اعمال مدل‌های رگرسیون بر زیرمجموعه‌های با همبستگی بالا برای بهبود جایگذاری مقادیر جاافتاده عددی

امیرمسعود سفیدیان[۱]، کارشناسی ارشد؛ نگین دانشپور[۲]، استادیار

۱- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران - amirmasoud.sefidian@sru.ac.ir
۲- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران - ndaneshpour@sru.ac.ir

چکیده: حضور مقادیر جاافتاده در داده‌های دنیای واقعی مشکلی بسیار رایج و غیرقابل‌اجتناب است. بنابراین لازم است تا پیش از عملیات اکتشاف دانش، این مقادیر جاافتاده به‌طور دقیق پُر شوند. در این مقاله، سه رویکرد جدید برای تخمین مقادیر جاافتاده عددی پیشنهاد می‌شود. در تمامی روش‌های پیشنهادی، مدل‌های رگرسیون بر زیرمجموعه‌هایی با همبستگی بالا اعمال می‌شوند. در انتخاب زیرمجموعه‌های مطلوب سعی می‌شود تا همبستگی بین صفت جاافتاده و دیگر صفات حداکثر شود. انتخاب این زیرمجموعه‌ها با استفاده از رویکردهایی مبتنی بر انتخاب روبه‌جلو انجام می شود. از معیار ضریب همبستگی برای اندازه‌گیری میزان ارتباط بین صفات استفاده‌شده است. همچنین در روش‌های پیشنهادی، ترتیب صفات جاافتاده برای انجام عمل جایگذاری اولویت‌دهی می‌شوند. عملکرد رویکردهای پیشنهادشده بر روی پنج مجموعه‌داده از دنیای واقعی با مقادیر مختلف جاافتادگی ارزیابی‌شده است. عملکرد رویکردهای ارائه‌شده با پنج رویکرد جایگذاری با مقدار میانگین، جایگذاری با استفاده از نزدیک‌ترین همسایگان، روش جایگذاری با خوشه‌بندی c-means فازی، روش جایگذاری با درخت تصمیم و روشی مبتنی بر رگرسیون به نام «الگوریتم جایگذاری با رگرسیون افزایشیِ صفات» (IARI) مقایسه شده است. از دو معیار شناخته‌شده‌ی ریشه میانگین مربعات خطا و ضریب تعیین برای مقایسه عملکرد رویکردهای پیشنهادی با دیگر روش‌های جایگذاری استفاده‌شده است. نتایج آزمایش‌ها نشان می‌دهد که رویکردهای ارائه‌شده، حتی زمانی که درصد جاافتادگی بالا است، بهتر از دیگر روش‌های مقایسه شده عمل می‌کنند.
واژه‌های کلیدی: جایگذاری مقادیر جاافتاده، همبستگی، رگرسیون.

Applying Regression Models on Subsets with High Correlations for a Better Numeric Missing Values Imputation

Amir Masoud Sefidian[1], MSc; Negin Daneshpour[2], Assitant Professor

1- Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, Email: amirmasoud.sefidian@sru.ac.ir
2- Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, Email: ndaneshpour@sru.ac.ir

Abstract: The presence of missing values in the real world data is a very prevalent and inevitable problem. So, it's necessary to fill up these missing values accurately, before they are used for knowledge discovery process. This paper proposes three novel methods to fill numeric missing values. All of the proposed methods apply regression models on subsets of data which there are strong correlations among them. These subsets are selected using forward selection based approaches. In the selection of the desired subsets, it is tried to maximize the correlation between missing attribute and other attributes. The correlation coefficient is used to measure the relationships between attributes. The priority of each missing attribute for imputation purpose is also considered in the proposed methods. The performance of proposed methods is evaluated on five real world datasets with different missing ratios. The efficiency of the proposed methods is compared with five different estimation methods, namely, the mean imputation, the k nearest neighbours imputation, a fuzzy c-means based imputation, a decision tree based imputation, and a regression based imputation algorithm, called "Incremental Attribute Regression Imputation" (IARI) method. Two well-known evaluation criteria, namely, Root Mean Squared Error (RMSE) and Coefficient of Determination (CoD) are used to compare the performance of proposed methods with other imputation methods. Experimental results show that the proposed methods perform better than other compared methods, even when the missing ratio is high.
***Keywords*:** Missing values imputation, Correlation, Regression.

تاریخ ارسال مقاله: ۱۳۹۶/۰۵/۰۸
تاریخ اصلاح مقاله: ۱۳۹۶/۰۸/۰۶
تاریخ پذیرش مقاله: ۱۳۹۶/۰۹/۲۰
نام نویسنده مسئول: نگین دانشپور
نشانی نویسنده مسئول: ایران - تهران - لویزان - دانشگاه تربیت دبیر شهید رجایی - دانشکده مهندسی کامپیوتر.

۱- مقدمه

اولین گام در فرآیند کشف دانش، جمع‌آوری داده‌ها است. در این گام، نمونه‌ها از تک منبع یا منابع چندگانه جمع‌آوری می‌شوند. داده‌کاویِ باکیفیت بالا فقط هنگامی حاصل می‌شود که داده‌های جمع‌آوری‌شده کیفیت بالایی داشته باشند. نواقص مختلفی مثل ناسازگاری‌ها، خطاها، داده‌های پرت و مقادیر جاافتاده در داده‌های جمع‌آوری‌شده وجود دارد. بنابراین برای حاصل شدن نتایج مطلوب از هدفِ کشف دانش، این داده‌ها نیاز به پیش‌پردازش دارند. پیش‌پردازش داده‌ها اهمیت فراوانی دارد، به‌نحوی‌که حدود ۸۰٪ از کل تلاش برای مهندسی داده صرف آماده‌سازی داده‌ها برای مراحل بعد می‌شود [۱]. ازاین‌رو، تلاش‌های متعددی برای بهبود کیفیت داده‌ها در حوزه‌های مختلف علمی و صنعتی صورت گرفته است. برای مثال از روش‌های ترکیبی داده‌کاوی برای بهبود کیفیت داده های جمع‌آوری‌شده در صنایع نفت و گاز [۲] و از شبکه‌های عصبی برای پیش‌پردازش در حوزه‌ی پردازش تصویر [۳] استفاده‌شده است.

یکی از مشکلات اساسی در دنیای واقعی، وجود مقادیر جاافتاده است. در این حالت مقدار یک یا چند صفت از یک یا چند نمونه در یک مجموعه‌داده به دلایل مختلف، نامعلوم یا تهی[۱] است. حضور مقادیر جاافتاده در داده‌ها نه‌تنها مشکلی رایج در بسیاری از تحقیقات تجربی مثل تحقیقات صنعتی، تجاری و علمی است، بلکه مشکلی غیرقابل‌اجتناب است [۴]. برای مثال، حدود ۴۵٪ از مجموعه‌های موجود در مخزن داده‌ی UCI [۵] (که یکی از مشهورترین مخازن داده برای اهداف یادگیری ماشین است) شامل مقادیر جاافتاده هستند [۶]. بنابراین یک وظیفه‌ی مهم در پیش‌پردازش داده‌ها، تخمین تا حد ممکن دقیقِ مقادیر جاافتاده با استدلال از داده‌های کاملِ مشاهده‌شده برای افزایش کیفیت داده‌ها است. از این عملیات با نام‌های مختلفی ازجمله جایگذاری[۲]، تخمین و یا پُر کردن مقادیر جاافتاده در تحقیقات علمی یاد می‌شود. جایگذاری مقادیر جاافتاده همانند دیگر وظایف پیش‌پردازش امری زمان‌بر است [۷].

مقادیر جاافتاده به دلایل متفاوتی ازجمله خرابی سیستم قدرت، عوامل محیطی، اندازه‌گیری‌های غلط، رویه‌های دستی برای ثبت داده و اختلال در عملکرد تجهیزات ذخیره‌سازی داده رخ می‌دهند [۱۰-۸].

مقادیر جاافتاده به‌ویژه هنگامی‌که نرخ جاافتادگی بالا باشد، موجب ایجاد مشکلات متعددی می‌شود. حضور مقادیر جاافتاده به‌طور مستقیم کیفیت نتایج نهایی به‌دست‌آمده از فرآیند یادگیری یا کاوش از داده‌ها را تحت تأثیر قرار می‌دهد [۱۱]. بسیاری از الگوریتم‌های تحلیل داده‌ها قابلیت اعمال به مجموعه‌داده‌هایی با مقادیر جاافتاده را ندارند، چراکه اصولاً این الگوریتم‌ها به‌گونه‌ای طراحی‌شده‌اند که فقط با داده‌های کامل می‌توانند کار کنند [۱۲]. وجود مقادیر جاافتاده می‌تواند منجر به نتیجه‌گیری‌های گمراه‌کننده، غیر صحیح و غیرمعقول از یک تحقیق علمی شود و ممکن است کل فرآیند جمع‌آوری و تحلیل داده‌ها را برای کاربران بی‌فایده کند [۱۳]. علاوه بر این، حضور مقادیر جاافتاده می‌تواند قابلیت تعمیمِ یافته‌های یک تحقیق را محدود کند [۱۰].

همبستگی[۳] بین صفات در یک مجموعه‌داده از ویژگی‌های طبیعی آن مجموعه‌داده است. اغلب ممکن است زیرمجموعه‌هایی درون یک مجموعه‌داده یافت که روابط قوی‌تری نسبت به روابط حاکم بر کل مجموعه‌داده در آن‌ها وجود داشته باشد. در کاربردهای یادگیری ماشین یا داده‌کاوی معمولاً صفات با همبستگی بالا به‌عنوان صفات افزونه[۴] در نظر گرفته می‌شوند و فقط از یکی از این صفات برای انجام تحلیل‌ها استفاده می‌شود. اگرچه، این روابط می‌توانند شامل اطلاعات بسیار مفیدی برای هدف تخمین مقادیر جاافتاده باشند و مقادیر نامعلوم را می توان با استفاده از این روابط تخمین زد.

روش‌های بسیار زیادی مبتنی بر رگرسیون[۵] برای هدف تخمین مقادیر جاافتاده وجود دارند. اکثر این روش‌ها مانند روش جایگذاری با استفاده از رگرسیون افزایشیِ صفات (IARI[۶]) [۱۴]، مدل‌های رگرسیون را بر روی کل مجموعه‌داده اعمال می‌کنند. بااین‌حال، نکته‌ی مهم در مورد این روش‌ها این است که دقت این روش‌ها هنگامی‌که روابط در کل مجموعه‌داده ضعیف باشند، به‌شدت افت می‌کند. اعمال مدل‌های رگرسیون بر افرازهایی از داده‌ها با همبستگی بالا می‌تواند منجر به تولید نتایج بهتری در تخمین داده‌ها شود. بنابراین، عملکرد این روش‌ها با توجه به این نکته می‌تواند بهبود یابد.

با مطالعات انجام‌شده توسط نویسندگان، مشاهده شد که تنها روش DMI[۷] [۱۵] بحث همبستگی در داده‌ها را برای بهبود تخمین داده‌های جاافتاده موردتوجه قرار داده است. اگرچه این روش بخش‌هایی با همبستگی بالا را توسط درخت تصمیم پیدا می‌کند، بااین‌حال از معیار دقیقی برای سنجش میزان همبستگی بین صفات استفاده نمی‌کند و همواره فرض می‌کند که استفاده از درخت تصمیم منجر به افزایش همبستگی در داده‌ها می‌شود. ازاین‌رو، ضروری به نظر می‌رسد که مقدار همبستگی را توسط معیاری دقیق برای اهداف ذکرشده اندازه‌گیری کرد. با توجه به این نکات، در این پژوهش روش‌هایی معرفی می‌شوند که قادر به یافتن بخش‌هایی از داده‌ها با همبستگی بالا توسط معیاری کمّی برای اندازه‌گیری همبستگی قوی هستند. بنابراین با استفاده از روش‌های پیشنهادشده ابتدا سعی می‌شود تا زیرمجموعه‌هایی با همبستگی بالا را نسبت به هر نمونه‌ی جاافتاده یافت. سپس روش رگرسیون را بر آن زیرمجموعه‌ها اعمال کرد تا دقت مدل تخمین‌زننده را تا حد امکان بالا برد. با توجه به مطالعات صورت گرفته، تاکنون چنین رویکردی برای استفاده در تخمین مقادیر جاافتاده برای مدل رگرسیون ارائه نشده است. به‌علاوه، همان‌طور که گفته شد، تاکنون از معیار دقیقی برای سنجش میزان همبستگی در یک مجموعه‌داده برای هدف بهبود تخمین مقادیر جاافتاده عددی استفاده‌نشده است.

با توجه به نکات گفته‌شده، در این مقاله سه روش جدید پیشنهاد می‌شود تا مشکلات روش‌های گذشته را تا حد امکان رفع کنند. به بیان دقیق‌تر، از معیاری کمّی به نام ضریب همبستگی، برای یافتن زیرمجموعه‌هایی از داده‌ها با همبستگی بالا نسبت به هر نمونه‌ی جاافتاده استفاده می‌شود. ترتیب انتخاب صفات جاافتاده برای عمل جایگذاری می

تواند کیفیت جایگذاری را تحت تأثیر قرار دهد. بنابراین در روش‌های پیشنهادی از رویکردی مبتنی بر جنگل تصادفی[۸] [۱۶] برای رتبه‌بندی صفات جاافتاده جهت بهبود دقت مقادیر تخمین زده‌شده، استفاده‌شده است [۱۴].

به‌طور خلاصه، در روش‌های ارائه‌شده پس از تعیین اولویت صفات جاافتاده برای پُر شدن، دو گام اصلی در رویکردهای پیشنهادی وجود دارد که عبارت‌اند از: گام حداکثرسازی و گام تخمین. در گام حداکثرسازی زیرمجموعه‌هایی با همبستگی بالا نسبت به نمونه جاافتاده، پیدا می‌شوند. در گام تخمین، مقادیر جاافتاده با اعمال یک یا چند مدل رگرسیون خطی بر زیرمجموعه‌های یافت شده تخمین زده خواهند شد. با ترکیب انواع مختلف گام‌های حداکثرسازی و تخمین، سه رویکرد متفاوت در این مقاله پیشنهاد می‌شود.

بخش‌های بعدی این مقاله بدین شرح است. در بخش دوم مروری بر پیشینه تحقیق مرور می‌شود. در بخش سوم جزئیات رویکرد پیشنهادی ارائه می‌شود. در بخش چهارم جزئیات آزمایش‌های انجام‌شده و نتایج تحلیل‌ها بیان می‌شوند. درنهایت در بخش پنجم نتیجه‌گیری بیان می‌شود.

۲- پیشینه تحقیق

در این بخش، مروری بر روش‌های مختلف موجود برای هدف جایگذاری مقادیر جاافتاده انجام می‌شود.

رویکردهای ساده‌ی متفاوتی برای رسیدگی به مقادیر جاافتاده وجود دارد:

- حذف نمونه‌هایی با مقادیر جاافتاده
- پُر کردن دستی مقادیر جاافتاده
- پُر کردن با مقدار ثابت یا مقادیر محتمل (مانند مقدار میانگین)

در رویکرد اول اگر تعداد مقادیر جاافتاده بالا باشد، بسیاری از داده‌های موجود را بیهوده هدر می‌دهد. رویکرد دوم فقط وقتی عملی است که در داده‌ها تعداد نسبتاً کمی مقدار جاافتاده وجود دارد. مشکل رویکرد سوم نادیده گرفتن روابط بین صفات است. به‌علاوه در این رویکرد بحث توزیع داده‌ها نادیده گرفته خواهد شد؛ چراکه تمام مقادیر جاافتاده با یک مقدار یکسان پُر خواهند شد. بنابراین این رویکردهای ساده نمی توانند دقت کافی برای پُر کردن مقادیر جاافتاده را فراهم کنند. به همین دلیل، محققان زیادی به مسئله‌ی جایگذاری مقادیر جاافتاده توجه کرده‌اند. این تحقیقات شامل طیف وسیعی از روش‌ها می‌شود؛ از روش‌های ساده‌ی آماری تا روش‌های پیچیده‌ی مبتنی بر داده‌کاوی و یادگیری ماشین.

دسته‌ای از روش‌ها با استفاده از قوانین انجمنی[۹] داده‌های جاافتاده را پر می‌کنند. محدودیت اصلی این روش‌ها این است که تنها قابل‌استفاده برای داده‌های غیر عددی هستند. یکی از روش‌های موجود در این دسته از رویکردها، امتیازدهی به قوانین انجمنی است [۱۷]. ابتدا مجموعه‌داده ی رابطه‌ای به مجموعه‌داده‌ی تراکنشی تبدیل می‌شود تا بتوان از آن قوانین انجمنی را استخراج نمود. پس از یافتن قوانین یافت شده از مجموعه‌داده‌ی تراکنشی، برای هر مقدار جاافتاده، مجموعه قوانینی که در سمت راست آن‌ها همان صفت جاافتاده حضور دارند، انتخاب می شوند. برای هر یک از قوانین انتخاب‌شده امتیازی محاسبه می‌گردد. قانونی با بیش‌ترین امتیاز به‌عنوان قانون برگزیده انتخاب‌شده و از مقدار سمت راست آن قانون به‌عنوان مقدار جایگزین صفت مقدار صفت جاافتاده استفاده می‌شود. از نقاط قوت این روش می‌توان به دخیل کردن معیار وابستگی دو طرف قوانین انجمنی و طول قوانین و حمایت از مقادیر جاافتاده‌ی چندگانه اشاره کرد.

از شبکه‌های عصبی نیز برای پُر کردن مقادیر جاافتاده استفاده‌شده است. رایج‌ترین روش مبتنی بر شبکه‌های عصبی برای تخمین مقادیر جاافتاده، استفاده از شبکه‌ی عصبی پرسپترون چندلایه (MLP[۱۰]) است. در یکی از روش‌ها، داده‌های جاافتاده برای هجده مجموعه‌داده تخمین زده‌شده‌اند [۷]. همچنین تأثیر روش‌های مختلف یادگیری و پارامترهای مختلف شبکه عصبی بر روی دقت نهایی نیز بررسی‌شده‌اند. با طراحی و آموزش یک شبکه عصبی نگاشت خودسازمان‌ده[۱۱] (SOM) نیز می‌توان مقادیر جاافتاده را تخمین زد [۱۸]. مشکل اصلی شبکه‌های عصبی زمان بر بودن فرآیند یادگیری و ساخت مدل بهینه برای یک مجموعه‌داده نسبت به دیگر روش‌ها و همچنین قابلیت تفسیر[۱۲] پایین است [۱۹]. به‌علاوه، اصولاً آموزش مناسب شبکه‌های عصبی نیاز به تعداد نمونه‌های ورودی بالایی دارد. بنابراین هنگامی‌که تعداد نمونه‌های جاافتاده بالا است، دقت تخمین این روش‌ها کاهش می‌یابد.

روش k نزدیک‌ترین همسایگان[۱۳] (kNN) یکی از معروف‌ترین و پایه ترین روش‌ها برای تخمین مقادیر جاافتاده است. سادگی و نتایج قابل‌قبول آن نسبت به دیگر روش‌ها از ویژگی‌های آن است. از دیگر مزایای این روش قابلیت استفاده برای هم صفات عددی و هم صفات غیر عددی با تغییر تابع سنجش فاصله است. همچنین در این روش زمانی صرف ساخت مدل برای تخمین داده‌های جاافتاده نمی‌شود [۲۰]. یک روش مبتنی بر رویکرد k نزدیک‌ترین همسایگان برای پُر کردن مقادیر جاافتاده، استفاده از درجه‌ی جاافتادگی (تعداد صفات جاافتاده) هر نمونه است [۲۱]. در این رویکرد ابتدا درجه‌ی جاافتادگی هر نمونه محاسبه می‌شود. نمونه‌هایی که درجه‌ی جاافتادگی آن‌ها بزرگ‌تر از دو است از ادامه‌ی پردازش خارج می‌شوند و سپس مقادیر جاافتاده با مقادیر نزدیک ترین نمونه‌های کامل پر می‌شوند. از نقاط قوت این روش می‌توان به‌سادگی رویکرد ارائه‌شده و قدرت تعمیم بالا برای کاربردهای مختلف با تغییر معیار شباهت اشاره کرد. عدم توانایی در پُر کردن مقادیر جاافتاده ی چندگانه، حساسیت به داده‌های پرت و حساسیت بالا به تابع شباهت از نقاط ضعف این روش است. همچنین لزوماً بررسی فقط نزدیک‌ترین نمونه، همواره بهترین گزینه نیست.

یک رویکرد ارائه‌شده برای تخمین مقادیر جاافتاده، روش k نزدیک‌ترین همسایه‌ی تکراری مبتنی بر گِری[۱۴] به نام GBKII[۱۵] است [۲۲]. رویکرد مطرح‌شده در یک فرآیند تکراری بر روی مقادیر جاافتاده،

تک‌به‌تک مقادیر جاافتاده را پر می‌نماید. از مزیت‌های این روش امکان پُر کردن مقادیر جاافتاده چندگانه و کاراتر بودن پیچیدگی زمانی به دلیل اینکه تنها بین نمونه‌های هم برچسب مقایسه شباهت انجام می‌گردد، است. نیاز به برچسب کلاس داشتن نمونه‌ها، ضعف اصلی این روش است. یک نسخه‌ی بهبودیافته مبتنی روش نزدیک‌ترین همسایگان برای تخمین مقادیر جاافتاده، روش k نزدیک‌ترین همسایه و استفاده از خوشه‌بندی داده‌های قرارگرفته در یک دسته (CKNNI[۱۶]) است [۲۰]. مشکل اصلی این روش نیاز به دانستن برچسب کلاس تمام نمونه‌ها است. درواقع اگر کلاس نمونه‌ها مشخص نباشند، نمی‌توان از روش ذکرشده استفاده نمود.

مشکل اصلی روش‌های مبتنی بر نزدیک‌ترین همسایگان این است که برای مجموعه‌داده‌های بزرگ زمان زیادی برای جایگذاری صرف می‌شود، چراکه در این روش‌ها برای هر نمونه‌ی جاافتاده، روی کل مجموعه‌داده جستجوی کامل انجام می‌شود. به‌علاوه، تعیین دقیق تابع فاصله و مقدار مناسب برای k ممکن است چالش‌برانگیز باشد. همچنین در این روش‌ها از روابط موجود بین صفات در تخمین صفات استفاده نمی‌شود.

در یک روش مبتنی بر خوشه‌بندی، هر مقدار جاافتاده با میانگین دو مقدار جایگزین می‌شود [۲۳]. مقدار اول فاصله‌ی نمونه‌ی جاافتاده تا مرکز خوشه‌ای است که نمونه‌ی جاافتاده به آن تعلق دارد. مقدار دوم برابر با مقدار صفتی از مرکز خوشه است که متناظرا در نمونه‌ی جاافتاده، نامعلوم است. چشم‌پوشی از روابط احتمالی بین نمونه‌های موجود در یک مجموعه‌داده مشکل اصلی این روش است. یک نوع خوشه‌بندی برای تخمین مقادیر جاافتاده، خوشه‌بندی همسایگان مشترک است [۲۴]. از نقاط قوت این روش می‌توان به استفاده از معیار شباهت برای صفات ترکیبی و دخیل کردن معیار جدیدی برای سنجش شباهت نمونه‌ها اشاره کرد. نحوه توزیع داده‌ها بر تعداد همسایگان مشترک در این روش تأثیرگذار است و ممکن است ماتریس همسایگان مشترک تنک[۱۷] شود. متداول‌ترین روش‌های جایگذاری مبتنی بر خوشه‌بندی، خوشه‌بندی‌های k-means و c-means فازی هستند [۲۵، ۲۶]. یک روش ترکیبیِ ارائه‌شده مبتنی بر خوشه‌بندی و شبکه‌های عصبی، شامل دو مرحله است [۲۷]. این رویکرد شامل یک تخمین اولیه از مقادیر جاافتاده توسط نزدیک‌ترین مرکز خوشه به روش خوشه‌بندی k-means و یک گام پالایش و بهبود مقدار تخمین زده‌شده از گام قبلی، توسط شبکه‌ی MLP است. یک نسخه‌ی بهبودیافته از روش ذکرشده، استفاده از خوشه‌بندی c-means فازی به‌جای خوشه‌بندی متداول k-means است [۲۸]. در این روش هر نمونه با یک احتمال متعلق به هر خوشه است. این روش نتایج بهتری را به همراه داشته است. از نقاط قوت این روش‌ها می‌توان به بهبود k-means برای پُر کردن مقادیر جاافتاده اشاره کرد. نیاز به آموزش یک شبکه‌ی MLP برای هر مقدار جاافتاده و آموزش مجدد روی کل مجموعه آموزشی و تعیین ساختار (به‌هم‌بندی[۱۸] و وزن‌ها) بهینه‌ی شبکه‌ی MLP برای هر بار آموزش برای پُر کردن هر مقدار جاافتاده از نقاط ضعف این روش است.

استفاده از رویکردی ترکیبی مبتنی قوانین انجمنی و k نزدیک‌ترین همسایه، از دیگر روش‌های ارائه‌شده برای پُر کردن مقادیر جاافتاده است [۲۹]. از نقاط قوت این روش می‌توان به بهره‌گیری از قدرت ترکیب دو روش متفاوت اشاره کرد. درواقع در این روش سعی در استفاده از ارتباطات و فواصل موجود برای پُر کردن مقادیر جاافتاده شده است. این روش قابلیت استفاده برای داده‌های عددی را به‌طور مستقیم ندارد (به دلیل استفاده از قوانین انجمنی).

یک روش ترکیبی برای پُر کردن مقادیر جاافتاده، روش DMI[۱۹] است [۱۵]. این روش مبتنی بر الگوریتم EM[۲۰] و درخت تصمیم است. ایده‌ی اصلی این روش این است که وابستگی بین افرازهای مجموعه‌داده قوی تر از وابستگی بر کل مجموعه‌داده است و بنابراین الگوریتم EM بر روی بخش‌های متفاوت مجموعه‌داده بهتر عمل خواهد کرد و درنهایت نتیجه به‌دست‌آمده بهتر خواهد شد. افراز بندی مجموعه‌داده توسط درخت تصمیم صورت می‌پذیرد. از نقاط قوت این روش توجه به این است که وابستگی بین صفات لزوماً در کل مجموعه‌داده برقرار نیست. ازآنجایی‌که دقت الگوریتم EM هنگامی بالا است که همبستگی بین صفات قوی باشد، این روش برای مجموعه‌داده‌هایی خوب عمل می‌کند که روابط قوی در بین صفات آن‌ها برقرار باشد. به‌بیان‌دیگر، عدم وجود همبستگی بالا بین صفات منجر به کاهش دقت تخمین خواهد شد.

رگرسیون نیز به‌عنوان یک روش پایه برای تخمین مقادیر جاافتاده می‌تواند مورداستفاده قرار گیرد. زمانی که همبستگی بین صفات یک مجموعه‌داده قوی‌تر باشد، دقت رگرسیون نیز بیشتر خواهد بود. با استفاده از محاسبه‌ی معادله‌ی رگرسور خطی می‌توان داده‌های جاافتاده را از طریق داده‌های موجود تخمین زد. در یکی از روش‌های جایگذاری مقادیر جاافتاده به نام روش رگرسیون افزایشی صفات (IARI) که اخیراً مطرح‌شده است، تخمین مقادیر جاافتاده با استفاده از یک رویکرد افزایشی و تولید دنباله‌ای از رگرسورها انجام می‌شود [۱۴]. در این روش پس از رتبه‌بندی صفات، یک تخمین اولیه از مقادیر جاافتاده انجام می‌شود. پس‌ازآن صفات جاافتاده تک‌به‌تک و به‌صورت افزایشی جایگذاری می‌شوند. عمل جایگذاری در هر مرحله توسط مدل‌های رگرسیون انجام می‌شود. این کار آن‌قدر ادامه می‌یابد تا تمام صفات جاافتاده جایگذاری شوند. مشکل اصلی روش‌های مبتنی بر رگرسیون ضعف عملکرد آن‌ها در صورت عدم وجود روابط قوی بین صفات پیش‌بینی کننده و صفت پیش‌بینی‌شده است.

در سال ۲۰۱۶، دو راهبرد برای بهبود دقت پُر کردن مقادیر جاافتاده ارائه شد [۱۱]. ایده‌ی اصلی در هر دو راهبرد، انجام عمل انتخاب نمونه[۲۱] است. در این عمل سعی می‌شود تا نمونه‌های شامل داده‌های پَرت و داده‌های مغشوش[۲۲] از مجموعه‌داده حذف شوند تا کیفیت داده‌ها پیش از عملیات پُر کردن مقادیر جاافتاده افزایش یابد. درواقع مجموعه‌داده از یک فیلتر حذف داده‌های پرت و مغشوش عبور داده می‌شود. در هر دو راهبرد، عملیات انتخاب نمونه، از طریق الگوریتم DROP[۲۳] [۳۰] انجام‌شده است. تفاوت دو راهبرد پیشنهادی در ترتیب انجام عملیات

انتخاب نمونه و پُر کردن داده‌ها است. انجام عملیات انتخاب نمونه منجر به بهبود دقت حاصل از دسته‌بندیِ داده‌های تخمین زده‌شده است.

با بررسی روش‌های موجود، می‌توان پی برد که اصولاً روش‌هایی که روابطی بین نمونه‌ها را مدل می‌کنند و سپس از روابط یافت شده برای تخمین مقادیر جاافتاده استفاده می‌کنند، منجر به‌دقت بالایی در عملیات تخمین می‌شوند. ازجمله‌ی این روش‌ها می‌توان به رگرسیون اشاره کرد. اگرچه در این روش‌ها لازم است که روابط قوی بین صفات در مجموعه‌داده برقرار باشد؛ از طرفی ممکن است این روابط به‌طور صریح از مجموعه‌داده اصلی قابل‌استخراج نباشند. بنابراین در این مقاله سعی می‌شود تا با ارائه‌ی روش‌های مختلف، روابطی که به‌طور ضمنی بین نمونه‌ها وجود دارد را یافته و سپس از آن‌ها برای هدف جایگذاری مقادیر جاافتاده استفاده نمود. به‌علاوه، اگرچه تعدادی از روش‌های گذشته، مانند روش‌های مبتنی بر رگرسیون، با فرض قوی بودن روابط بر روی کل مجموعه‌داده به نتایج مناسبی دست یافتند، بااین‌حال، این روش‌ها از معیاری دقیق برای سنجش میزان روابط موجود استفاده نکرده‌اند. بنابراین، لازم است که ابتدا فرض قوی بودن روابط به‌طور دقیق ارزیابی شود و سپس این روش‌ها را اعمال نمود.

۳- رویکرد پیشنهادی

در این بخش و زیر بخش‌های آن مراحل مختلف رویکردهای پیشنهادی برای جایگذاری مقادیر جاافتاده شرح داده می‌شوند. همان‌طور که در دو بخش گذشته نیز ذکر شد، دو نکته‌ی مهم در روش‌های پیشنهادی وجود دارد. اول این‌که ترتیب پُر کردن صفات جاافتاده می‌تواند درنتیجه نهایی پُر کردن مقادیر جاافتاده تأثیرگذار باشد. دوم این‌که شناسایی بخش‌هایی از مجموعه‌داده‌ی که روابط خطی قوی بر آن‌ها حاکم است و سپس اعمال مدل‌های خطی داخل این بخش‌ها منجر به تخمین‌های دقیق‌تری از مقادیر جاافتاده می‌شود.

به‌طور خلاصه، در روش‌های ارائه‌شده پس از تعیین اولویت صفات جاافتاده برای پُرشدن، دو گام اصلی حداکثرسازی و تخمین اجرا می‌شوند. ابتدا در گام حداکثرسازی تلاش می‌شود تا زیرمجموعه‌های با همبستگی بالا نسبت به نمونه جاافتاده، پیدا شوند. برای این کار ابتدا یک مجموعه پایه از نمونه‌های کامل انتخاب می‌شود. پس‌ازآن، زیرمجموعه‌هایی با روابط قوی بین صفات، با بسط دادن مجموعه پایه ساخته می‌شوند. در گام تخمین مقادیر جاافتاده، با اعمال یک یا چند مدل رگرسیون خطی بر زیرمجموعه‌های یافت شده تخمین زده خواهند شد. رویکردهای پیشنهادی ماهیتی افزایشی دارند؛ یعنی پس از جایگذاری مقادیر جاافتاده برای هر صفت جاافتاده، از آن صفت به‌عنوان متغیرهای پیش‌بینی کننده برای تخمین‌های دیگر صفات استفاده خواهد شد. ماهیت افزایشی امکان استفاده از اطلاعات موجود در مقادیر جاافتاده را فراهم می‌کند.

فرض کنید مجموعه‌داده‌ی رابطه‌ای D شامل n نمونه است و هر نمونه مثل x شامل m صفت عددی $A_1, A_2, \ldots, A_m$ است. x_i^j مقدار صفت jام، در نمونه i ام را در D نشان می‌دهد. شکل (۱) رویه‌ی کلی را برای روش‌های ارائه‌شده نمایش می‌دهد.

Input:
D: An incomplete data set containing n instances, defined on schema $(A_1, A_2, \ldots, A_m)$
p: Percentage of complete records which will be selected as the base set
θ: Threshold for maximization algorithms

Output:
D: Imputed data set

$\mathbb{A} \leftarrow$ Find a set of attributes containing missing values in D
Calculate priority of each missing attribute for imputation using a Random Forest model
while $\mathbb{A} \neq \emptyset$ **do**
  $D_{comp} \leftarrow$ selectCompleteRecords(D)
  $D_{miss} \leftarrow$ selectMissingRecords(D)
  $j \leftarrow$ Index of the most important attribute in $\mathbb{A}$
  foreach $x \in D_{miss}$ *where* $x^j = NULL$ **do**
    $\mathfrak{B} \leftarrow$ findBaseSet(D_{comp}, x, p)
    $D' \leftarrow D_{comp} \backslash \mathfrak{B}$
    $maximizedDatasets \leftarrow$ maximizeCorrelation($x, j, \mathfrak{B}, D', \theta$)
    $x^j \leftarrow$ estimateByRegression($x, j, maximizedDatasets$)
  end
  $\mathbb{A} \leftarrow$ Delete A_j from $\mathbb{A}$
end
return D

شکل ۱: رویه کلی رویکردهای پیشنهادی برای تخمین مقادیر جاافتاده

در ابتدا، اولویت و اهمیت هر صفت جاافتاده برای پُر شدن، توسط یک مدل جنگل تصادفی محاسبه می‌شود (خطوط ۱ و ۲). پس‌ازآن، مجموعه‌داده‌ی ورودی به دو مجموعه‌داده‌ی کامل (D_{comp}) و مجموعه‌داده‌ی با مقادیر جاافتاده (D_{miss}) تقسیم می‌شود (خطوط ۴ و ۵). سپس صفتی با بالاترین اولویت جهت پُر شدن انتخاب می‌شود (خط ۶). آنگاه، نمونه‌هایی که مقدار جاافتاده‌ای در صفت jام (اندیس مهم‌ترین صفت جاافتاده) خود دارند، تک‌به‌تک برای فرآیند پُر شدن انتخاب می‌شوند (خط ۷). بعدازاین کار، برای هر نمونه انتخاب‌شده یک مجموعه پایه از نمونه‌های کامل انتخاب می‌شود. با اعمال یک روش حداکثرسازی، بسته به نوع روش، یک یا چند زیرمجموعه از نمونه‌های کامل با همبستگی بالا ساخته خواهند شد (خطوط ۸ تا ۱۰). حداکثرسازی با اعمال رویکردهایی مبتنی بر انتخاب روبه‌جلو[۲۴] بر مجموعه‌ی پایه ($\mathfrak{B}$) و با کمک نمونه‌های کاملِ غیرپایه (D') انجام می‌شود. درنهایت، مقدار جاافتاده در یک نمونه با اعمال یک یا چند مدل خطی بر روی زیرمجموعه یا زیرمجموعه‌های ساخته‌شده تخمین زده می‌شود. این کار آن‌قدر ادامه می‌یابد تا تمام صفات جاافتاده پر شوند.

همان‌طور که الگوریتم پیشنهادی نشان می‌دهد، پس از پُر کردن هر صفت جاافتاده، مقادیر آن صفت برای تخمین دیگر صفات جاافتاده در تکرارهای بعدی استفاده خواهد شد. درواقع، مجموعه‌داده‌های کامل (D_{comp}) پس از هر تکرارِ حلقه **while** به‌روزرسانی می‌شود. الگوریتم کلی پیشنهادی، دو پارامتر اصلی، یعنی p و θ دارد که نیاز به مقداردهی مناسب دارند. در زیر بخش‌های بعدی این بخش، پارامترها، نحوه‌ی انتخاب مجموعه پایه و گام‌های حداکثرسازی و تخمین و دیگر جزئیات روش‌های پیشنهادی شرح داده خواهند شد. درنهایت ترکیبات مختلف از گام‌های حداکثرسازی و تخمین برای تولید روش‌های پیشنهادی مختلف ذکر خواهند شد.

۳-۱- محاسبه اولویت هر صفت جاافتاده برای هدف تخمین داده‌های جاافتاده

صفات جاافتاده در روش‌های پیشنهادی تک‌به‌تک به ترتیب اهمیتشان طبق رویکردی مبتنی بر جنگل تصادفی پر خواهند شد. صفتی با بیش ترین اهمیت، ابتدا و صفتی با کم‌ترین اهمیت در انتها پر خواهد شد. اولویت هر صفت جاافتاده برای پُر شدن در سه گام محاسبه می‌شود. ابتدا مقادیر جاافتاده در مجموعه‌داده‌ی ورودی (شامل مقادیر جاافتاده) با استفاده از یک رویکرد تخمین ساده، مثل تخمین با مقدار میانگین، پر می‌شوند. سپس، یک مدل جنگل تصادفی بر روی مجموعه‌داده‌ی پرشده ساخته می‌شود. درنهایت، مدل ساخته‌شده به نمونه‌های خارج از کیسه[۲۵] اعمال می‌شود تا اهمیت هر صفت اندازه‌گیری شود. پس از پیدا کردن ترتیب هر صفت برای پُر شدن، صفتی با بیشترین اهمیت برای ادامه فرآیند پُر کردن انتخاب خواهد شد. بنابراین با فرض این‌که jاندیس مهم‌ترین صفت جاافتاده باشد، آنگاه نمونه‌هایی که در صفت j ام خود مقدار جاافتاده دارند، تک‌به‌تک برای ادامه در مراحل بعدیِ فرآیندِ جایگذاری انتخاب خواهند شد.

۳-۲- انتخاب مجموعه پایه

تا این مرحله، ترتیب تخمین هریک از صفات جاافتاده تعیین شد. از این مرحله به بعد، نحوه پُر کردن مقادیر جاافتاده برای هر نمونه‌ی جاافتاده شرح داده خواهد شد. در رویکردهای ارائه‌شده، هر صفت جاافتاده از یک نمونه‌ی جاافتاده با اعمال یک یا چند مدل رگرسیون خطی بر زیرمجموعه (هایی) از نمونه‌های کامل، تخمین زده خواهند شد. ویژگی این زیرمجموعه‌های انتخاب‌شده این است که همبستگی قوی بین صفت جاافتاده و دیگر صفات وجود دارد.

در گام‌های بعدی، روش‌های حداکثرسازیِ همبستگی از رویکردهای مبتنی بر انتخاب روبه‌جلو برای پیدا کردن زیرمجموعه‌های مطلوب استفاده خواهند کرد. ازاین‌رو، لازم است که یک مجموعه پایه مناسب از نمونه‌ها انتخاب شود. این مجموعه‌ی پایه باید دیدِ کلیِ خوبی از روابط موجود در نمونه‌هایی که در رویکرد انتخاب روبه‌جلو استفاده خواهند شد، بدهد. برای این منظور از روش معروف k نزدیک‌ترین همسایگان برای تشکیل مجموعه پایه برای هر نمونه‌ی جاافتاده استفاده می‌شود. بدین منظور p درصد از نزدیک‌ترین نمونه‌های کامل به نمونه‌ی جاافتاده، به‌عنوان k نزدیک‌ترین همسایگان برای آن نمونه جاافتاده انتخاب می‌شوند. این همسایگان به‌عنوان مجموعه پایه برای آن نمونه جاافتاده انتخاب می‌شوند. بنابراین خروجی این گام برای هر نمونه‌ی جاافتاده یک زیرمجموعه از نمونه‌های کامل خواهد بود.

فاصله‌ی بین دو نمونه‌ی x_i و x_j با استفاده از رابطه‌ی فاصله‌ی اقلیدسی مطابق با معادله (۱) محاسبه می‌شود.

$$d(x_i,x_j)=\sqrt{\sum_{k=1}^{m}\left[(x_i^k-x_j^k)^2 I(k)\right]},$$
$$I(k)=\begin{cases}1, & \text{if } x_i^k \text{ and } x_j^k \neq \text{NULL}\\ 0, & \text{otherwise}\end{cases} \qquad (۱)$$

همان‌طور که پیداست، در محاسبه‌ی فاصله‌ی دو نمونه، فقط صفات با مقادیر معلوم در محاسبات دخیل می‌شوند.

۳-۳- حداکثر کردن همبستگی بین صفات با بسط دادن مجموعه پایه

پس از انتخاب یک مجموعه پایه مناسب، الگوریتم تلاش می‌کند تا رابطه‌ی خطی بین صفت جاافتاده و دیگر صفات نمونه جاافتاده را در مجموعه پایه افزایش دهد. این افزایش با استفاده از بسط دادن مجموعه پایه توسط نمونه‌های غیرپایه به نحوی انجام می‌گیرد که همبستگی بین صفت جاافتاده و دیگر صفات حداکثر شود. این کار با به‌کارگیری یک رویکرد انتخاب روبه‌جلو انجام می‌شود. ایده‌ی اصلی برای انجام چنین کاری این است که در یک مدل رگرسیون خطی، وجود روابط خطی قوی‌تر بین متغیر (های) پیش‌بینی کننده[۲۶] و متغیر پاسخ[۲۷] منجر به تخمین بهتر متغیر پاسخ می‌شود.

رابطه خطی بین دو متغیر تصادفی را می‌توان با استفاده از ضریب همبستگی پیرسون[۲۸] (PCC) اندازه‌گیری کرد. مقدار PCC میزان همبستگی دو متغیر تصادفی را نشان می‌دهد. PCC می‌تواند مقداری بین ۱- و ۱+ داشته باشد. علامتِ ضریب همبستگی پیرسون، جهت رابطه‌ی خطی را نشان می‌دهد. با محاسبه مقدار PCC برای هر جفت از m صفات یک مجموعه‌داده مثل D، یک ماتریس متقارن همبستگی با اندازه‌ی $m\times m$ حاصل می‌شود. مدخل i,j در این ماتریس، مقدار PCC بین صفات A_i و A_j را در D نشان می‌دهد. در ادامه مقاله، نماد $\boldsymbol{\rho}(D)$ ماتریس همبستگی برای مجموعه‌داده‌ی D را نشان می‌دهد. نماد $\boldsymbol{\rho}(D)_{i,j}$ اشاره به مدخل ردیف i ام و ستون j ام در ماتریس همبستگی برای مجموعه‌داده‌ی D خواهد داشت. معادلات (۲) و (۳) نحوه محاسبه ماتریس همبستگی و همبستگی بین دو متغیر تصادفی A_i و A_j را نشان می‌دهد.

$$\boldsymbol{\rho}(D)=\begin{bmatrix}1 & \rho_{1,2} & \cdots & \rho_{1,m}\\ \rho_{2,1} & 1 & \cdots & \rho_{2,m}\\ \vdots & \vdots & \ddots & \vdots\\ \rho_{m,1} & \rho_{m,2} & \cdots & 1\end{bmatrix}_{m\times m} \qquad (۲)$$

که در آن

$$\boldsymbol{\rho}(D)_{i,j}=\left|\frac{Cov(A_i,A_j)}{\sigma_{A_i}\sigma_{A_j}}\right| \qquad (۳)$$

که در آن، $Cov(A_i,A_j)$ مقدار کوواریانس[۲۹] بین مقادیر دو صفت A_i و A_j را نشان می‌دهد. σ_{A_i} و σ_{A_j} به ترتیب بیانگر مقادیر انحراف از معیار برای صفات A_i و A_j هستند.

ازآنجایی‌که میزان همبستگی بین صفات (قوی یا ضعیف بودن) در روش‌های ارائه‌شده مهم است، نه جهت همبستگی (همبستگی مثبت یا منفی)، از قدر مطلق مقدار ضریب همبستگی در محاسبات ماتریس همبستگی استفاده می‌شود. بنابراین برای هر مقدار i و j خواهیم داشت: $۰ \leq \boldsymbol{\rho}(D)_{i,j} \leq ۱$.

با داشتن یک نمونه جاافتاده مثل x که در آن مقدار x^j جاافتاده است (صفت j ام آن نمونه مقدار تهی دارد)، الگوریتم‌های پیشنهادی تلاش خواهند کرد تا همبستگی بین صفت جاافتاده و دیگر صفات را با اعمال یک روش انتخاب روبه‌جلو بر روی مجموعه پایه افزایش دهند.

یک رویکرد کلی برای انتخاب یک ترکیب k تایی از D که مقدار ضریب همبستگی در آن زیرمجموعه بین صفت جاافتاده و دیگر صفات حداکثر است، انجام جستجوی کامل[۳۰] است؛ یعنی مقدار ضریب همبستگی را برای تمام زیرمجموعه‌های k تایی از D محاسبه کرده و زیرمجموعه‌های با بیشترین مقدار PCC را انتخاب کنیم. با فرض این‌که اندازه‌ی D برابر با n باشد، برای این کار $n!/k!(n-k)!$ انتخاب مختلف وجود خواهد داشت. بنابراین این رویکرد، به‌ویژه زمانی که اندازه‌ی D بسیار بزرگ است، بسیار زمان‌بر است. بنابراین نحوه‌ی انتخاب زیرمجموعه‌های مطلوب باید هوشمندانه‌تر باشد. پیشنهاد ما استفاده از رویکردی مبتنی بر انتخاب روبه‌جلو برای یافتن زیرمجموعه‌های مطلوب است. یعنی مجموعه‌ی پایه را با استفاده از رویکرد انتخاب روبه‌جلو بسط دهیم تا زیرمجموعه‌ی مطلوب حاصل شود. بدیهی است، این رویکرد یافتن بهترین زیرمجموعه‌ی ممکن را ضمانت نمی‌کند؛ بااین‌حال از رویکرد جستجوی کامل، منطقی‌تر به نظر می‌رسد. دو نوع الگوریتم حداکثرسازیِ همبستگی در این مقاله ارائه‌شده است. در زیر بخش‌های بعدی این الگوریتم‌ها شرح داده خواهند شد.

۳-۳-۱- حداکثر کردن همبستگی هر یک از صفات معلوم نسبت به صفت جاافتاده

الگوریتمِ اولِ حداکثرسازی سعی می‌کند تا مجموعه پایه را با در نظر گرفتن هر صفت غیر جاافتاده (معلوم) به‌طور جداگانه بسط دهد. ورودی‌های این الگوریتم عبارت‌اند از: اندیس یک صفت مثل j که صفات معلوم در مجموعه‌داده‌های خروجی نسبت به آن صفت حداکثر می‌شوند، یک مجموعه از نمونه‌های پایه، یک مجموعه دیگر از نمونه‌ها که نامزدهای اضافه شدن به مجموعه پایه هستند و یک پارامترِ حد آستانه (θ) که توسط کاربر تعریف می‌شود. خروجی الگوریتم m مجموعه‌داده‌ی $D_۱, D_۲, \ldots, D_m$ خواهد بود، به‌نحوی‌که همبستگی بین صفت A_i و صفت جاافتاده‌ی A_j در D_i حداکثر شده است. جزئیات این الگوریتم در شکل (۲) نمایش داده‌شده است.

در ابتدا، مقادیر تمام مجموعه داده‌های $D_i (i = ۱,۲,\ldots m$ برابر با مجموعه پایه قرار داده می‌شوند (خطوط ۱ تا ۳). سپس الگوریتم به‌طور موقت نمونه‌های غیرپایه (D') را تک‌تک به هر D_i اضافه می‌کند. پس از هر اضافه کردن، الگوریتم بررسی می‌کند که آیا مقدار همبستگی بین صفات A_i و A_j در D_i افزایش‌یافته است یا خیر (خط ۷). اگر مقدار PCC پس از اضافه کردن نمونه‌ی جدید بیش از مقدار حد آستانه، یعنی θ، افزایش‌یافته بود، آن نمونه به‌طور دائم به D_i اضافه خواهد شد (خط ۸). در غیر این صورت از آن نمونه چشم‌پوشی می‌شود و الگوریتم به سراغ نمونه‌ی بعدی می‌رود. این کار آن‌قدر ادامه می‌یابد تا تمام نمونه‌های غیرپایه بررسی شوند.

Input:
$\mathfrak{B}$: A base set of complete records for maximization
D': A set of records
j: Index of attribute which the output data sets will be maximized according to it, i.e., index of missing value ($1 \leq j \leq m$)
θ: Threshold value for adding a record from D' to $\mathfrak{B}$ ($0 \leq \theta \leq 1$)

Output:
$D_1, D_2, \ldots, D_m$: Set of data sets such that correlation between A_i and A_j(missing attribute) is maximized in data set D_i

```
1  for i ← 1 to m do
2  |  D_i ← B
3  end
4  foreach x ∈ D' do
5  |  for i ← 1 to m do
6  |  |  if i ≠ j then
7  |  |  |  if (ρ(D_i ∪ x)_{i,j} − ρ(D_i)_{i,j}) > θ then
8  |  |  |  |  D_i ← (D_i ∪ x)
9  |  |  |  end
10 |  |  end
11 |  end
12 end
13 return D_1, D_2, ..., D_M
```

شکل ۲: الگوریتم حداکثرسازی همبستگی هریک از صفات نسبت به صفات جاافتاده

۳-۳-۲- حداکثر کردن میانگین مقادیر همبستگی بین صفات معلوم و صفت جاافتاده

الگوریتم دومِ پیشنهادی سعی می‌کند تا میانگین مقادیر همبستگی بین صفات غیر جاافتاده و صفت جاافتاده را افزایش دهد. ورودی‌های این الگوریتم همانند الگوریتم قبلی هستند، اما خروجی تولیدشده متفاوت خواهد بود. خروجی یک تک مجموعه‌داده‌ی ($D_{\max}$) خواهد بود که در آن میانگین مقادیر همبستگی بین صفات غیر جاافتاده و صفت جاافتاده حداکثر شده است. مراحل این روش در شکل (۳) نمایش داده‌شده‌اند.

Input:
$\mathfrak{B}$: A base set of complete records for maximization
D': A set of records
j: Index of attribute which the output data set will be maximized according to it, i.e., index of missing value ($1 \leq j \leq m$)
θ: Threshold value for adding a record from D' to $\mathfrak{B}$ ($0 \leq \theta \leq 1$)

Output:
D_{max}: A data set in which mean of correlation values between non-missing attributes and A_j (missing attribute) is maximized.

```
1  D_max ← B
2  foreach x ∈ D' do
3  |  μ_current ← (Σ_{i=1, i≠j}^{m} [ρ(D_max)_{i,j}]) / (m−1)
4  |  μ_new ← (Σ_{i=1, i≠j}^{m} [ρ(D_max ∪ x)_{i,j}]) / (m−1)
5  |  if (μ_new − μ_current) > θ then
6  |  |  D_max ← (D_max ∪ x)
7  |  end
8  end
9  return D_max
```

شکل ۳: الگوریتم حداکثرسازی میانگین مقادیر همبستگی بین صفات غیر جاافتاده نسبت به صفت جاافتاده

در ابتدا، مقدار اولیه $D_{\max}$ برابر با مجموعه پایه قرار می‌گیرد (خط ۱). سپس الگوریتم به‌طور موقت نمونه‌های غیرپایه (D') را تک‌تک به $D_{\max}$ اضافه می‌کند. پس از هر اضافه کردن، الگوریتم بررسی می‌کند

که آیا میانگین مقدار همبستگی بین صفاتِ با مقادیر معلوم و صفت A_j افزایش‌یافته است یا خیر (خط ۷). اگر مقدار PCC پس از اضافه کردن نمونه‌ی جدید بیش از مقدار حد آستانه یعنی θ افزایش‌یافته بود، آن نمونه به‌طور دائم به $D_{\max}$ اضافه خواهد شد (خط ۸). در غیر این صورت از آن نمونه چشم‌پوشی می‌شود و الگوریتم به سراغ نمونه‌ی بعدی می رود. این کار آن‌قدر ادامه می‌یابد تا تمام نمونه‌های غیرپایه بررسی شوند.

۳-۴- تخمین مقادیر جاافتاده با استفاده از زیرمجموعه‌های یافته شده

پس از پیدا کردن زیرمجموعه‌های مطلوب از داده‌های کامل، آخرین گام، تخمین صفت جاافتاده در نمونه‌ی جاافتاده است. بسته به نوعِ روش حداکثرسازیِ همبستگی در گام قبلی، رویکرد تخمین مقدار جاافتاده متفاوت خواهد بود. در رویکرد اول، تعداد m مدل رگرسیون خطیِ ساده (تک صفتی) بر روی m مجموعه‌داده‌ی ساخته‌شده اعمال خواهد شد؛ به‌نحوی‌که در مجموعه‌داده‌ی D_i صفت A_i به‌عنوان متغیر مستقل و A_j به‌عنوان متغیر وابسته در ساخت مدل رگرسیون استفاده خواهد شد. هر مدل خطی رگرسیون در این حالت مطابق با معادله‌ی (۴) خواهد بود.

$$A_j = \boldsymbol{\alpha}_0 + \boldsymbol{\alpha} A_i \qquad (۴)$$

که در آن A_j نشان‌دهنده متغیر وابسته، A_i بیانگر متغیر مستقل و $\boldsymbol{\alpha}_0$ و $\boldsymbol{\alpha}$ ضرایب مدل رگرسیون ساخته‌شده هستند.

رویکرد پیشنهادی، هر مقدار جاافتاده را با محاسبه میانگین وزن‌دار m مقدار تخمین زده‌شده برای آن صفت جاافتاده جایگزین خواهد کرد. وزن مقدارِ تخمین زده‌شده‌ی i ام برابر با این است که تا چه میزان دو صفت A_i و A_j توانسته‌اند در D_i به یکدیگر به‌صورت خطی مرتبط شوند. یعنی وزن مقدار تخمین زده‌شده‌ی i ام برابر با $\boldsymbol{\rho}(D_i)_{i,j}$ خواهد بود.

در دومین رویکردِ تخمین، فقط یک مدل رگرسیون خطی چند صفتی بر روی خروجی گام حداکثرسازی اعمال خواهد شد. از صفت جاافتاده به‌عنوان متغیر وابسته و از باقی صفات (غیر جاافتاده) به‌عنوان متغیرهای مستقل برای ساخت مدل خطی استفاده خواهد شد. معادله‌ی یک مدل رگرسیون خطی چند صفتی مطابق با (۵) خواهد بود.

$$A_j = \boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1 A_1 + \boldsymbol{\alpha}_2 A_2 + \ldots + \boldsymbol{\alpha}_m A_m = \boldsymbol{\alpha}_0 + \sum_{i=1}^{m} \boldsymbol{\alpha}_i A_i \qquad (۵)$$

که در آن A_j نشان‌دهنده متغیر وابسته، $A_i\ (i = ۱,۲,...,m)$ بیانگر متغیرهای مستقل و $\boldsymbol{\alpha}_i\ (i = ۰,۱,۲,...,m)$ ضرایب مدل رگرسیون ساخته‌شده هستند.

۳-۵- ترکیب گام‌های حداکثرسازی و تخمین

با ترکیب روش‌های مختلف حداکثرسازی و تخمین ارائه‌شده، روش‌های متفاوتی را برای تخمین مقادیر جاافتاده می‌توان ارائه داد. جدول (۱) حالات مختلف ممکن برای تولید روش‌های مختلف را نشان می‌دهد. ستون اول، شناسه‌ی روش، ستون دوم، نحوه‌ی حداکثرسازی همبستگی و ستون سوم نحوه‌ی تخمین نهایی مقدار جاافتاده را نمایش می‌دهد.

جدول ۱: ترکیب‌های مختلف گام‌های حداکثرسازی و تخمین

شناسه	روش حداکثرسازی	روش تخمین
۱	حداکثر کردن همبستگی هریک از صفات	تک مدل خطی چند صفته
۲	حداکثر کردن همبستگی هریک از صفات	نظرخواهی وزن‌دار از چندین مدل خطی تک صفته
۳	حداکثر کردن میانگین مقادیر همبستگی تمام صفات	تک مدل خطی چند صفته

روش‌های ۱ و ۲ از الگوریتم ارائه‌شده در شکل (۲) برای گام حداکثرسازی استفاده می‌کنند. روش ۳ از الگوریتم ارائه‌شده در شکل (۳) برای گام حداکثرسازی استفاده می‌نماید.

در روش ۱ پس از تولید شدن خروجی گام حداکثرسازی ($D_1, D_2, \ldots, D_m$)، مجموعه‌داده‌ی D_{i^*} که در آن i^* از رابطه‌ی (۶) به دست می‌آید، انتخاب‌شده و سپس برای گام تخمین، به تک مدل خطیِ چند صفته (رویکرد دوم تخمین) داده می‌شود.

$$i^* = \underset{i}{argmax}(\rho(D_i)_{i,j}), \qquad i^* \neq j \qquad (۶)$$

که در آن j، اندیس صفت جاافتاده است.

خروجی گام حداکثرسازی در روش ۲ برای گام تخمین به‌طور مستقیم به تک مدل خطیِ چند صفته داده می‌شود. خروجی گام حداکثرسازی در روش ۳، یعنی $D_{\max}$، برای گام تخمین، به m مدل خطی تک صفته داده می‌شود و نتیجه‌ی نهایی از طریق روش میانگین وزن‌دار (نظرخواهی وزن‌دار) حاصل می‌شود (رویکرد اول تخمین).

۴- تحلیل پیچیدگی زمانی رویکردهای پیشنهادی

در این بخش، پیچیدگی زمانی رویکردهای پیشنهادی تحلیل می‌شود. فرض کنید مجموعه‌داده ورودی شامل n نمونه است. از این n نمونه p تای آن‌ها کامل و q تای آن‌ها شامل مقادیر جاافتاده است ($p+q=n$). همچنین فرض کنید هر نمونه شامل m ویژگی است. روش‌های مطرح‌شده شامل سه گام اصلی هستند. در گام اول، با استفاده از یک مدل جنگل تصادفی، صفات رتبه‌بندی می‌شوند. مرتبه زمانی این گام در بدترین حالت برابر با $O(T(mn\log n))$ است؛ که در آن T بیانگر تعداد درخت‌ها در مدل جنگل تصادفی است. در گام دوم (حداکثرسازی)، ابتدا زمانی برابر با $O(m.p)$ صرف عملیات انتخاب مجموعه پایه می‌شود. سپس، زمانی برابر با $O(m.p.p\log p)$ صرف اجرای الگوریتم‌های حداکثرسازی (الگوریتم‌های شکل ۲ و ۳) خواهد شد. در این محاسبات $O(p\log p)$، زمان محاسبه‌ی مقدار PCC است. سومین و آخرین گام، گام تخمین مقادیر جاافتاده است. در این مرحله روش رگرسیون خطی با پیچیدگی زمانی $O(m^۲.p)$ اجرا می‌شود. با در نظر گرفتن موارد گفته‌شده، در بدترین حالت، پیچیدگی الگوریتم‌های ارائه‌شده از مرتبه‌ی

$O(m.p^2.\log p + m^2.p)$ خواهد بود. با فرض این‌که $p >> m$ ، آنگاه پیچیدگی زمانی الگوریتم‌های پیشنهادی از مرتبه‌ی $O(m.p^2.\log p)$ خواهد بود. اگر فرض کنیم که در بدترین حالت $p \approx n$ ، آنگاه پیچیدگی الگوریتم‌های پیشنهادی از مرتبه‌ی $O(m.n^2 \log n)$ خواهند بود. در ادامه، پیچیدگی زمانی برای دیگر روش‌های مقایسه شده ذکر می‌شوند. پیچیدگی زمانی رویکرد جایگذاری با روش k نزدیک‌ترین همسایگان [۳۱] از مرتبه‌ی $O(nk+nm)$ است؛ که در آن k برابر با تعداد نزدیک‌ترین همسایگان است. پیچیدگی زمانی رویکرد جایگذاری با روش FCM [۲۶] از مرتبه‌ی $O(nmc^2)$ است؛ که در آن c برابر با تعداد خوشه‌ها در الگوریتم خوشه‌بندی c-means فازی است. برای روش‌های DMI [۱۵] و IARI [۱۴]، پیچیدگی زمانی از مرتبه‌ی $O(mn^2)$ است.

عملیات پیش‌پردازش تنها یک‌بار و پیش از عملیات اصلی داده‌کاوی انجام می‌شود. بنابراین پیچیدگی زمانی عملیات پیش‌پردازش در اولویت پایین‌تری نسبت به کیفیت این امر قرار می‌گیرد. بااین‌حال برای توصیف دقیق‌تر، پیچیدگی زمانی رویکردهای پیشنهادی در این بخش تحلیل شد.

۵- آزمایش‌ها و نتایج

در این قسمت، روش‌های مورد مقایسه با روش‌های پیشنهادی، مجموعه‌داده‌های استفاده‌شده برای آزمایش‌ها و معیارهای ارزیابی کارایی رویکردهای متفاوت بررسی می‌شوند. علاوه بر این، آزمایش‌هایی برای بررسی تأثیر پارامترهای p و θ بر روی نتایج تخمین نهایی انجام می‌شود. عملکرد رویکردهای پیشنهادی با پنج روش تخمین با استفاده از مقدار میانگین، تخمین با استفاده از k نزدیک‌ترین همسایگان [۳۱]، روش پُر کردن با خوشه‌بندی c-means فازی (FCM) [۲۶]، روش پُر کردن با درخت تصمیم (DMI) [۱۵] و جایگذاری با استفاده از رگرسیون افزایشیِ صفات (IARI) [۱۴] مقایسه شده است.

مهم‌ترین عامل تصادفی که ممکن است در کیفیت الگوریتم‌های جایگذاری مقادیر جاافتاده تأثیرگذار باشد، تعداد مقادیر جاافتاده در هریک از نمونه‌های جاافتاده است. به‌منظور حذف عامل تصادفی بودن در آزمایش‌ها، هر آزمایش ۱۰ بار تکرار شده است. میانگین این ۱۰ بار تکرار به‌عنوان نتیجه نهایی در این پژوهش گزارش‌شده است. کدهای مورداستفاده برای رویکردهای مختلف همگی با زبان برنامه‌نویسی پایتون و با کمک بعضی از بسته‌های یادگیری ماشین از مخزن Scikit-Learn [۳۲] پیاده‌سازی شده‌اند.

۵-۱- مشخصات آزمایش‌ها

روش‌های مختلف جایگذاریِ مقادیر جاافتاده بر روی پنج مجموعه‌داده از دنیای واقعی که در مخزن داده‌ی یادگیری ماشین UCI به‌طور عمومی در دسترس هستند، اعمال شدند. جدول (۲) مشخصات این مجموعه‌داده‌ها را نمایش می‌دهد. این مجموعه‌داده‌ها بدون مقادیر جاافتاده می‌باشند. در این مجموعه‌داده‌های کامل، ابتدا مقادیر جاافتاده به‌طور مصنوعی ایجاد شدند. سپس مقادیر جاافتاده با استفاده از روش‌های مختلف جایگذاری پُر شدند. ازآنجایی‌که مقادیر اصلی برای مقادیر جاافتاده‌ی ساخته‌شده معلوم است، بنابراین عملکرد رویکردهای مختلف با دانستن مقادیر واقعی قابل ارزیابی است.

برای این‌که امکان مقایسه نتایج بین رویکردهای مختلف وجود داشته باشد و مقایسه‌ها بامعنا باشند، مقادیر هر صفت از مجموعه‌داده‌ها با استفاده از نرمال‌سازی به روش نمره‌ی z[۳۱] [۱۹] نرمال شدند. این کار با محاسبه و جایگذاری هر مقدار x_i^j در معادله‌ی (۷) انجام می‌شود.

$$x_i^j{}_{(normalized)} = \frac{x_i^j - \overline{A_j}}{\sigma_{A_j}} \qquad (۷)$$

که در آن $\overline{A_j}$ و σ_{A_j} به ترتیب بیانگر مقادیر میانگین و انحراف معیار برای صفت A_j هستند.

به‌منظور ایجاد مقادیر جاافتاده در مجموعه‌داده‌ها بعضی صفات انتخاب می‌شوند و بعضی از مقادیر این صفات به مقدار تهی تغییر می‌یابند. برای این منظور ابتدا نیمی از صفات به‌طور تصادفی انتخاب می‌شوند. سپس در ۳۰٪، ۴۰٪، ۵۰٪ و ۶۰٪ از نمونه‌ها، مقادیر جاافتاده به‌طور تصادفی ایجاد می‌شود. بنابراین هر نمونه‌ی جاافتاده ممکن است به تعداد ۱ تا $m/2$ صفت جاافتاده داشته باشد. همه‌ی مقادیر جاافتاده مطابق با الگوی جاافتاده به‌طور تصادفی (MAR) ایجادشده‌اند. در تمامی روش‌ها، از مدل رگرسیون خطیِ ساخته‌شده از طریق برازش[۳۲] بهترین خط بر روی داده‌ها به کمک روش حداقل مربعات[۳۳] استفاده‌شده است.

از دو معیار شناخته‌شده به نام‌های ریشه میانگین مربعات خطا[۳۴] (RMSE) و ضریب تعیین[۳۵] (CoD) یا R^2 برای ارزیابی کیفیت روش‌های مختلف پُر کردن مقادیر جاافتاده در این مقاله استفاده‌شده است. شاخص RMSE نشان‌دهنده‌ی میانگین اختلافات مقادیر واقعی از مقادیر تخمین زده‌شده توسط مدل‌های جایگذاریِ مقادیر جاافتاده است. ضریب تعیین در تحلیل مدل‌های آماری برای ارزیابی این‌که یک مدل چقدر خوب خروجی‌های آینده را پیش‌بینی می‌کند، استفاده می‌شود. ضریب تعیین مقداری بین ۰ و ۱ دارد. مقادیر کمتر برای میانگینِ خطای مطلق و مقادیر بیش‌تر برای ضریب تعیین، بیانگر کیفیت بالاتر از تخمین مقادیر است.

معادلات (۸) و (۹) به ترتیب نحوه‌ی محاسبه‌ی معیارهای ریشه میانگین مربعات خطا و ضریب تعیین را نشان می‌دهند.

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(y_k - y_k^{'})^2} \qquad (۸)$$

$$CoD = 1 - \frac{\sum_{k=1}^{n}(y_k^{'} - y_k)^2}{\sum_{k=1}^{n}(y_k - \bar{y})^2} \qquad (۹)$$

که در آن‌ها y_k نشان‌دهنده مقدار واقعی، $y_k^{'}$ مقدار پیش‌بینی‌شده، $\bar{y}$ میانگین مقادیر مشاهده‌شده و n برابر با تعداد مقادیر تخمین زده‌شده است.

۵-۲- مقایسه عملکرد رویکردهای ارائه‌شده با دیگر روش‌ها

در این بخش آزمایش‌هایی به‌منظور مقایسه عملکرد رویکردهای پیشنهادی با دیگر روش‌های موجود انجام می‌شود. عملکرد روش‌های ارائه‌شده با پنج روش تخمین با استفاده از مقدار میانگین، تخمین با استفاده از k نزدیک‌ترین همسایگان (kNN) [۳۱]، روش پُر کردن با خوشه‌بندی c-means فازی (FCM) [۲۶]، روش پُر کردن با درخت تصمیم (DMI) [۱۵] و جایگذاری با استفاده از رگرسیون افزایشیِ صفات (IARI) [۱۴] مقایسه شده‌اند. جداول (۳) تا (۷) عملکرد هریک از روش‌های جایگذاری مقادیر جاافتاده مبتنی بر همبستگی پیشنهادی را در مقایسه با پنج روش دیگر با توجه به معیارهای RMSE و CoD برای درصدهای مختلف جاافتادگی در مجموعه‌داده‌های مختلف نشان می‌دهد. مواردی که رویکرد پیشنهادی بهتر از تمام دیگر روش‌ها عمل کرده است، به‌صورت کج و پررنگ نوشته‌شده‌اند. از جداول ذکرشده پیداست که با افزایش درصد جاافتادگی ایجادشده در مجموعه‌داده‌ها، خطای کلیِ تخمین در همه‌ی روش‌ها افزایش می‌یابد. به‌عبارت‌دیگر، تعدادِ بیش‌ترِ داده‌های کامل منجر به افزایش دقت در تخمین‌ها می‌شود. این پدیده منطقی است، چراکه با افزایش درصد جاافتادگی اطلاعات مفید بیش‌تری از دست می‌رود.

در همه‌ی مجموعه‌داده‌ها به‌غیراز مجموعه‌داده Haberman، روش IARI تقریباً در همه‌ی موارد از چهار روش DMI، روش میانگین، روش FCM و روش kNN بهتر عمل کرده است. از طرفی در اکثر موارد، عملکرد کلی روش‌های ارائه‌شده برای هر دو معیار مقایسه شده بهتر از IARI بوده است. بنابراین می‌توان گفت که به‌طورکلی روش‌های ارائه‌شده بهتر از دیگر روش‌های مقایسه شده عمل می‌کنند. قابل‌ذکر است که روش IARI به‌طور پیش‌فرض از برچسب دسته‌ی نمونه‌ها به‌عنوان متغیر پیش‌بینی کننده استفاده می‌کند.

در مجموعه‌داده Haberman اگرچه روش‌های IARI، DMI، FCM و kNN ضعیف‌تر از روش میانگین عمل کرده‌اند، بااین‌حال مشاهده می‌شود که روش‌های پیشنهادی کمی بهتر از روش میانگین عمل کرده‌اند. ضعف نسبی روش‌های دیگر نسبت به روش میانگین احتمالاً به دلیل تعداد کم متغیرهای پیش‌بینی کننده در این مجموعه‌داده است. برای مجموعه‌داده‌های Wholesale Customers، Iris و Wine، تمام روش‌های پیشنهادی در تمام حالات از دیگر روش‌ها با توجه به دو شاخص ارزیابی بهتر عمل کرده‌اند. برای مجموعه‌داده‌ی Glass در شرایطی که ۳۰٪ یا ۶۰٪ جاافتادگی در نمونه‌ها وجود دارد، در همه‌ی حالات، روش‌های پیشنهادی بهتر از روش IARI عمل کرده‌اند. روش‌های ۱، ۲ و ۳ هنگامی‌که ۴۰٪ و فقط روش ۲ هنگامی‌که ۵۰٪ جاافتادگی وجود دارد، ضعیف‌تر از روش IARI عمل کرده‌اند. بااین‌حال، در این حالات، روش‌های پیشنهادی بهتر از چهار روش DMI، FCM، kNN و روش میانگین عمل کرده‌اند.

به‌طورکلی مهم‌ترین عاملی که در کیفیت هریک از روش‌های پیشنهادی تأثیر می‌گذارد، نوع و میزان همبستگی موجود بین صفات در یک مجموعه‌داده است. برای روش‌های ۱ و ۲ که از روش حداکثر کردن همبستگی هریک از صفات استفاده می‌کنند، هنگامی‌که چند صفت با همبستگی قوی در یک مجموعه‌داده حضورداشته باشد، این روش‌ها بهتر از روش ۳ عمل می‌کنند. از طرف دیگر، اگر در یک مجموعه‌داده میانگین همبستگی بین صفات بالا باشد (حتی اگر چند صفتِ بسیار همبسته هم وجود نداشته باشند)، روش ۳ که از روش حداکثر کردن میانگین مقادیر همبستگی تمام صفات استفاده می‌کند، بهتر از روش‌های ۱ و ۲ عمل خواهد کرد. برای مثال در مجموعه‌داده‌های Glass، Wine و Haberman که می‌توان چند صفت با همبستگی بالا را یافت، در اکثر موارد دقت روش‌های ۱ و ۲ بیش‌تر از روش ۳ است. از طرف دیگر چون در مجموعه‌داده‌ی Wholesale Customers، چندین صفت با همبستگی بالا یافت نمی‌شود، دقت روش ۳ از روش‌های ۱ و ۲ بیش‌تر است. برای مجموعه‌داده‌ی Iris، چون همبستگی بین صفات هم به‌صورت تک‌به‌تک و هم به‌طور میانگین بالاست، گام تخمین نیز در عملکرد روش‌ها تأثیرگذار است. نکته‌ی مهم دیگرِ قابل‌مشاهده این است که به‌طورکلی کیفیت جایگذاری دو روش ۱ و ۲ بسیار نزدیک به هم است. یعنی گام حداکثرسازی در دقت جایگذاری تأثیر بیشتری نسبت به گام تخمین دارد.

همان‌طور که جداول نشان می‌دهند، در همه‌ی مجموعه‌داده‌ها به غیراز حالت ۴۰٪ جاافتادگی در مجموعه‌داده‌ی Glass، حداقل یکی از روش‌های پیشنهادی از باقی روش‌ها بهتر عمل کرده است. برای مجموعه‌داده‌ی Haberman، به دلیل کم بودن تعداد صفات پیش‌بینی کننده در این مجموعه‌داده (سه صفت)، عملکرد روش‌های پیشنهادی با اختلاف اندکی بهتر از دیگر روش‌ها بوده است. برای مجموعه‌داده‌ی Glass، اگرچه در ۴۰٪ جاافتادگی، روش‌های ارائه‌شده ضعیف‌تر از بهترین روش (IARI) عمل کرده‌اند، بااین‌حال برای درصدهای بالای جاافتادگی (۵۰٪ و ۶۰٪) روش‌های پیشنهادی بهتر از بهترین روش عمل کرده‌اند. بنابراین به نظر می‌رسد که عامل تصادفی بودن باعث شده است که فقط در حالت ۴۰٪ جاافتادگی از مجموعه‌داده‌ی Glass، روش‌های پیشنهادی ضعیف‌تر از روش IARI عمل کنند؛ چراکه در باقی حالات و مجموعه‌داده‌ها، روش‌های پیشنهادی بهتر از باقی روش‌ها عمل کرده‌اند. درمجموع روش‌های پیشنهادی بر روی مجموعه‌داده‌هایی با دو ویژگی، بهتر عمل خواهند کرد. اول این است که تعداد صفات یک مجموعه‌داده بسیار کم نباشد؛ چراکه اصولاً دقت روش‌های رگرسیون با افزایش تعداد صفات پیش‌بینی کننده افزایش می‌یابد. دوم این‌که اگر همبستگی بین صفات در یک مجموعه‌داده به‌صورت تک‌به‌تک و یا به‌صورت میانگین بالا باشد، دقت روش‌های پیشنهادی نیز بالاتر خواهد بود.

جدول ۳: مقایسه عملکرد رویکردهای ارائه‌شده با پنج روش دیگر برای مجموعه‌داده‌ی Wholesale Customers

		CoD					RMSE					
درصد جاافتادگی	شناسه روش	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI
۳۰%	۱	***۰/۹۷۷۴***	۰/۹۵۶۱	۰/۹۶۹۴	۰/۹۷۱۸	۰/۹۴۷۱	۰/۹۶۴۷	***۰/۱۴۲۷***	۰/۲۰۰۳	۰/۱۶۶۲	۰/۱۶۱۰	۰/۲۰۹۸	۰/۱۷۱۰
	۲	***۰/۹۷۷۰***						***۰/۱۴۴۲***					
	۳	***۰/۹۷۹۵***						***۰/۱۳۶۸***					
۴۰%	۱	***۰/۹۶۸۳***	۰/۹۴۰۸	۰/۹۵۷۵	۰/۹۶۰۴	۰/۹۲۵۷	۰/۹۵۴۷	***۰/۱۷۰۷***	۰/۲۳۵۳	۰/۱۹۷۹	۰/۱۹۱۸	۰/۲۵۹۲	۰/۲۰۱۱
	۲	***۰/۹۶۷۲***						***۰/۱۷۳۲***					
	۳	***۰/۹۷۲۳***						***۰/۱۶۰۳***					
۵۰%	۱	***۰/۹۵۸۲***	۰/۹۲۶۳	۰/۹۴۴۵	۰/۹۴۵۱	۰/۹۱۵۷	۰/۹۳۷۵	***۰/۱۹۸۶***	۰/۲۶۴۱	۰/۲۲۸۲	۰/۲۲۵۸	۰/۲۸۰۷	۰/۲۴۸۰
	۲	***۰/۹۵۸۱***						***۰/۱۹۸۱***					
	۳	***۰/۹۶۱۰***						***۰/۱۹۱۸***					
۶۰%	۱	***۰/۹۴۷۲***	۰/۹۰۵۷	۰/۹۳۱۵	۰/۹۳۱۸	۰/۸۹۲۳	۰/۹۲۴۹	***۰/۲۲۴۰***	۰/۲۹۹۳	۰/۲۵۴۵	۰/۲۵۲۸	۰/۳۲۱۱	۰/۲۶۴۵
	۲	***۰/۹۴۷۹***						***۰/۲۲۲۴***					
	۳	***۰/۹۵۱۴***						***۰/۲۱۵۷***					

جدول ۴: مقایسه عملکرد رویکردهای ارائه‌شده با پنج روش دیگر برای مجموعه‌داده‌ی Wine

		CoD					RMSE					
درصد جاافتادگی	شناسه روش	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI
۳۰%	۱	***۰/۹۸۷۴***	۰/۹۷۲۶	۰/۹۸۳۱	۰/۹۸۵۲	۰/۹۸۰۱	۰/۹۸۵۷	***۰/۱۱۰۱***	۰/۱۶۴۵	۰/۱۲۷۹	۰/۱۲۴۲	۰/۱۴۰۴	۰/۱۲۷۸
	۲	***۰/۹۸۹۷***						***۰/۱۰۰۴***					
	۳	***۰/۹۸۷۲***						***۰/۱۱۱۱***					
۴۰%	۱	***۰/۹۸۳۶***	۰/۹۶۴۸	۰/۹۷۷۹	۰/۹۸۱۵	۰/۹۷۳۰	۰/۹۷۸۶	***۰/۱۳۶۶***	۰/۱۸۶۶	۰/۱۴۶۸	۰/۱۴۴۶	۰/۱۶۳۲	۰/۱۵۱۷
	۲	***۰/۹۸۶۰***						***۰/۱۱۶۹***					
	۳	***۰/۹۸۳۲***						***۰/۱۳۷۲***					
۵۰%	۱	***۰/۹۷۸۰***	۰/۹۵۲۱	۰/۹۷۲۲	۰/۹۷۴۴	۰/۹۶۵۸	۰/۹۷۱۷	***۰/۱۴۵۶***	۰/۲۱۸۰	۰/۱۶۴۴	۰/۱۷۲۳	۰/۱۸۴۰	۰/۱۷۳۴
	۲	***۰/۹۸۰۴***						***۰/۱۳۸۴***					
	۳	***۰/۹۷۸۵***						***۰/۱۴۴۷***					
۶۰%	۱	***۰/۹۷۳۷***	۰/۹۴۱۷	۰/۹۶۶۹	۰/۹۶۸۵	۰/۹۵۷۲	۰/۹۶۵۹	***۰/۱۶۱۴***	۰/۲۴۰۶	۰/۱۷۹۶	۰/۱۸۸۸	۰/۲۰۶۴	۰/۱۸۹۳
	۲	***۰/۹۷۶۰***						***۰/۱۵۲۹***					
	۳	***۰/۹۷۵۷***						***۰/۱۵۵۸***					

جدول ۵: مقایسه عملکرد رویکردهای ارائه‌شده با پنج روش دیگر برای مجموعه‌داده‌ی Iris

		CoD					RMSE					
درصد جاافتادگی	شناسه روش	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI
۳۰%	۱	***۰/۹۸۱۷***	۰/۹۰۹۴	۰/۹۷۱۳	۰/۹۷۷۷	۰/۹۶۳۰	۰/۹۷۶۲	***۰/۱۲۸۸***	۰/۲۹۹۴	۰/۱۶۲۱	۰/۱۴۲۴	۰/۱۹۱۲	۰/۱۴۴۰
	۲	***۰/۹۷۹۷***						***۰/۱۳۶۰***					
	۳	***۰/۹۸۱۶***						***۰/۱۲۹۳***					
۴۰%	۱	***۰/۹۷۵۴***	۰/۸۷۹۳	۰/۹۶۲۹	۰/۹۷۰۵	۰/۹۵۱۴	۰/۹۶۵۴	***۰/۱۴۹۷***	۰/۳۴۵۷	۰/۱۸۵۴	۰/۱۶۳۸	۰/۲۱۹۸	۰/۱۶۵۴
	۲	***۰/۹۷۲۸***						***۰/۱۵۷۶***					
	۳	***۰/۹۷۵۹***						***۰/۱۴۸۰***					
۵۰%	۱	***۰/۹۶۹۱***	۰/۸۵۴۲	۰/۹۵۳۱	۰/۹۶۴۰	۰/۹۳۴۵	۰/۹۶۱۷	***۰/۱۶۸۸***	۰/۳۸۰۱	۰/۲۰۹۸	۰/۱۸۲۱	۰/۲۵۵۴	۰/۱۸۷۲
	۲	***۰/۹۶۷۲***						***۰/۱۷۴۷***					
	۳	***۰/۹۷۰۳***						***۰/۱۶۵۵***					
۶۰%	۱	***۰/۹۶۰۱***	۰/۸۲۸۶	۰/۹۴۰۱	۰/۹۵۴۳	۰/۹۱۴۴	۰/۹۵۰۰	***۰/۱۹۱۵***	۰/۴۱۱۹	۰/۲۳۵۵	۰/۲۰۲۹	۰/۲۹۲۲	۰/۲۰۳۴
	۲	***۰/۹۵۸۴***						***۰/۱۹۵۶***					
	۳	***۰/۹۶۱۰***						***۰/۱۸۸۷***					

جدول ۶: مقایسه عملکرد رویکردهای ارائه‌شده با پنج روش دیگر برای مجموعه‌داده‌ی Glass

		CoD						RMSE					
درصد جاافتادگی	شناسه روش	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI
۳۰%	۱	***۰/۹۸۶۷***	۰/۹۵۳۷	۰/۹۷۹۶	۰/۹۸۳۰	۰/۹۵۷۸	۰/۹۷۴۶	***۰/۱۰۸۵***	۰/۲۰۷۲	۰/۱۳۶۳	۰/۱۲۴۹	۰/۱۹۲۵	۰/۱۴۷۳
	۲	***۰/۹۸۳۱***						***۰/۱۲۳۰***					
	۳	***۰/۹۸۸۳***						***۰/۱۰۳۲***					
۴۰%	۱	۰/۹۷۳۹	۰/۹۴۱۸	۰/۹۶۱۳	۰/۹۷۸۴	۰/۹۴۱۹	۰/۹۵۷۴	۰/۱۴۷۳	۰/۲۳۴۳	۰/۱۸۱۸	۰/۱۴۱۵	۰/۲۲۸۴	۰/۱۵۶۱
	۲	۰/۹۷۱۸						۰/۱۵۷۸					
	۳	۰/۹۷۴۵						۰/۱۴۷۳					
۵۰%	۱	***۰/۹۷۳۰***	۰/۹۲۷۶	۰/۹۵۳۳	۰/۹۷۲۶	۰/۹۳۲۱	۰/۹۵۰۱	***۰/۱۵۸۸***	۰/۲۶۳۰	۰/۲۰۵۵	۰/۱۵۹۴	۰/۲۴۶۸	۰/۱۷۲۸
	۲	۰/۹۶۸۵						۰/۱۷۱۹					
	۳	***۰/۹۷۳۲***						***۰/۱۵۸۶***					
۶۰%	۱	***۰/۹۷۵۹***	۰/۹۱۵۵	۰/۹۴۷۸	۰/۹۶۷۶	۰/۹۲۰۰	۰/۹۴۳۶	***۰/۱۴۵۰***	۰/۲۸۵۱	۰/۲۱۹۵	۰/۱۷۵۶	۰/۲۶۹۰	۰/۱۹۶۴
	۲	***۰/۹۶۳۰***						***۰/۱۵۵۱***					
	۳	***۰/۹۷۵۳***						***۰/۱۴۶۸***					

۵-۳- تأثیر پارامترهای p و θ

در این بخش تأثیر پارامترهای ورودی بر روی کیفیت عملکرد رویکردهای ارائه‌شده بررسی می‌شود. همان‌طور که در بخش‌های پیش توضیح داده شد، در روش‌های ارائه‌شده، دو پارامتر اصلی p و θ نیاز به تنظیم دارند. پارامتر θ تعیین می‌کند که آیا یک رکورد به مجموعه‌ی پایه اضافه شود یا خیر. پارامتر p، درصدِ انتخاب از نزدیک‌ترین همسایگان کامل نسبت به هر نمونه‌ی جاافتاده را (به‌عنوان مجموعه‌ی پایه) تعیین می‌کند.

برای بررسی تأثیر پارامتر p، مقدار p برای مقدار ثابتِ θ بین ۳۰ تا ۶۰ با فواصل ۵ تغییریافته و مقدار RMSE برای هر مقدار p ثبت شد. شکل (۶) مقدار RMSE را برای مقادیر مختلف p بر روی مجموعه‌داده ی Iris با ۳۰٪ جاافتادگی نشان می‌دهد. نتایج برای باقی مجموعه‌داده‌ها و دیگر معیارها به دلیل کمبود فضا نشان داده نشده‌اند. برای روش ۱، عملکرد بهینه در p=۴۰ حاصل‌شده است. برای روش‌های ۲ و ۳ مقدار RMSE پس از رسیدن به مقدار بهینه روندی صعودی داشته است.

به‌طورکلی، نتایج نشان داده‌اند که مقادیر خیلی پایین و خیلی بالا برای p مناسب نیستند. مقادیر بسیار بالا، مفهوم انتخاب روبه‌جلو را بی‌معنا می‌کند. در مقابل، مقادیر بسیار پایین دید خوبی از روابط موجود در مجموعه پایه نخواهد داد.

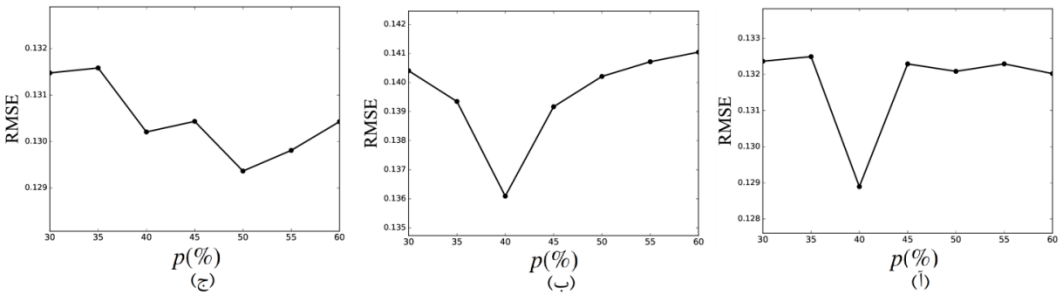
برای بررسی تأثیر پارامتر θ، مقدار θ برای یک مقدار ثابتِ p بین ۰/۰ تا ۱/۰ با فواصل ۰/۰۱ تغییریافته و مقدار RMSE برای هر مقدار θ ثبت شد. شکل (۷) مقدار RMSE را برای مقادیر مختلف θ بر روی مجموعه‌داده‌ی Iris با ۳۰٪ جاافتادگی نشان می‌دهد. مجدداً به دلیل کمبود فضا، نتایج برای باقی مجموعه‌داده‌ها و دیگر معیارها نشان داده نشده‌اند. نمودارها ثابت ماندن مقدار RMSE برای مقادیر بالای θ (وقتی θ به ۱ میل می‌کند) را نشان می‌دهند. این روند منطقی است چراکه در این شرایط الگوریتم‌ها به‌طور سخت‌گیرانه اجازه‌ی اضافه شدن رکوردهای غیرپایه را به مجموعه پایه خواهند داد.

مقدار کمینه سراسری برای روش‌های ۱ و ۳ در حدود آستانه نزدیک به صفر رخ‌داده‌اند. برای روش ۲، مقدار بهینه‌ی RMSE در $\theta \approx 1$ ظاهرشده است.

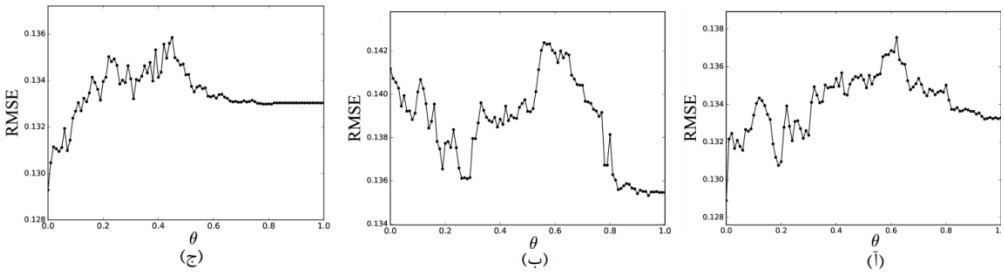
جدول ۷: مقایسه عملکرد رویکردهای ارائه‌شده با پنج روش دیگر برای مجموعه‌داده‌ی Haberman

		CoD						RMSE					
درصد جاافتادگی	شناسه روش	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI	روش پیشنهادی	روش میانگین	روش kNN	روش IARI	روش FCM	روش DMI
۳۰%	۱	***۰/۸۹۳۵***	۰/۸۸۴۲	۰/۸۳۱۴	۰/۸۶۸۲	۰/۸۸۲۲	۰/۸۵۲۵	***۰/۳۲۶۹***	۰/۳۳۵۹	۰/۴۱۱۶	۰/۳۶۳۴	۰/۳۴۱۳	۰/۳۷۳۷
	۲	***۰/۸۹۵۹***						***۰/۳۲۳۳***					
	۳	***۰/۸۹۳۵***						***۰/۳۲۶۹***					
۴۰%	۱	***۰/۸۵۶۴***	۰/۸۴۹۷	۰/۷۸۷۹	۰/۸۲۰۸	۰/۸۳۵۷	۰/۸۰۳۴	***۰/۳۸۰۱***	۰/۳۹۵۹	۰/۴۶۲۲	۰/۴۲۳۷	۰/۴۰۴۴	۰/۴۲۶۷
	۲	***۰/۸۶۱۱***						***۰/۳۷۴۰***					
	۳	***۰/۸۵۶۴***						***۰/۳۸۰۱***					
۵۰%	۱	***۰/۸۰۶۲***	۰/۷۹۸۶	۰/۷۳۶۱	۰/۷۷۳۰	۰/۸۰۴۳	۰/۷۶۳۷	***۰/۴۴۱۲***	۰/۴۴۷۳	۰/۵۱۶۴	۰/۴۷۶۶	۰/۴۴۱۶	۰/۴۸۵۲
	۲	***۰/۸۱۵۵***						***۰/۴۳۰۶***					

	۳	۰/۸۰۵۹					۰/۴۴۱۵				
۶۰%	۱	۰/۷۵۸۳	۰/۷۵۱۸	۰/۶۶۹۳	۰/۷۳۱۱	۰/۷۶۸۱	۰/۴۹۳۳	۰/۷۱۵۷	۰/۴۹۹۲	۰/۵۷۷۷	۰/۵۱۹۴	۰/۴۸۰۷	۰/۵۲۱۳
	۲	۰/۷۷۳۹					۰/۴۷۷۴						
	۳	۰/۷۵۸۳					۰/۴۹۳۳						



شکل ۶: تأثیر پارامتر p بر مقدار RMSE در مجموعه‌داده Iris برای (آ) روش ۱، (ب) روش ۲ و (ج) روش ۳



شکل ۷: تأثیر پارامتر θ بر مقدار RMSE در مجموعه‌داده Iris برای (آ) روش ۱، (ب) روش ۲ و (ج) روش ۳

۶- نتیجه‌گیری

در این مقاله سه روش جدید برای جایگذاری داده‌های جاافتاده ارائه شد. ویژگی اصلی و مشترک این روش‌ها تلاش برای حداکثر کردن همبستگی خطی بین صفت جاافتاده و دیگر صفات معلوم و سپس اعمال مدل رگرسیون است. نتایج بررسی‌شده بر روی پنج مجموعه‌داده‌ی مختلف نشان داد که رویکردهای پیشنهادی در اکثر موارد در مقایسه با پنج روش مختلف دیگر، خطای کمتری در تخمین دارند. تأثیر پارامترهای رویکرد پیشنهادی در میزان خطا نیز بررسی شدند. از مشکلات اصلی رویکرد پیشنهادی می‌توان به چالش انتخاب بهینه برای پارامترهای رویکرد پیشنهادی اشاره کرد. اگرچه این چالش نیز با عملیاتِ زمان‌برِ جستجوی کامل برای انتخاب بهترین ترکیب پارامترها قابل برطرف شدن است. بررسی تأثیر دیگر معیارهای سنجش همبستگی، مانند معیار اطلاعات متقابل[۳۶]، در کیفیت پُر کردن مقادیر جاافتاده و اعمال دیگر مدل‌های تخمین برای پژوهش‌های آینده در دست بررسی است.

مراجع

[1] Y. Qin, S. Zhang, X. Zhu, J. Zhang, and C. Zhang, "Pop algorithm: Kernel-based imputation to treat missing values in knowledge discovery from databases," Expert Systems with Applications, vol. 36, no. 2, pp. 2794–2804, 2009.

[۲] مرتضی خرّم کشکولی و مریم دهقانی، «تشخیص، شناسایی و جداسازی عیب توربین گاز پالایشگاه دوم پارس جنوبی با استفاده از روش‌های ترکیبی داده‌کاوی، k-means، تحلیل مؤلفه‌های اصلی (PCA) و ماشین بردار پشتیبان (SVM)»، مجله علمی پژوهشی مهندسی برق دانشگاه تبریز، جلد ۴۷، شماره ۲، صفحات ۵۰۱-۵۱۵، ۱۳۹۶.

[۳] علیرضا سردار و رمضان هاونگی، «بهبود عملکرد الگوریتم خوشه‌یابی خودکار تصاویر رنگی به کمک پیش‌پردازش با شبکه عصبی خودسامانده (SOM)»، مجله علمی پژوهشی مهندسی برق دانشگاه تبریز، جلد ۴۷، شماره ۳، صفحات ۱۰۸۲-۱۰۷۳، ۱۳۹۶.

[4] G. Sun, J. Shao, H. Han, and X. Ding, "Missing value imputation for wireless sensory soil data: A comparative study," in 2nd International Conference on Big Data Computing and Communications, pp. 172–184, Springer, Shenyang, China, 2016.

[5] M. Lichman, UCI Machine Learning Repository, Available online at: http://archive.ics.uci.edu/ml, Accessed June 2017.

[6] P. J. Garcia-Laencina, J. L. Sancho-Gomez, and AR. Figueiras-Vidal, "Pattern classification with missing data: a review," Neural Computing and Applications, vol. 19, no. 2, pp. 263–282, 2010.

[7] E. L. Silva-Ramrez, R. Pino-Mejas, and M. Lpez-Coello, "Single imputation with multilayer perceptron and multiple imputation combining multi-layer perceptron and k-nearest neighbours for monotone patterns," Applied Soft Computing, vol. 29, no. 1, pp. 65 – 74, 2015.

[8] M. G. Rahman and M. Z. Islam, "Missing value imputation using a fuzzy clustering-based EM approach," Knowledge and Information Systems, vol. 46, no. 2, pp. 389–422, 2016.

[9] M. Amiri and R. Jensen, "Missing data imputation using fuzzy-rough methods," Neurocomputing, vol. 205, no. 1, pp. 152–164, 2016.

[10] H. Wang and S. Wang, "Mining incomplete survey data through classification," Knowledge and information systems, vol. 24, no. 2, pp. 221–233, 2010.
[11] C.F. Tsai and F.Y. Chang, "Combining instance selection for better missing value imputation," Journal of Systems and Software, vol. 122, no. 1, pp. 63– 71, 2016.
[12] C. T. Tran, M. Zhang, P. Andreae, and B. Xue, "Improving performance for classification with incomplete data using wrapper-based feature selection," Evolutionary Intelligence, vol. 9, no. 3, pp. 81–94, 2016.
[13] M. G. Rahman and M. Z. Islam, "Data quality improvement by imputation of missing values," in 5th International Conference on Computer Science and Information Technology (CSIT-2013), pp. 82– 88, Yogyakarta, Indonesia, 2013.
[14] B. van Stein and W. Kowalczyk, "An incremental slgorithm for repairing training sets with missing values," in International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, vol. 611, no. 1, pp. 175–186. Springer International Publishing, Eindhoven, Netherlands, 2016.
[15] G. Rahman and Z. Islam, "A decision tree-based missing value imputation technique for data pre-processing," in Proceedings of the Ninth Australasian Data Mining Conference, vol. 121, no. 1, pp. 41-50. Australian Computer Society, Inc., 2011.
[16] L. Breiman, "Random Forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.
[17] C.H. Wu, C.H. Wun, and H.J. Chou, "Using association rules for completing missing data," in 4th International Conference on Hybrid Intelligent Systems, pp. 236–241, Kitakyushu, Japan, IEEE, 2004.
[18] N. Singh, A. Javeed, S. Chhabra, and P. Kumar, "Missing value imputation with unsupervised kohonen self organizing map," Emerging Research in Computing, Information, Communication and Applications, vol. 1, no. 1, pp. 61–76. Springer, New Delhi, India, 2015.
[19] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 3rd Edition, 2011.
[20] C. Jiang and Z. Yang, "CKNNI: an improved knn-based missing value handling technique," in 11th International Conference on Intelligent Computing, pp. 441–452, Springer, Fuzhou, China, 2015.
[21] R. Krishnamoorthy, S. S. Kumar, and B. Neelagund, "A new approach for data cleaning process," in Recent Advances and Innovations in Engineering (ICRAIE), pp. 1-5, IEEE, Jaipur, India, 2014.
[22] C. Zhang, X. Zhu, J. Zhang, Y. Qin, and S. Zhang, "GBKII: An imputation method for missing values," in Advances in Knowledge Discovery and Data Mining: 11th Pacific-Asia Conference, pp. 1080–1087, Springer, Nanjing, China, 2007.
[23] B. M. Patil, R. C. Joshi, and D. Toshniwal, "Missing value imputation based on k-mean clustering with weighted distance," in 3rd International Conference on Contemporary Computing, pp. 600–609, Springer, Noida, India, 2010.
[24] V. Ayuyev, J. Jupin, P. Harris, and Z. Obradovic, "Dynamic clustering-based estimation of missing values in mixed type data," in 11th International Conference on Data Warehousing and Knowledge Discovery, pp. 366–37, Springer, Linz, Austria, 2009.
[25] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Towards missing data imputation: A study of fuzzy k-means clustering Method," Rough Sets and Current Trends in Computing, vol. 3066, no. 1, pp. 573–579, Springer, 2004.
[26] P. Raja and K. Thangavel, "Soft clustering based missing value imputation," in Digital Connectivity-Social Impact: 51st Annual Convention of the Computer Society of India, pp. 119–133, Springer, Coimbatore, India, 2016.
[27] N. Ankaiah and V. Ravi, "A novel soft computing hybrid for data imputation," in 7th International Conference on Data Mining (DMIN), Las Vegas, USA, 2011.
[28] S. Azim, S. Aggarwal, "Hybrid model for data imputation: using fuzzy c-means and multi-layer perceptron," in Advance Computing Conference (IACC), 2014 IEEE International, pp. 1281–1285, Gurgaon, India, 2014.
[29] S. Bashir, S. Razzaq, U. Maqbool, S. Tahir, and A. R. Baig, "Using association rules for better treatment of missing values," in 10th WSEAS International Conference on Computers, Wisconsin, USA, pp. 1133-1138, 2009.
[30] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," Machine learning, vol. 38, no. 3, pp. 257–286, 2000.
[31] Batista, G. E., & Monard, M. C. (2002). A study of k-nearest neighbour as an imputation method. Hybrid Intell Syst (HIS), vol. 87, no. 1, pp. 251-260, 2002.
[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, no. 1, pp. 2825–2830, 2011.

زیرنویس‌ها

[1] Null
[2] Imputation
[3] Correlation
[4] Redundant
[5] Regression
[6] Incremental Attribute Regression Imputation
[7] Decision tree based Missing value Imputation technique
[8] Random Forest
[9] Association Rules
[10] Multi-Layer Perceptron
[11] Self Organizing Map
[12] Interpretability
[13] k Nearest Neighbors
[14] Grey
[15] Grey-Based K-NN Iteration Imputation
[16] Class-based K-clusters Nearest Neighbor Imputation
[17] Sparse
[18] Topology
[19] Decision tree based Missing value Imputation technique
[20] Expectation Maximization
[21] Instance Selection
[22] Noise Data
[23] Decremental Reduction Optimization Procedure
[24] Forward Selection
[25] Out of bag
[26] Predictor
[27] Response
[28] Pearson Correlation Coefficient
[29] Covariance
[30] Exhaustive Search
[31] z-score
[32] Fitting
[33] Least Square Method
[34] Root Mean Squared Error
[35] Coefficient of Determination
[36] Mutual Information