

# گزارش تبیین پذیری سیستم‌های نرم‌افزاری: از آنالیز نیازمندی‌ها تا ارزیابی سیستم

ملیکا محمدی گل

علیرضا سلطانی نشان  
a.soltani@iau-tnb.ac.ir

سودابه آشوری

۶ فروردین ۱۴۰۳

## فهرست مطالب

۳	۱	مقدمه
۳	۲	رویکرد تبیین پذیری
۳	۱.۲	تبیین پذیری چیست؟
۳	۲.۲	چالش های تبیین پذیری
۳	۱.۲.۲	پیچیدگی سیستم ها
۳	۲.۲.۲	طبیعت Black box
۳	۳.۲.۲	زمینه گرایی توضیح یا Subjectivity of Explanation
۴	۴.۲.۲	تریدآف همراه با تاثیرگذاری روی عملکرد یا Trade-off with Performance

دوران الگوریتمیک یا Algorithmic age منحصر به زمان کنونی و آینده جهان است، به گونه‌ای که الگوریتم‌های گوناگونی در زندگی انسان‌ها وارد شده‌اند. از ساده‌ترین عملیات گرفته تا پیچیده‌ترین تصمیم‌گیری‌ها را می‌توان پیاده‌سازی کرد. این الگوریتم‌ها حوزه‌های مختلف از قبیل اقتصاد، بهداشت، حمل و نقل و غیره را تحت تاثیر خود قرار داده است. از انتخاب کوتاه‌ترین مسیر رانندگی گرفته تا پیشبینی و تشخیص سرطان و طراحی پروتئین و داروهای مختلف.

این برگه به روشنی از قابلیت‌های این سیستم‌ها، که تصمیم‌گیری‌های مختلف را به گونه‌ای انجام می‌دهند که کاربر نمی‌داند که این پیشبینی چگونه رخ داده است یا این ماشین با چه داده‌هایی به چنین نتیجه‌ای رسیده است را Black box systems می‌نامد. زمانی که کاربر نمی‌تواند متوجه شود که پشت صحنه این پیشبینی‌ها به چه شکلی می‌باشد می‌گوییم که این سیستم‌ها فاقد شفافیت<sup>۱</sup> هستند. فاقد شفافیت نگرانی‌هایی را در رابطه با پارامترهایی مانند مسئولیت‌پذیری، انصاف و پیامدهای اخلاقی ایجاد می‌کند به ویژه در حوزه‌هایی که سیستم تصمیماتی را می‌گیرد که می‌تواند روی مردم و جامعه به طور کلی تاثیر گذار باشد.

بر همین اساس، این برگه بحث‌هایی را در مورد شفافیت و اخلاق سیستم‌های مدرن مطرح کرده است. درک چگونگی ادغام این نگرانی‌ها در سیستم‌ها و در نتیجه نحوه برخورد با آنها در طول مهندسی نرم‌افزار و مهندسی نیازمندی‌ها بسیار مهم است.

## ۲ رویکرد تبیین‌پذیری

رویکرد روش‌شناسی‌ای که این مقاله در مورد آن صحبت می‌کند تبیین‌پذیری در سیستم‌های نرم‌افزاری و حتی مدل‌های هوش مصنوعی است تا بتواند ضعف عدم شفافیت سیستم‌ها را رفع کند.

### ۱.۲ تبیین‌پذیری چیست؟

تبیین‌پذیری یک روش مفید است تا از نگرانی‌های اخلاقی نرم‌افزارها و مدل‌ها بکاهد. به معنای قابلیت شرح نرم‌افزار و سیستم است. وقتی یک سیستم یا مدل هوش مصنوعی تبیین‌پذیر است، به این معناست که عملکرد و تصمیمات آن قابل تفسیر و توجیه است. به عبارت دیگر، می‌توان به راحتی فهمید که یک سیستم به چه شکلی کار می‌کند و چگونه به تصمیمات خود رسیده است. تبیین‌پذیری یک ویژگی بسیار مهم در سیستم‌های نرم‌افزاری است که موجب افزایش اعتماد به آن می‌شود و ارزش‌های اخلاقی و قانونی را در رابطه با سیستم تعریف خواهد کرد. امروزه به مسئله تبیین‌پذیری سیستم‌ها بسیار اهمیت داده می‌شود و یکی از مهم‌ترین نیازمندی‌های Non-functional محسوب می‌شود. در حالی که به کاربران این اجازه را می‌دهد که خودشان بتوانند انتخاب کنند که از این سیستم استفاده کنند یا از آن دوری کنند چرا که بر روی رابطه قابلیت اعتماد و اتکای سیستم بسیار تاثیرگذار می‌باشد.

نکته: با توجه به قدرت هوش مصنوعی در تمام حوزه‌های زندگی بشر، تبیین‌پذیری به عنوان یکی از مهم‌ترین پایه‌های اعتماد در نیازمندی‌های نرم‌افزار می‌باشد. همچنین در این مقاله در مورد رابطه بین جنبه‌های کیفی و تبیین‌پذیری صحبت می‌شود.

### ۲.۲ چالش‌های تبیین‌پذیری

دلایل زیر نشان‌دهنده آن است که جمع‌آوری و استخراج داده، مذاکره و اعتبارسنجی در فرایند تبیین‌پذیری با چالش‌هایی رو به رو می‌باشد:

#### ۱.۲.۲ پیچیدگی سیستم‌ها

در سیستم‌هایی که مبنی بر هوش مصنوعی و فرایند یادگیری ماشین هستند با وجود الگوریتم‌های مختلف که وظیفه تصمیم‌گیری را در سیستم دارند، سطح پیچیدگی بسیار بالا می‌باشد. درک و توضیح این سیستم‌ها با فرایندهایشان برای کاربران مختلف به مفهوم ساده، بسیار سخت و غیرقابل درک می‌باشد.

#### ۲.۲.۲ طبیعت Black box

از نظر کاربران، بسیاری از الگوریتم‌ها به شکل جادویی عمل می‌کنند، بدان معنا که فرایندهای داخلی این الگوریتم‌ها کاملاً به صورت مات می‌باشد و توسط انسان بدون دانش قبلی به راحتی قابل درک نیست.

#### ۳.۲.۲ زمینه‌گرایی توضیح یا Subjectivity of Explanation

زمینه‌گرایی توضیح به معنای نسبی بودن یا وابستگی توضیحات به نگرش و دیدگاه فردی است. در حالت کلی تفسیر هر چیزی توسط ذینفعان می‌تواند کاملاً متفاوت از نظر معنا و دیدگاه باشد. مذاکره برای به اجماع رسیدن در سطح و نوع توضیح مورد نیاز می‌تواند چالش برانگیز باشد، به ویژه زمانی که با دیدگاه‌ها و علایق گوناگون سروکار داریم.

<sup>۱</sup>Transparency

## ۴.۲.۲ تریدآف همراه با تاثیرگذاری روی عملکرد یا Trade-off with Performance

گاهی افزایش تبیین پذیری در یک سیستم می تواند به قیمت عملکرد و کارایی تمام شود. یک مهندس نیازمندی باید بتواند بین تبیین پذیری با سایر الزامات سیستم System requirements تعادل ایجاد کند. برای درک این چالش مثال زیر را مطالعه کنید:

تصور کنید یک شرکت در حال توسعه سیستم توصیه گرا برای اپلیکیشن تجاری خود می باشد. این سیستم الگوریتم های پیچیده ML را برای تحلیل رفتارها و ترجیحات<sup>۲</sup> کاربران استفاده می کند تا بتواند محصولات مشابه علاقه مندی آنها را به نحوی معرفی کند که کاربران انتظار داشتند. یکی از نیازمندی های NFR این سیستم، ارائه توضیحات برای هر کدام از نتایج محصولات توصیه شده می باشد تا بتواند موجب اعتماد و رضایت کاربران شود. در این صورت گنجاندن توضیحات به همراه جزئیات چرایی انتخاب این مورد (محصول) به عنوان مورد مرتبط برای این سیستم تاثیر به سزایی در عملکرد آن خواهد داشت. این عمل باعث تاخیری در تولید این موارد برای کاربران می شود که از نظر تجربه کاربری<sup>۳</sup> یک ضعف محسوب می شود به ویژه زمانی که کاربران انتظار دارند که تمام تقاضاهایشان از سیستم در کمتر از پنج ثانیه پاسخ داده شود.

احتمالاً برای این مثال راهکارهای زیر در نظر گرفته می شود تا ضمن تبیین پذیری سیستم، عملکرد سیستم نیز مانند سابق با سرعت بالا حفظ شود:

۱. کاهش پیچیدگی توضیحات: به جای آنکه توضیحات کاملی در مورد عملکرد الگوریتم های هوش مصنوعی به ازای هر مورد فراهم شود، سیستم می تواند بسیار ساده با ارائه خلاصه ای مفید، فاکتورهای مهم و اساسی دلیل انتخاب موارد به عنوان توصیه کاربر را مشخص کند.

۲. استفاده از متدهای فنی در مهندسی نرم افزار مانند فرایندهای کش کردن انتخاب های کاربر (براساس کلیک های مختلف روی محصولات یا مدت زمانی که روی محصول مورد نظر کاربر مطالعه داشته) محاسبات از پیش تعیین شده ای در مورد چرایی انتخاب محصول به عنوان توصیه را مشخص کند.

انتخاب استراتژی مناسب برای حفظ تبیین پذیری به همراه سرعت و کارایی بالا در عملکرد سیستم توصیه گرا، یک تریدآفی است که وظیفه آن بر عهده تیم توسعه، طراح و معماری نرم افزار می باشد.

---

<sup>۲</sup> Preferences  
<sup>۳</sup> User experience