ایجاد ابهامات و مشکلات تفسیری می‌شد. برای کاهش این تفاوت‌ها، ما CryptoPredictions (کتابخانه ای با ۹ مدل، ۳۰ اندیکاتور و ۱۰ متریک) را ایجاد کردیم. این کتابخانه مزایای زیر را دارد:

۱. در ابتدای کار، ما با چالش جدی کمبود مجموعه داده مواجه شدیم. بسیاری از مقالات و مخازن داده‌ها را از طریق وب سایت‌های مختلف مانند $Yahoo\ Finance$ دریافت کردند. با این حال، ما با استفاده از پلتفرم‌هایی مانند Bitmex که ساختار مشترکی را برای ارزهای مختلف ارائه می‌دهد، بر این مانع غلبه کرده ایم.

۲. قبل از ظهور کتابخانه ما، کاربران مجبور بودند کدهای مختلفی را برای مدل‌های مختلف اجرا کنند که مقایسه عادلانه آنها را دشوار می‌کرد. خوشبختانه، CryptoPredictions امکان انجام یک ارزیابی یکپارچه و عادلانه از مدل‌های مختلف را فراهم کرده است.

[۱] github.com/alimohammadiamirhossein/CryptoPredictions

۳. با هیدرا [۲] کاربران به راحتی می‌توانند آرگومان‌ها را ساختاردهی و درک کنند، و اجرای کدها در تنظیمات مختلف و بررسی نتایج را آسان‌تر می‌کنند. با استفاده از هیدرا کاربران درک بهتری از آرگومان‌ها دارند. علاوه بر این، اجرای یک کد در تنظیمات مختلف و بررسی نتیجه بسیار ساده تر است.

۴. در حالی که برخی از مدل‌ها ممکن است از نظر دقت عملکرد فوق‌العاده‌ای داشته باشند، اما اغلب به یک استراتژی کاملاً تعریف‌شده برای معامله موفق نیاز دارند. بک‌تستر ما می‌تواند به کاربران کمک کند تا اثربخشی مدل مورد استفاده را در سناریوهای دنیای واقعی تعیین کنند.

۵. ما می‌دانیم که ارزیابی مدل‌ها می‌تواند چالش‌برانگیز باشد، به همین دلیل است که معیارهای مختلفی را برای کمک به کاربران برای اندازه‌گیری پیشرفت در انجام وظایفشان ارائه می‌دهیم. با تجزیه و تحلیل معیارهای متعدد، می‌توان زمینه‌های بهبود را شناسایی کرد و مواردی را که کار نمی‌کند اصلاح کرد.

۶. در CryptoPredictions اندیکاتورها را از وب‌سایت‌های مختلف واکشی نمی‌کنیم، زیرا منجر به مشکلاتی مانند ردیف‌های پوچ و کمبود اطلاعات در مورد اندیکاتورها برای ارزهای دیجیتال می‌شود. در عوض، CryptoPredictions آنها را به گونه ای محاسبه می‌کند که مشکلات ذکر شده را نداشته باشد و می‌تواند به مجموعه داده‌های دیگر تعمیم یابد.

۴ـ۲ پیش‌بینی روند بازار

برای پیش‌بینی روند بازار از رگرسیون چندکی همدیس استفاده کردیم. در ابتدای پروژه ما به دنبال روشی برای پیش‌بینی نقاطی بودیم که روند بازار در آن‌ها تغییر می‌کند. برای این کار نیاز به تعریف روند داریم. در [۵۱] برای تعریف روند بازه‌های قیمت را به سطل‌های گوناگون تقسیم کرده است. برای مثال یک بازه ۰ تا ۲ درصد، بازه بعد ۲ تا ۴، دیگری ۴ تا ۶، سپس ۶ تا ۸ و آخرین بازه ۸ تا ۱۰ درصد تغییر است. بدین گونه تغییرات دو روز متوالی طبق میزان تغییری که دارد اندازه‌گیری می‌شود و روندهایی که به صورت شهودی از کند تا تند مرتب می‌شوند. همچنین یک ایده رایج دیگر که برای مثال در [۵۲] دیده می‌شود تعریف روند بدین صورت است که آیا در هر روز قیمت نسبت به روز گذشته افزایش داشته است یا کاهش و تعریف روند صرفا در روند صعودی یا نزولی روزهای متوالی خلاصه می‌شود.

ما به دنبال تعریف جدیدی برای این مسئله بودیم پس ایده‌ی استفاده از رگرسیون چندکی همدیس را

[۲] Hydra

```
# Get scores
cal_scores = np.maximum(cal_labels-model_upper(cal_X), model_lower(cal_X)-cal_labels)
# Get the score quantile
qhat = np.quantile(cal_scores, np.ceil((n+1)*(1-alpha))/n, interpolation='higher')
# Deploy (output=lower and upper adjusted quantiles)
prediction_sets = [val_lower - qhat, val_upper + qhat]
```

شکل ۴ـ۱: کد پایتون رگرسیون چندکی همدیس

انتخاب کردیم. رگرسیون چندکی همدیس به پیش‌بینی هر روز با محاسبه نمره انطباق، یک بازه نسبت می‌دهد. اگر نقطه نزدیک به پایین یا بالای بازه باشد و یا حتی بیرون بازه باشد نشان دهنده این موضوع است که عملکرد مدل در آن روز خوب نبوده است. این عملکرد ضعیف می‌تواند به دو مسئله مربوط باشد.

۱. خطای مدل در آن داده به خصوص

۲. تغییر روند بازار که باعث می‌شود پیش‌بینی ما خارج از بازه‌ی پیش‌بینی شده بیفتد

برای پیاده‌سازی رگرسیون چندکی همدیس، از رگرسیون چندک به عنوان مدل پایه خود استفاده می‌کنیم. به عنوان یادآور الگوریتم رگرسیون چندک تلاش می‌کند تا چندک y مربوط به $Ytest|Xtest = x$ را برای هر x ممکن به دست بیاورد.

اگر به چندک واقعی $t_y(x)$ و به چندک برازش داده‌شده $\hat{t}_y(x)$ بگوییم آنگاه طبق تعریف $Ytest|Xtest = x$ به احتمال $\alpha/۲$ درصد زیر $t_{\alpha/۲}(x)$ و به احتمال $\alpha/۲$ درصد بالای $t_{۱-\alpha/۲}(x)$ قرار می‌گیرد. در نتیجه ما انتظار داریم بازه‌ی $[\hat{t}_{\alpha/۲}(x), \hat{t}_{۱-\alpha/۲}(x)]$ پوشش تقریبا $۱ - \alpha$ درصدی داشته باشد. با این حال، از آنجایی که ممکن است چندک‌های برازش دقیق نباشد، آن‌ها را همدیس می‌کنیم. شبه کد پایتون برای رگرسیون چندک همدیس در شکل ۴ـ۱ آمده است.

پس از آموزش الگوریتمی برای خروجی دو چندک (این کار را می‌توان با یک تابع از دست دادن استاندارد انجام داد، به زیر مراجعه کنید)، $t_{\alpha/۲}(x)$ و $t_{۱-\alpha/۲}(x)$ ، می‌توانیم امتیاز را به عنوان تفاوت بین y و نزدیکترین چندک آن تعریف کنیم.

$$s(x, y) = max(\hat{t}_{\alpha/۲}(x) - y, y - \hat{t}_{۱-\alpha/۲}(x))$$

پس از محاسبه امتیازات در مجموعه کالیبراسیون و قراردادن $\hat{q}$ برابر با $\frac{\lceil (n+۱)(۱-\alpha) \rceil}{n}$ مین چندک مجموعه $\{s_۱, ..., s_n\}$ ، می‌توانیم با گرفتن فواصل پیش‌بینی معتبر ایجاد کنیم.

$$\mathbf{C}(x) = [\hat{t}_{\alpha/۲}(x) - q,\ \hat{t}_{۱-\alpha/۲}(x) + q]$$

آزمایش‌های فصل ۶ نتایج ارزیابی این ایده را نشان می‌دهند.

فصل ۵

نتایج مربوط به عملکرد مدل‌ها

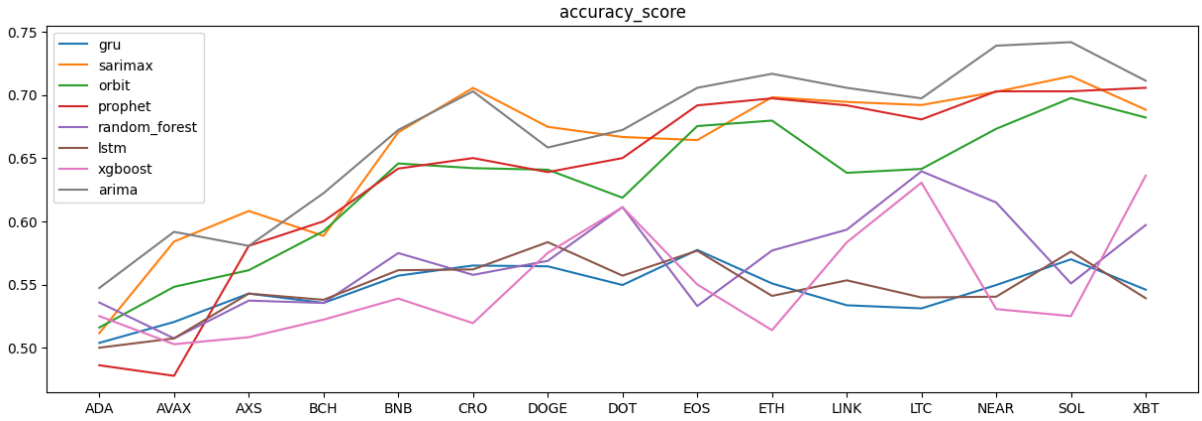
برای داشتن یک ارزیابی منصفانه تصمیم گرفتیم مدل‌های مختلف را در ارزهای دیجیتال و معیارهای مختلف مقایسه کنیم. مجموعه داده آموزشی در همه مدل‌ها برابر است که از : ۱۳ ۱۳ – ۱۱ – ۲۰۲۲ ۰۰ : ۳۰ تا ۰۰ : ۳۰ : ۹ ۰۱ – ۰۱ – ۲۰۲۳ است. همچنین مجموعه داده آزمایشی در همه مدل‌ها نیز برابر است و از ۰۰ : ۳۰ : ۱۰ ۰۱ – ۰۱۲۰۲۳۰۱ تا ۰۰ : ۳۰ : ۱۰ ۱۶ – ۰۲ – ۲۰۲۳ است. ما از داده‌های ساعتی استفاده کردیم و نتایج را در نمودارهای زیر گزارش کردیم.

۵ـ۱ امتیاز دقت و امتیاز F۱

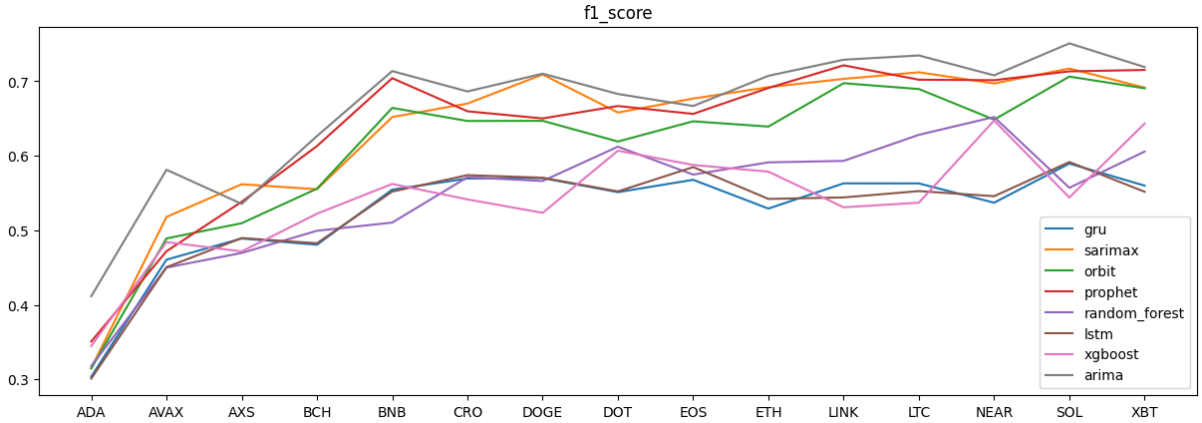
همانطور که در شکل ۵ـ۱ و عکس ۵ـ۲ مشاهده می‌شود، نمودارهای ارزهای دیجیتال مختلف به یکدیگر نزدیک هستند. تقریبا در همه مدل‌ها، آریما و ساریمکس بهترین نتایج را دارند. در وهله دوم، پرافت را داریم که در نزدیکی آریما و ساریمکس نتیجه فوق العاده ای دارد. پس از آن اوربیت را داریم که نتیجه آن به خوبی سه مدل دیگر نیست، اما قابل قبول است. در نهایت XGBoost ، جنگل تصادفی، حافظه طولانی کوتاه مدت و واحد بازگشتی دروازه‌ای را داریم که نتایج نزدیک به هم از نظر دقت حدود ٪۵۵ و از نظر امتیاز F۱ هم حدود ۵.۰ دارند.

۵ـ۲ امتیاز بازیابی و امتیاز دقیق

در شکل ۵ـ۳ & عکس ۵ـ۴ مشهود است که نتایج ساریمکس، آریما، پرافت و اوربیت تفاوت قابل توجهی را با سایرین نشان می‌دهد. همچنین می‌توان این مسئله را مشاهده کرد که در ارزهای ADA ،



شکل ۵ـ۱: امتیاز دقت



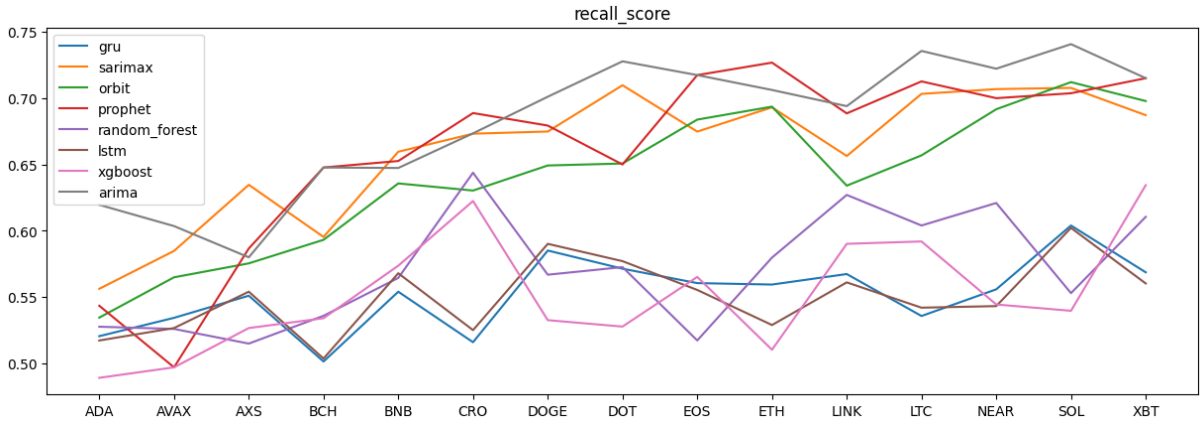
شکل ۵ـ۲: امتیاز F۱

AVAX و AXS امتیاز پایین‌تری را داریم و این موضوع می‌تواند نشان‌دهنده‌ی سخت‌تر بودن پیش‌بینی برای این ارزها نسبت به سایرین باشد.

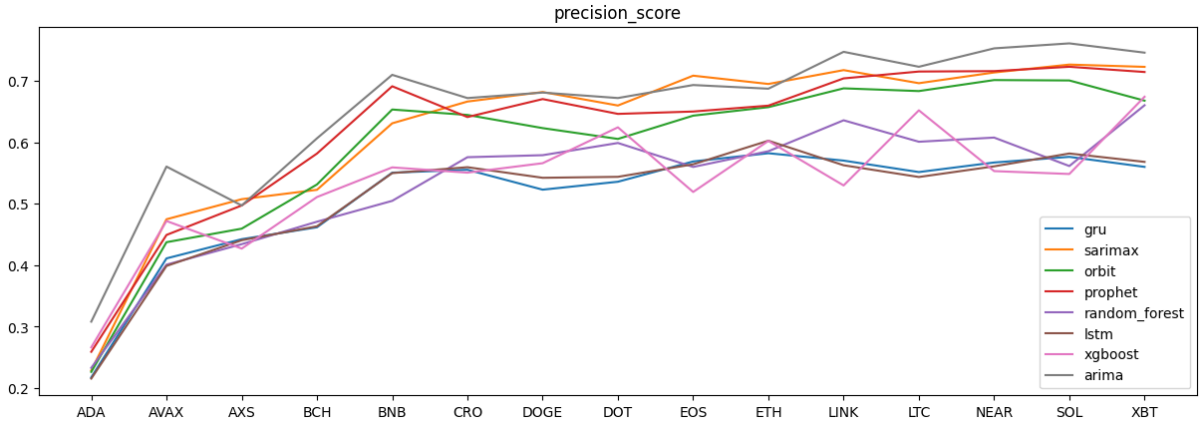
۵ـ۳ سایر متریک‌ها

با این وجود، ما نتایج متفاوتی در رابطه با میانگین درصد خطای مطلق [۱] ، میانگین درصد خطای متقارن [۲] ، میانگین خطای مقیاس مطلق [۳] و میانگین مربع خطای گزارش [۴] داریم. نتایج در شکل ۵ـ۶ & عکس ۵ـ۷ & عکس ۵ـ۸ نشان داده شده است. با وجود نتیجه خیره کننده از نظر دقت و امتیاز F۱، ساریمکس نتیجه

[۱] Mean Absolute Percentage Error(MAPE)
[۲] Symmetric Mean Absolute Percentage Error(SMAPE)
[۳] Mean Absolute Scaled Error(MASE)
[۴] Mean Squared Log Error (MSLE)

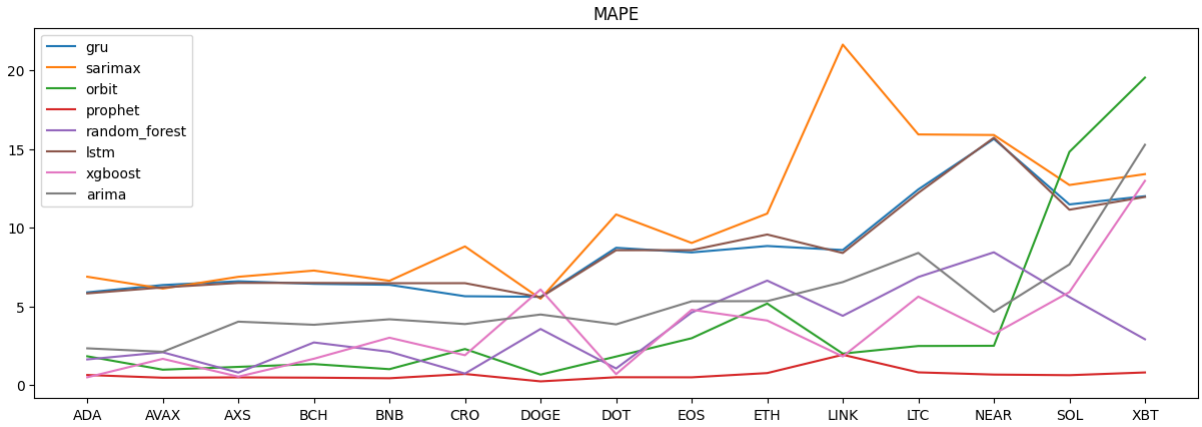


شکل ۵ـ۳: امتیاز بازیابی

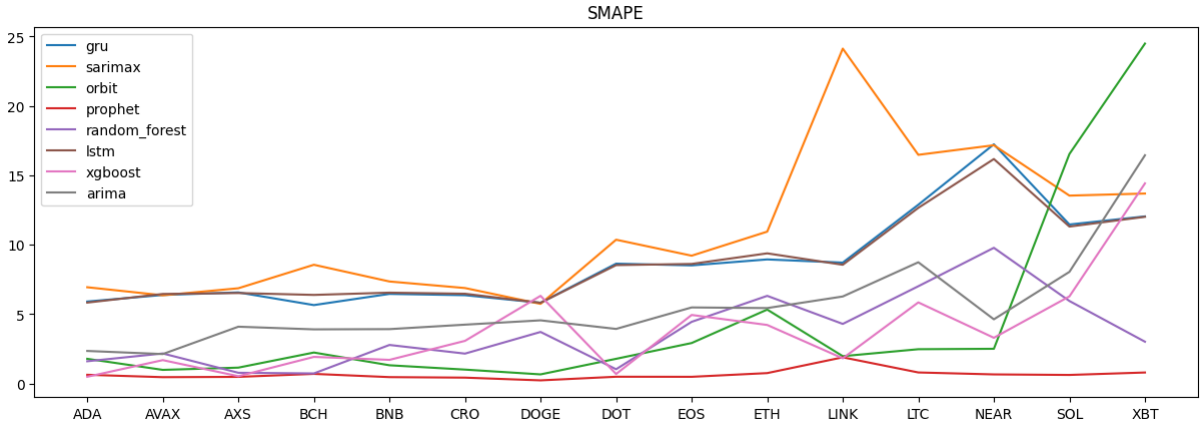


شکل ۵ـ۴: امتیاز دقیق

ضعیفی را نشان می‌دهد. بنابراین ممکن است انتخاب مناسبی برای پیش‌بینی قیمت نباشد. واحد بازگشتی دروازه‌ای و جنگل تصادفی در رتبه دوم این معیارها نتیجه ضعیفی دارند. مشاهده می‌شود که پرافت در تمام این معیارها بهترین نتیجه را نشان می‌دهد. از طرف دیگر نتایج اوربیت، آریما، حافظه طولانی کوتاه مدت و XGBOOST بین ساریمکس و پرافت قرار می‌گیرند.



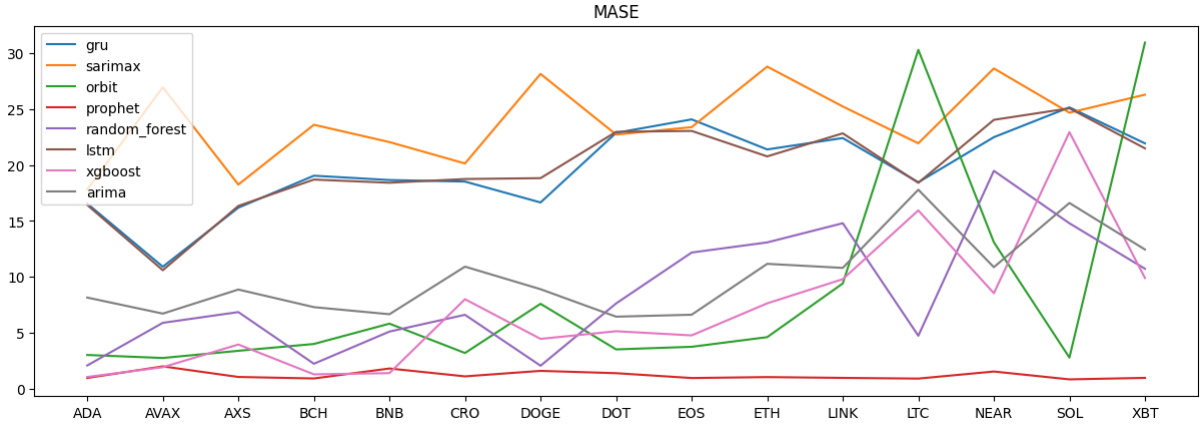
شکل ۵ـ۵: میانگین درصد خطای مطلق



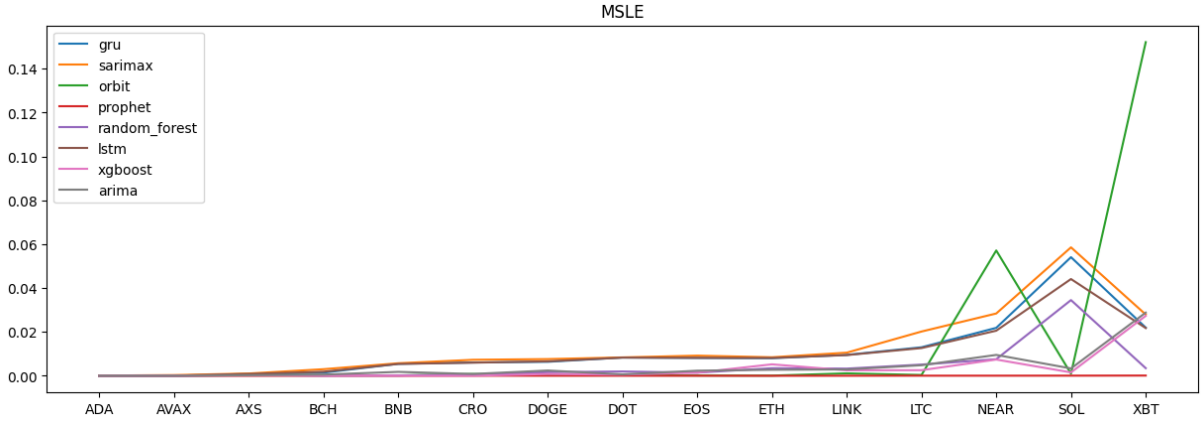
شکل ۵ـ۶: میانگین درصد خطای متقارن

۵ـ۴ نتایج در بیت‌کوین

در این بخش، ما تجربیات قبلی را برای بیت‌کوین انجام می‌دهیم. جزئیات آزمایش‌ها در جدول ۵ـ۱ مشهود است. در مورد بیت‌کوین ما می‌توانیم نتایج مشابهی را ببینیم. جزئیات نتایج در شکل ۵ـ۹ تا شکل ۵ـ۱۷ نشان داده شده است.



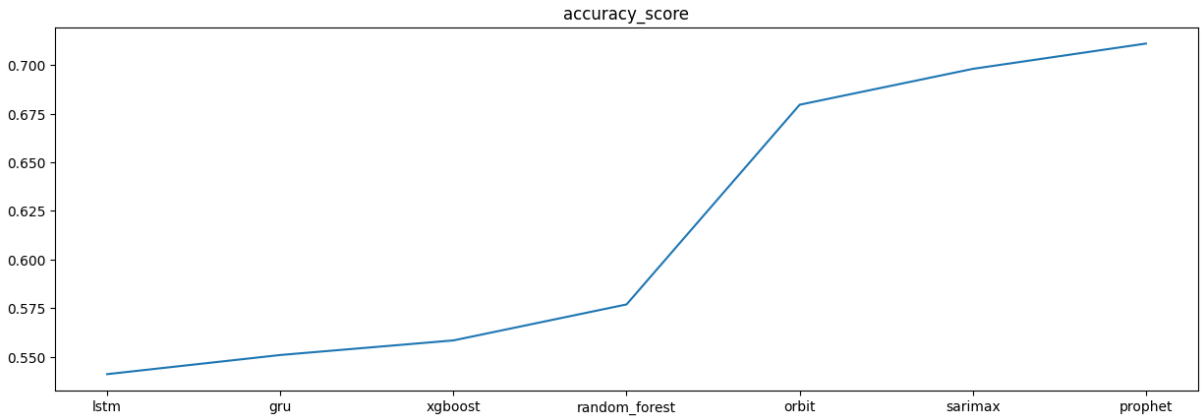
شکل ۵ـ۷: میانگین خطای مقیاس مطلق



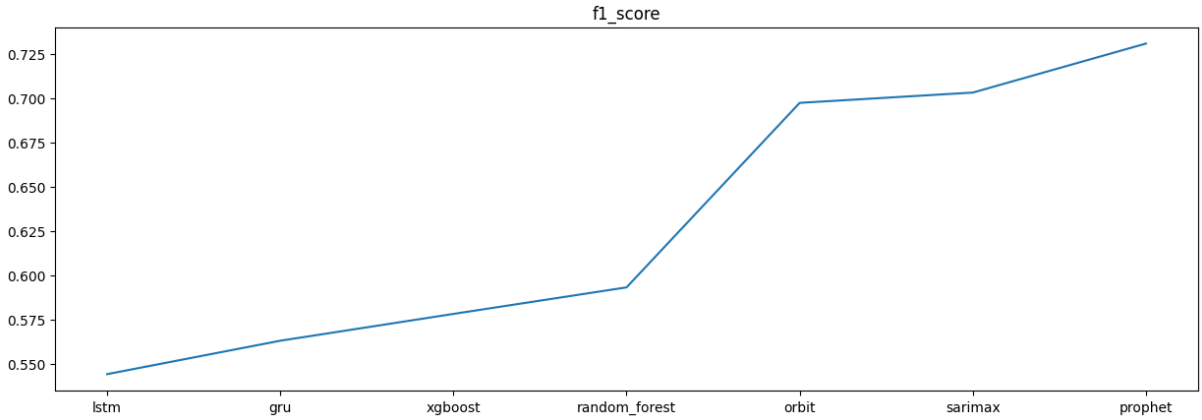
شکل ۵ـ۸: میانگین مربع خطای گزارش

جدول ۵ـ۱: نتایج آزمایش برای بیت‌کوین

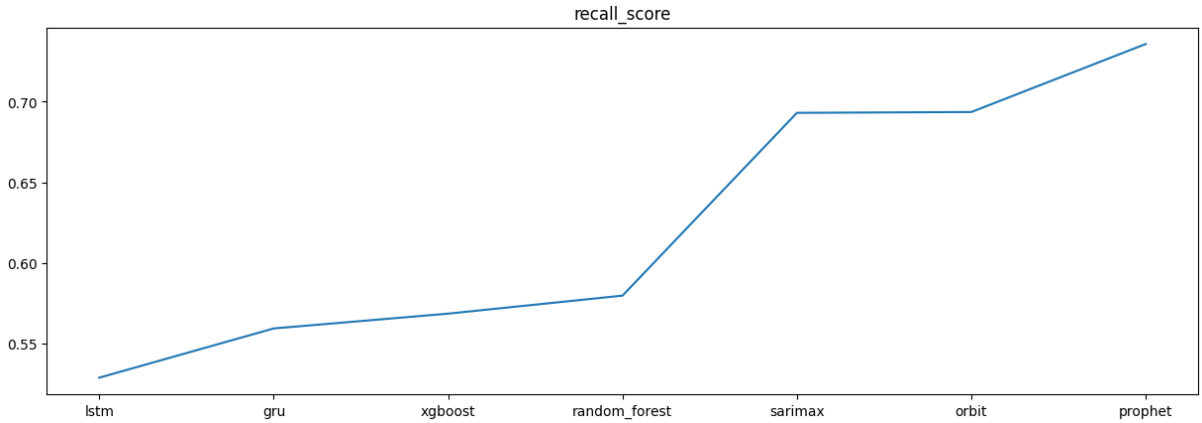
MSLE	MASE	SMAPE	MAPE	RMSE	MAE	Precision	Recall	F۱	Accuracy	
۰/۰۰۰۰۴	۱/۷	۰/۴۰	۰/۴۰	۱۱۶	۸۳	۰/۷۳	۰/۷۴	۰/۷۳	۰/۷۱	پرافت
۰/۰۰۰۱۰	۲/۷	۰/۶۶	۰/۶۵	۱۸۴	۱۳۷	۰/۷۰	۰/۶۹	۰/۷۰	۰/۶۸	اوربیت
۰/۰۰۸۴۶	۲۴/۶	۵/۷۴	۵/۴۸	۱۳۹۱	۱۱۱۹	۰/۷۱	۰/۶۹	۰/۷۰	۰/۷۰	ساریمکس
۰/۰۰۲۷۷	۱۶/۶	۴/۵۵	۴/۴۷	۱۱۶۳	۱۰۰۹	۰/۷۵	۰/۷۱	۰/۷۳	۰/۷۲	آریما
۰/۰۰۳۵۲	۱۵/۱	۳/۷۶	۳/۶۰	۱۱۱۵	۷۹۶	۰/۵۹	۰/۵۷	۰/۵۸	۰/۵۶	XGBOOST
۰/۰۰۸۱۶	۲۵/۱	۵/۸۱	۵/۵۶	۱۴۳۴	۱۱۴۰	۰/۵۶	۰/۵۳	۰/۵۴	۰/۵۴	حافظه طولانی کوتاه مدت
۰/۰۰۸۰۴	۲۵/۲	۵/۸۲	۵/۶۰	۱۴۲۴	۱۱۴۰	۰/۵۷	۰/۵۶	۰/۵۶	۰/۵۵	واحد بازگشتی دروازه‌ای
۰/۰۰۳۴۷	۱۴/۸	۳/۷۱	۳/۵۵	۱۰۹۸	۷۸۷	۰/۶۱	۰/۵۸	۰/۵۹	۰/۵۸	جنگل تصادفی



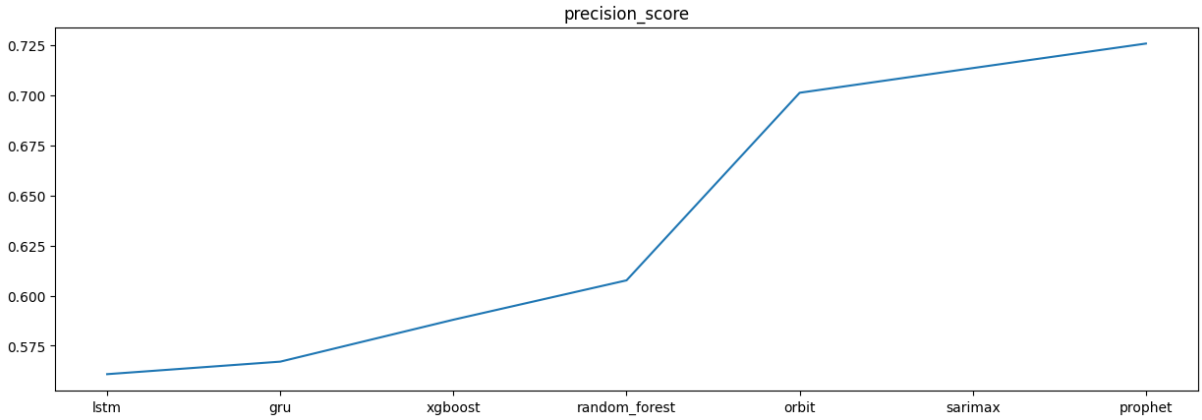
شکل ۵ـ۹: خطای دقت



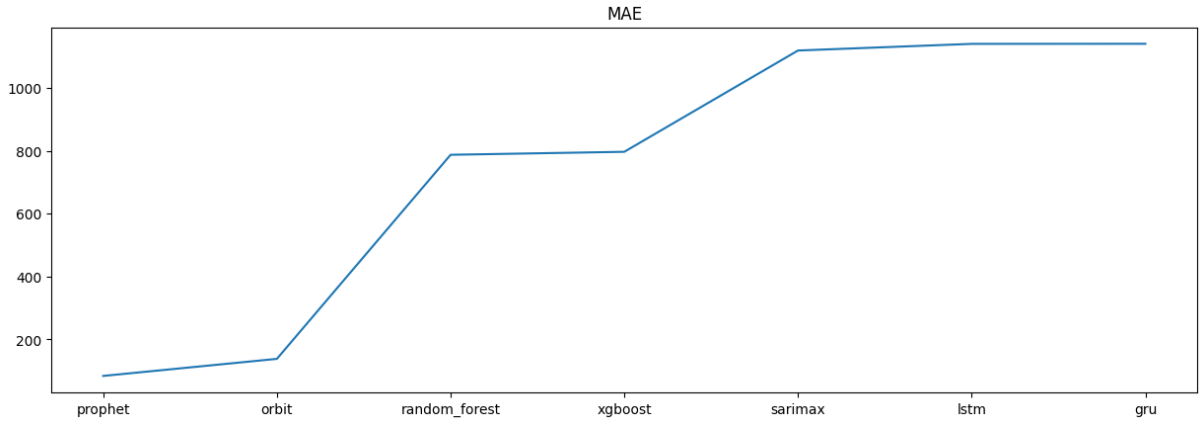
شکل ۵ـ۱۰: امتیاز F۱



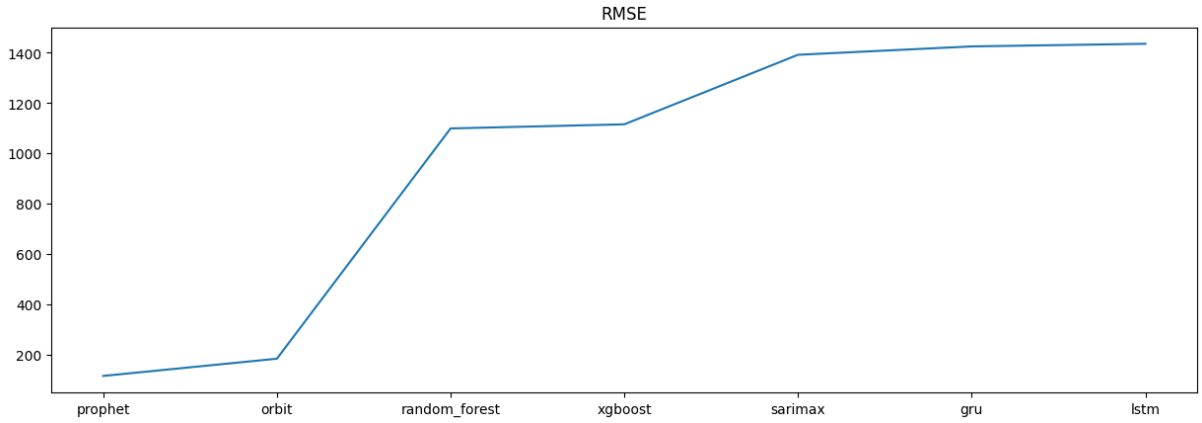
شکل ۵ـ۱۱: امتیاز بازیابی



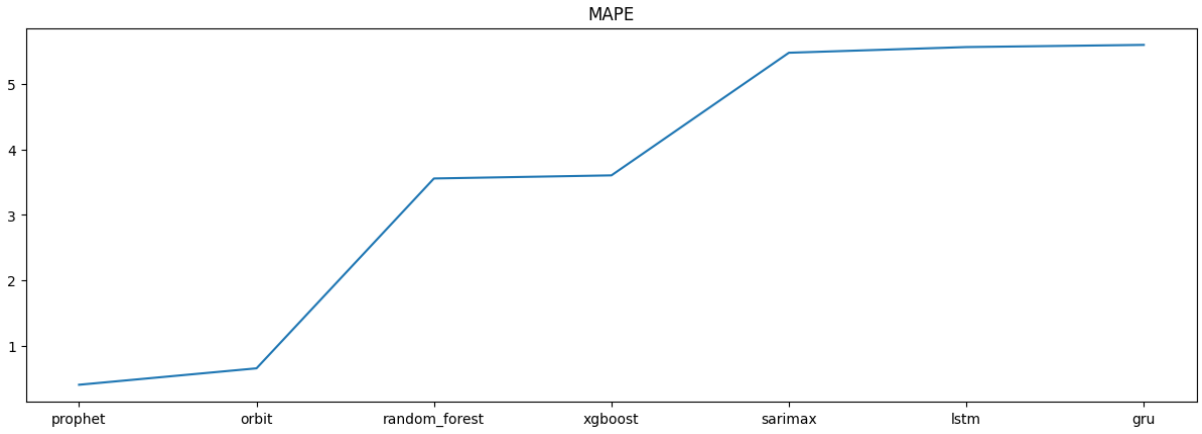
شکل ۵ـ۱۲: امتیاز دقیق



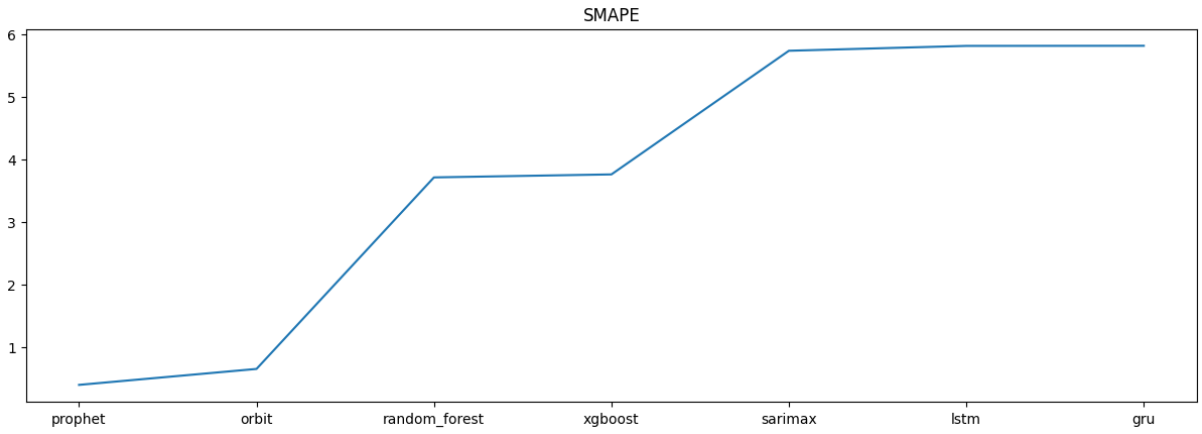
شکل ۵ـ۱۳: میانگین خطای متوسط



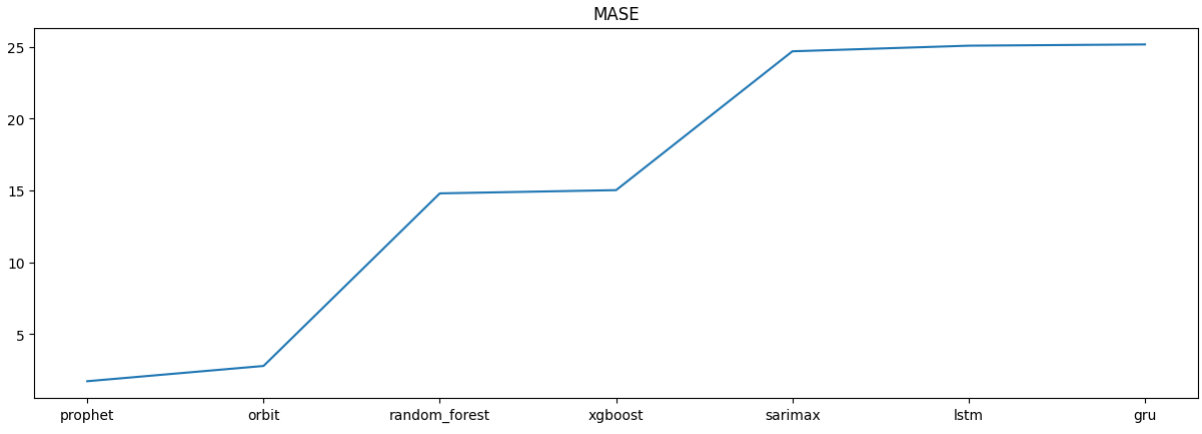
شکل ۵ـ۱۴: مجذور میانگین خطای متوسط



شکل ۵ـ۱۵: میانگین درصد خطای مطلق



شکل ۵ـ۱۶: میانگین درصد خطای متقارن



شکل ۵ـ۱۷: میانگین خطای مقیاس مطلق

فصل ۶

آزمایش‌های مربوط به پیش‌بینی روند بازار

پس از این که توسط آزمایشات فصل قبل به این نتیجه رسیدیم که مدل پرافت بیشترین دقت را دارد، این مدل را برای آزمایش‌های این بخش انتخاب کردیم. آزمایش‌های این بخش نتایج ایده‌ی شرح داده‌شده در فصل روش‌شناسی را نشان می‌دهند. در دو آزمایش از داده‌های روزانه و در آزمایش سوم از داده‌های ساعتی استفاده کردیم. سعی کردیم برای تنوع بیشتر در آزمایش‌ها از بازه‌های زمانی گوناگونی استفاده کنیم. آزمایش ۱ از سال‌های ۲۰۱۷ تا ۲۰۲۲ برای آموزش استفاده کرده‌است و سپس سال ۲۰۲۳ را مورد پیش‌بینی قرار داده‌است. آزمایش ۲ از سال ۲۰۱۶ تا ۲۰۲۱ برای آموزش استفاده کرده‌است و سال ۲۰۲۲ را پیش‌بینی کرده است. طول داده‌ی تست آزمایش ۱ و ۲ برابر با ۲۹۸ روز می‌باشد. در آزمایش آخر داده‌ی تست ما، داده‌های ساعتی سال ۲۰۱۸ تا اواسط ۲۰۲۱ می‌باشند و داده‌ی تست ما اواسط ۲۰۲۱ تا اواسط ۲۰۲۲ شامل ۷۶۷۱ ساعت می‌باشد.

جدول ۶ـ۱: نتایج متریک‌های گوناگون برای دو حالت با و بدون نقاط خارج از بازه‌ی پیش‌بینی

	Accuracy	F۱	Recall	Precision	شروع داده‌های آموزش	پایان داده‌های آموزش	شروع داده‌های تست	پایان داده‌های تست	نوع داده
آزمایش ۱ ـ تمام نقاط	۰/۶۹	۰/۷۰	۰/۷۱	۰/۶۹	۱/۱/۲۰۱۷	۷/۲۸/۲۰۲۲	۷/۲۸/۲۰۲۲	۶/۱۳/۲۰۲۳	روزانه
آزمایش ۱ ـ بدون نقاط خارج از بازه	۰/۶۸	۰/۶۹	۰/۶۸	۰/۷۰	۱/۱/۲۰۱۷	۷/۲۸/۲۰۲۲	۷/۲۸/۲۰۲۲	۶/۱۳/۲۰۲۳	روزانه
آزمایش ۱ ـ تنها نقاط خارج از بازه	۰/۷۴	۰/۷۴	۰/۷۲	۰/۷۶	۱/۱/۲۰۱۷	۷/۲۸/۲۰۲۲	۷/۲۸/۲۰۲۲	۶/۱۳/۲۰۲۳	روزانه
آزمایش ۲ ـ تمام نقاط	۰/۶۵	۰/۶۳	۰/۶۵	۰/۶۱	۱/۱/۲۰۱۶	۷/۲۸/۲۰۲۱	۷/۲۸/۲۰۲۱	۶/۱۳/۲۰۲۲	روزانه
آزمایش ۲ ـ بدون نقاط خارج از بازه	۰/۶۵	۰/۶۳	۰/۶۱	۰/۶۵	۱/۱/۲۰۱۶	۷/۲۸/۲۰۲۱	۷/۲۸/۲۰۲۱	۶/۱۳/۲۰۲۲	روزانه
آزمایش ۲ ـ تنها نقاط خارج از بازه	۰/۷۲	۰/۶۲	۰/۵۷	۰/۶۷	۱/۱/۲۰۱۶	۷/۲۸/۲۰۲۱	۷/۲۸/۲۰۲۱	۶/۱۳/۲۰۲۲	روزانه
آزمایش ۳ ـ تمام نقاط	۰/۷۰	۰/۷۱	۰/۷۱	۰/۷۰	۱/۱/۲۰۱۸	۷/۲۸/۲۰۲۱	۷/۲۸/۲۰۲۱	۶/۱۳/۲۰۲۲	ساعتی
آزمایش ۳ ـ بدون نقاط خارج از بازه	۰/۷۰	۰/۷۱	۰/۷۰	۰/۷۱	۱/۱/۲۰۱۸	۷/۲۸/۲۰۲۱	۷/۲۸/۲۰۲۱	۶/۱۳/۲۰۲۲	ساعتی
آزمایش ۳ ـ تنها نقاط خارج از بازه	۰/۹۳	۰/۸۶	۱	۰/۷۵	۱/۱/۲۰۱۸	۷/۲۸/۲۰۲۱	۷/۲۸/۲۰۲۱	۶/۱۳/۲۰۲۲	ساعتی

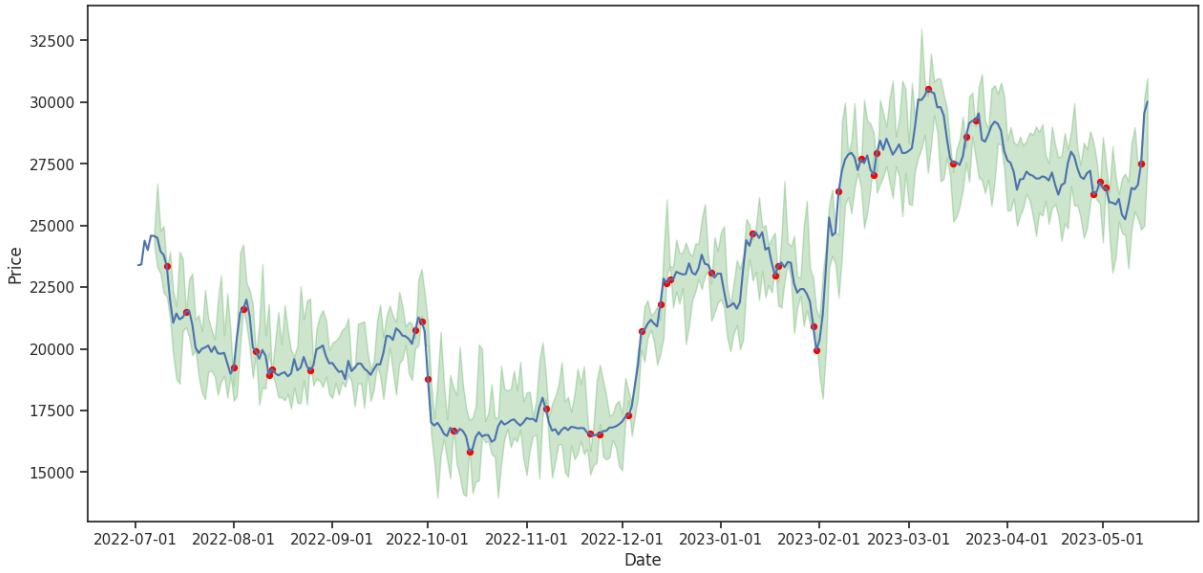
در آزمایش اول با قرار دادن سطح اطمینان [1] برابر با ۰/۹ با استفاده از روش رگرسیون چندکی همدیس، بازه‌های پیش‌بینی را به دست آوردیم. سپس اگر روزی در داده‌ی تست بیرون بازه اطمینانش می‌افتاد آن روز را به عنوان روزی که ممکن است در آن تغییر روند داشته باشیم مشخص می‌کردیم و روز بعد را معامله نمی‌کردیم. نتایج آزمایش در ۶ـ۱ قابل مشاهده است. سه بازه‌ی متفاوت برای داده‌ها انتخاب کردیم و در دو آزمایش اول از داده‌های روزانی و در آزمایش سوم از داده‌های ساعتی استفاده کردیم. در هر آزمایش ابتدا دقت کل نقاط داده‌ی تست را به دست آوردیم. سپس دقت مدل بعد از حذف نقاط روز بعد (یا ساعت بعد) هنگامی که داده‌ای خارج از بازه پیش‌بینی می‌افتد. و در آخر دقت مدل تنها در نقاط بعدی زمان‌هایی که نقطه واقعیت عینی [2] بیرون بازه پیش‌بینی قرار بگیرد. همان طور که در جدول مشاهده می‌شود بر خلاف فرض اولیه‌ای که احتمالا مدل در هنگام زمان‌هایی که تغییر روند داریم عملکرد ضعیف‌تری داشته باشد، شاهد عملکرد بهتری هستیم.

جدول ۶ـ۲: نتایج میزان بازدهی استراتژی برای دو حالت با و بدون نقاط خارج از بازه‌ی پیش‌بینی

	شروع سهام	پایان سهام	اوج سهام	بازده	بازدهی خرید و نگهداری	Profit Factor	Sharpe Ratio	مدت
آزمایش ۱	۱۰۰۰۰۰	۱۰۵۹۰۸	۱۱۵۲۳۷	۵/۹۱	۸/۳۷	۱/۱۱	۰/۰۳	۲۹۸
آزمایش ۱ ـ بدون نقاط خارج از بازه	۱۰۰۰۰۰	۶۵۳۰۴	۱۰۰۰۰۰	−۳۴/۷	۸/۳۷	۰/۴۹	−۰/۳۹	۲۹۸
آزمایش ۲	۱۰۰۰۰۰	۱۱۲۶۱۳	۱۲۹۶۶۵	۱۲/۶۱	−۳۶/۶۵	۱/۲۵	۰/۰۸	۲۹۸
آزمایش ۲ ـ بدون نقاط خارج از بازه	۱۰۰۰۰۰	۱۰۷۶۷۸	۱۲۲۶۱۱	۷/۶۸	−۳۶/۶۵	۱/۱۵	۰/۰۴	۲۹۸
آزمایش ۳	۱۰۰۰۰۰	۸۷۶۷۹	۱۱۵۰۶۱	−۱۲/۳۲	−۴۲/۱۷	۰/۸۷	−۰/۰۹	۷۶۷۱
آزمایش ۳ ـ بدون نقاط خارج از بازه	۱۰۰۰۰۰	۹۷۱۴۴	۱۱۲۰۱۰	−۲/۸۶	−۴۲/۱۷	۰/۹۸	−۰/۰۳	۷۶۷۱

در جدول ۶ـ۲ شاهد محاسبه بازدهی سهام در دو حالت گوناگون هستیم، یک حالت زمانی که با احتساب تمام نقاط معامله کنیم و دیگری در حالتی که بدون نقاط خارج از بازه پیش‌بینی معامله کنیم. لازم به ذکر است که مدل‌های یادگیری ماشین ما همیشه سودده نیستند و رسیدن به مدلی که بتواند به صورت قطعی به ما سود دهد، کاری است که در شرکت‌های سبدگردانی بزرگ جهان با سرمایه و تحقیقات بسیار زیاد صورت می‌گیرد. در نتیجه می‌توان عملکرد ضعیف مدل در بعضی نمونه‌های پیش‌بینی را یک مسئله عادی تلقی کرد. به صورت کل با خروجی مدل‌ها و بازه‌های زمانی معرفی شده در جدول ۶ـ۱ شروع به معامله کردیم و در آزمایش ۱ و ۲ شاهد این مسئله هستیم که نتایج بدتری بعد از حذف نقاط گرفته‌ایم و بازدهی کمتری نصیبمان شده است. در آزمایش ۳ عملکرد مدل ضرر ده بوده است اما با حذف این نقاط ضرر خود را کمتر کرده‌ایم و از ۱۲ هزار به ۲ هزار کاهش داده‌ایم. این مسئله که بازدهی ما کاهش می‌یابد می‌تواند منطقی باشد زیرا در جدول ۶ـ۱ میانگین دقت پیش‌بینی نقاط خارج از بازه ۷۹/۶ بود که عدد بسیار خوبی

[1] confidence level
[2] Ground Truth



شکل ۶ـ۱: نمودار مربوط به آزمایش ۱ ـ نقاط خارج از بازه‌ی پیش‌بینی با قرمز مشخص شده اند.

می‌باشد.

با توجه به عدم تطبیق توقع ما از ضعیف عمل کردن مدل در نقاطی که بیرون از بازه‌ی پیش‌بینی قرار دارند و نتایج آزمایش، در شکل ۶ـ۱ می‌توان مشاهده کرد که نقاط تغییر روند به خوبی توسط روش رگرسیون چندکی همدیس شناسایی شده‌اند. بازه‌ی پیش‌بینی هر داده توسط رنگ سبز مشخص شده است و اگر نقطه‌ای بیرون بازه مربوطه‌اش باشد با رنگ قرمز مشخص شده است. با وجود این که در این رابطه همچنان تعریف دقیق ریاضی ارائه نشده است اما به صورت شهودی می‌توان مشاهده نمود که اکثر نقاط تغییر روند توسط روش رگرسیون چندکی همدیس پیدا می‌شوند.

فصل ۷

نتیجه‌گیری

به صورت خلاصه، در این گزارش جنبه‌های مختلف پیش‌بینی ارزهای دیجیتال بررسی شد. مدل‌های مختلف یادگیری ماشین معرفی و پیاده‌سازی شد و مورد ارزیابی قرار گرفت. در ابتدا مروری بر ارزهای دیجیتال، ماهیت غیر متمرکز آن‌ها و تاثیر قابل توجه آن‌ها بر چشم‌انداز مالی ارائه می‌شود. به منابسب بودن مدل‌های یادگیری ماشین برای استراتژی‌های معاملات ارزهای دیجیتال اشاره شد و بر توانایی آن‌ها برای کشف روابط داده‌های پنهان تأکید شد.

کتابخانه CryptoPredictions به عنوان یک پلتفرم ارزشمند برای پیش‌بینی قیمت ارزهای دیجیتال، برای غلبه بر چالش‌هایی مانند کمبود مجموعه داده و نیاز به ارزیابی یکپارچه مدل‌های مختلف، معرفی شد. چند مورد از ویژگی‌های کتابخانه عبارت است از جمع آوری داده‌ها، ارزیابی مدل، و محاسبه اندیکاتورها. مدل‌هایی که در این پایان‌نامه به آن‌ها اشاره شد و در کتابخانه هم پیاده‌سازی شده‌اند عبارتند از جنگل تصمیم تصادفی، حافظه طولانی کوتاه مدت، اوربیت، آریما، ساریمکس، XGBoost و پرافت. هر مدل به طور مجزا به همراه جزئیات تئوری شرح داده شد.

سپس به ارزیابی عملکرد مدل‌های معرفی شده پرداخته شد. متریک‌های میانگین خطای مطلق، میانگین مربعات خطا، ریشه میانگین مربعات خطا، میانگین درصد مطلق خطا، خطای درصد مطلق میانگین متقارن، میانگین خطای مقیاس مطلق و میانگین مربعات خطای لگاریتمی به عنوان ابزار ضروری برای ارزیابی دقت و اثربخشی مدل‌های پیش‌بینی معرفی شدند.

به طور کلی، در حالی که مدل‌های مختلف سطوح متفاوتی از عملکرد را از نظر دقت و معیار نشان دادند، مشاهده شد که اوربیت و به‌ویژه پرافت به‌طور مداوم نتایج قوی را در معیارهای ارزیابی متعدد نشان دادند. این مدل‌ها پتانسیل ارائه پیش‌بینی دقیق و قابل اعتماد قیمت ارزهای دیجیتال را نشان دادند. در کنار این دو مدل، مدل آریما و ساریمکس هم در متریک‌های امتیاز دقت و امتیاز $F1$ نتایج خیلی خوبی را از

خود نشان دادند.

پس از یافتن بهترین مدل، به بخش اصلی پروژه یعنی پیش‌بینی تغییر روند بازار پرداختیم.برای این منظور از روش رگرسیون چندکی همدیس استفاده شد. با استفاده از این روش، بازه‌های قیمتی تعریف شده و نمره‌ای به هر روز یا هر ساعت اختصاص داده می‌شود که نشان می‌دهد در چه محدوده‌ای از بازه‌های قیمتی قرار دارد. با استفاده از این نمره انطباق، می‌توان نقاطی که تغییر روند در آن‌ها اتفاق می‌افتد را شناسایی کرد.

در آزمایشات انجام شده، عملکرد مدل پس از حذف نقاطی که خارج از بازه پیش‌بینی قرار می‌گیرند مورد بررسی قرار گرفت. نتایج نشان داد که عملکرد مدل در زمان‌هایی که تغییر روند را از داده حذف نکنیم، بهبود می‌یابد (بر خلاف فرض اولیه‌مان) و بازدهی بهتری حاصل می‌شود. همچنین، تغییر روند با استفاده از روش رگرسیون چندکی همدیس به صورت شهودی و با به تصویر کشیدن خروجی نشان‌دهنده این است که به خوبی تشخیص داده می‌شود و بیشتر نقاط تغییر روند مورد شناسایی قرار می‌گیرند.

با این وجود، باید توجه داشت که عملکرد مدل‌های پیش‌بینی بازار همیشه به صورت قطعی و سودآور نیست و ممکن است در برخی مواقع عملکرد آن‌ها ضعیف باشد. در این پروژه، هدف اصلی ما تشخیص نقاطی بود که تغییر روند در آن‌ها رخ می‌دهد و این هدف با استفاده از رگرسیون چندکی همدیس آزمایش شد.

برای کارهای آینده می‌توان روش‌های دیگر پیش‌بینی عدم اطمینان مدل را استفاده کرد. همچنین می‌توان بر روی ارائه یک فرمول ریاضی دقیق برای تعریف تغییرات روند بازار کار کرد. به دلیل تازگی این ایده، پتانسیل امتحان کردن روش‌های گوناگونی وجود دارد.

با توجه به نتایج حاصل از این پروژه و مشاهده عملکرد مدل، می‌توان این روش را در تحلیل و پیش‌بینی روند بازار و استراتژی‌های معامله استفاده کرد. این روش پتانسیل آزمایش‌های بسیار بیشتری را دارد که متاسفانه به دلیل کمبود وقت موفق به انجام آن نشدیم. اما امیدوارم که اطلاعاتی که ارائه شد و کتابخانه‌ای که آن را پیاده‌سازی کردیم مقدمه‌ای برای پیشبرد تحقیقات در این زمینه باشد.

مراجع

[1] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Annals of mathematics*, 2008.

[2] Coin market cap. *coinmarketcap.com/currencies/bitcoin/*.

[3] . S. A. Makarov, I. Trading and arbitrage in cryptocurrency markets. *Journal of Financial Economics*, 135(2):293–319, 2020.

[4] e. a. McNally, Sean. Predicting the price of bitcoin using machine learning. *Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 339–343, 2018.

[5] A. Geron. Hands-on machine learning with scikit-learn, keras, and tensorflow. *O'Reilly Media*, 2019.

[6] B. C. J. K. L. S. A. . V. V. Drucker, H. Support vector regression machines. advances in neural information processing systems. *Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 135:155–161, 1997.

[7] F. J. S. C. J. . O. R. A. Breiman, L. Hands-on machine learning with scikit-learn, keras, and tensorflow. *Classification and regression trees. CRC press*, 1984.

[8] L. Breiman. Random forests. *Machine learning*, 1(45):5–32, 2001.

[9] B. Y. . H. G. LeCun, Y. Deep learning. *Nature*, 521(7553):436–444, 2015.

[10] S. J. Z. X. L. F. . L. G. Zheng, H. Deep reinforcement learning for stock trading: From models to reality. *IEEE Transactions on Neural Networks and Learning Systems*, 44(3):113–125, 2018.

[11] . H. W. Grootveld, M. Machine learning for trading. *The Journal of Portfolio Management*, 44(3):113–125, 2018.

[12] M. H. . Z. X. Bollen, J. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

[13] G. W. . F. Y. Ma, J. News-driven stock market prediction using multi-scale deep neural networks. *Expert Systems with Applications*, 150:113–274, 2020.

[14] R. Koenker and G. B. Jr. Regression quantiles. *Econometrica: Journal of the Econometric Society*.

[15] I. Steinwart and A. Christmann. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211–225, 2011.

[16] D. R. Hunter and K. Lange. Journal of computational and graphical statistics. *Econometrica: Journal of the Econometric Society*, 9(1):60–77, 2000.

[17] J. W. Taylor. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of Forecasting*, 19(4):299–311, 2000.

[18] R. Koenker and K. F. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156, 2001.

[19] T. D. S. Ichiro Takeuchi, Quoc V. Le and A. J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

[20] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[21] N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

[22] A. G. Volodya Vovk and C. Saunders. Machine-learning applications of algorithmic randomness. *In International Conference on Machine Learning*, pages 444–453, 1999.

[23] A. G. Vladimir Vovk and G. Shafer. Algorithmic learning in a random world. *Springer*, 2005.

[24] V. V. Harris Papadopoulos, Kostas Proedrou and A. Gammerman. Inductive confidence machines for regression. *In European Conference on Machine Learning*, pages 345–356, 2002.

[25] H. Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. *In Tools in artificial intelligence. IntechOpen*, 2008.

[26] E. P. Y. Romano and E. Cand'es. Conformalized quantile regression. *in Advances in Neural Information Processing Systems*, pages 3543–3553, 2019.

[27] N. Tagasovska and D. Lopez-Paz. Frequentist uncertainty estimates for deep learning. *arXiv preprint arXiv:1811.00908*, 2018.

[28] T. M. Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999.

[29] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[30] L. Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[31] G. Biau. Analysis of a random forests model. *The Journal of Machine Learning Research*, 13(1):1063–1095, 2012.

[32] J. J. M. W. P. Aldi and A. Aditsania. Analisis dan implementasi long short term memory neural network untuk prediksi harga bitcoin. *eProceedings Eng*, 5(2), 2018.

[33] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput*, 9(8):1753–1780, 1997.

[34] F. Qian and X. Chen. Stock prediction based on lstm under different stability. *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 483–486, 2019.

[35] P. S. Y. Bengio and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans*, 1994.

[36] M. A. K. E. Kristensen, S. Østergaard and C. Enevoldsen. Technical indicators of financial performance in the dairy herd. *Dairy Sci*, 2008.

[37] C. Scheier and W. Tschacher. Appropriate algorithms for nonlinear time series analysis in psychology. *in Nonlinear dynamics in human behavior, World Scientific*, pages 27–43, 1996.

[38] K. C. Y. B. Junyoung Chung, Caglar Gulcehre. Empirical evaluation of gated recurrent neural networks on sequence modeling.

[39] Introduction to gated recurrent unit (gru).

[40] H. C. S. Y. Edwin Ng, Zhishi Wang and S. Smyl. Orbit: Probabilistic forecast with exponential smoothing. *in Nonlinear dynamics in human behavior, World Scientific*, 2004.

[41] M. D. H. D. L. Bob Carpenter, Andrew Gelman. Stan : A probabilistic programming language.

[42] J. P. C. Eli Bingham. Pyro: Deep universal probabilistic programming.

[43] E. B. Dagum. The x-ii-arima seasonal adjustment method. 2005.

[44] A. Hendranata. Arima (autoregressive integrated moving average). 2003.

[45] Y. S. Lee and L. I. Tong. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. *Knowledge-Based Syst.*, 24(1):66–72, 2011.

[46] S. C. Hillmer and G. C. Tiao. An arima-model-based approach to seasonal adjustment. 10(1):5–24, 2017.

[47] S. E. Said and D. A. Dickey. Testing for unit roots in autoregressivemoving average models of unknown order. pages 599–607, 1984.

[48] From ar to sarimax: Mathematical definitions of time series models. *phosgene89.github.io*.

[49] Prophet: Automatic forecasting procedure. *facebook.github.io/prophet/*.

[50] C. G. Tianqi Chen. Xgboost: A scalable tree boosting system.

[51] Y. M. L. O. E. C. A. Picasso, S. Merello. Technical analysis and sentiment embeddings for market trend prediction. *Expert Syst. Appl*, 135:60–70, 2019.

[52] X. M. Y. Huang and Y. Deng. Natural visibility encoding for time series and its application in stock trend prediction. *Knowledge-Based Systems*, 232:107478, 2021.

واژه‌نامه

الف

اعتبارسنجی متقاطع ..... cross-validation

افزایش گرادیان شدید Extreme Gradient Boosting

ب

بارهای کاری ..... workload

بسته‌بندی ..... bagging

بیش‌برازش ..... Overfitting

پ

پوشش ..... coverage

ت

تابع ..... function

تنظیم کننده بالقوه ..... potential regularizer

تلفیقی ..... Integrated

توضیح ..... Explicative

تکنیک مجموعه‌ای ..... ensemble method

ج

جنگل تصمیم تصادفی ..... Random Forest

جنگل‌های تصادفی چندکی quantile random forests

چ

چین ..... fold

ح

حافظه طولانی کوتاه مدت ..... Long Short-Term Memory

خ

خارج از کیسه ..... out-ot-bag

خودرگرسیون ..... Autoregressive

د

دروازه تنظیم مجدد ..... Reset Gate

درخت تصمیم ..... Decision Trees

دروازه به روز رسانی ..... Update Gate

ر

رسمی ..... formal

رگرسیون چندکی همدیس Conformalized Quantile Regression

رگرسیون چندک شرطی ..... conditional quantile regression

روش تقسیم همدیس .... split conformal method

ش

شبکه عصبی بازگشتی . Recurrent Neural Network

ک

کمینه ..... minimum

م

مجموعه ..... set

مدل‌های سری زمانی ساختاری بیزی ..... structural Bayesian time series models

ن

نمره انطباق ..... conformity score

نزدک ..... NASDAQ

و

واحد بازگشتی دروازه‌ای ... Gated Recurrent Unit

هـ

همبستگی خودکار ..... autocorrelation

Abstract

Digital currencies have become an important asset in recent years, and the ability to predict their price is of interest to investors and traders. This article examines the use of machine learning models to predict digital currency.

In this thesis, we introduce and implement the CryptoPredictions library, which provides a platform for implementing and evaluating different machine learning models for cryptocurrency price prediction. We then describe the various models we implemented in the library, including Random Forest, LSTM, Orbit, Arima, Sarimax, XGBoost, and Prophet. Through various tests, we came to the conclusion that the best performance is related to the Prophet model.

Another part of this thesis is the use of the Conformalized Quantile Regression(CQR) method to predict the change of the market trend. This method defines price prediction ranges and assigns each day or hour a conformity score that indicates how our price prediction interval should be calculated. Using these prediction intervals, it is possible to identify the points where the trend changes.

After that, we conduct experiments to measure the effectiveness of our new method. During these tests, we come to the conclusion that, contrary to the initial assumption that we have the possibility of reducing the accuracy of the model when the trend changes, the accuracy of the model increases at the points where the trend changes.

Keywords: Digital currency, machine learning, forecasting, trend prediction

Sharif University of Technology

Department of Computer Engineering

B.S. Thesis

Predicting Changes in the Trend of Cryptocurrencies

By:

Amirhossein Alimohammadi

Supervisor:

Dr. Ehsaneddin Asgari

July 2023