

Amirkabir University of Technology
(Tehran Polytechnic)

Department of Industrial Engineering & Management Systems

JANOS: An Integrated Predictive and Prescriptive Modeling Framework

David Bergman, Teng Huang, Philip Brooks, Andrea Lodi,
Arvind U. Raghunathan

By:

Mahdi Mohammadi – 402125059

Leila Salimi - 401125023

Instructor:

Dr. Pooya Hoseinpour

Data Driven Modeling - Final Project

Fall Semester 2023

فهرست مطالب

۱- معرفی ژورنال.....	۲
۲- تشریح مقاله.....	۲
۱-۲- مقدمه.....	۲
۲-۲- مسئله تخصیص بودجه کمیته پذیرش دانشگاه.....	۲
۳-۲- چهارچوب کلی مدل سازی.....	۳
۴-۲- جزئیات الگوریتمی.....	۵
۱-۴-۲- مقدمه.....	۵
۲-۴-۲- بهینه سازی روی شبکه های عصبی.....	۶
۳- پیاده سازی و بررسی نتایج.....	۹
۱-۳- چهارچوب JANOS.....	۹
۲-۳- پیاده سازی در Pyomo.....	۹
۳-۳- الگوریتم ابتکاری.....	۱۰
۴-۳- بررسی نتایج عددی.....	۱۰
۱-۴-۳- مجموعه داده.....	۱۰
۲-۴-۳- آزمون برابری میانگین ها.....	۱۱
۳-۴-۳- بررسی یکسان بودن نتیجه مدل پیاده سازی شده و مدل اصلی.....	۱۱
۴-۴-۳- تاثیر solverهای مختلف شبکه عصبی.....	۱۱
۵-۴-۳- صحت سنجی نتایج.....	۱۱
پی نوشت.....	۱۳

۱- معرفی ژورنال

این مقاله در ژورنال INFORMS Journal On Computing Volume 34, Issue 2 در تاریخ ۲۸ سپتامبر ۲۰۲۱ منتشر شده و تاکنون ۲۶ بار توسط سایر مقالات مورد ارجاع قرار داده است.

ژورنال INFORMS Journal On Computing مقالاتی را در تقاطع حوزه‌های تحقیق در عملیات (OR) و علوم کامپیوتر (CS) منتشر می‌کند. H-index این ژورنال در حال حاضر ۸۵ است و این ژورنال، جزو ژورنال‌های Q₁ توسط SJR محسوب می‌شود.^۱

۲- تشریح مقاله

۲-۱- مقدمه

در سال‌های اخیر، گسترش قابل توجهی در تحقیقات مربوط به کاربرد یادگیری ماشین و بهینه‌سازی گسسته صورت گرفته است. این دو حوزه تحلیلی اغلب در یک مسئله تصمیم‌گیری تجاری اما برای اهداف متفاوت استفاده شده‌اند. تکنیک‌های یادگیری ماشین معمولاً برای پیش‌بینی آنچه در آینده اتفاق می‌افتد استفاده می‌شوند و از روش‌های بهینه‌سازی برای جستجوی هدفمند راه‌حل‌های امکان‌پذیر استفاده می‌شود. این مقاله با معرفی یک چهارچوب^۲ مدل‌سازی برای تحلیل مشترک پیش‌بینی-تجویزی با نام JANOS قصد دارد این دو حوزه را تلفیق کرده و گپ تحقیقاتی موجود را پوشش بدهد.

در ادامه ابتدا یک مسئله عددی را شرح می‌دهیم سپس با کمک این مسئله مفاهیم کلی توضیح داده شده مقاله را بازگو خواهیم کرد.

۲-۲- مسئله تخصیص بودجه کمیته پذیرش دانشگاه

دفتر پذیرش یک دانشگاه می‌خواهد به دانشجویان کاندیدای خود بورسیه‌هایی ارائه دهد و هدف آن‌ها این است که طوری بودجه خود را به دانشجویان کاندید خود تخصیص بدهند که تعداد مورد انتظار دانشجویان آینده آن‌ها را به حداکثر برساند.

دفتر پذیرش از سال‌های ثبت نام قبلی، SAT، GPA، پیشنهاد ارائه شده به کاندیداها و نتایج تصمیم آن‌ها را جمع آوری کرده است (یعنی اینکه آیا دانشجو این پیشنهاد را پذیرفته است یا خیر). امسال، فرض کنید دانشگاه N دانشجوی کاندیدا دارد. علاوه بر این، فرض کنید بودجه کمیته برای بورسیه‌های ارائه شده $0.2 \times N \times 10^4$ دلار

¹ <https://www.scimagojr.com/journalsearch.php?q=25040&tip=sid>

² Framework

است که از این پس با BUDGET به آن اشاره می‌شود. مبلغ بورسیه‌هایی که می‌توان به هر متقاضی خاص اختصاص داد بین ۰ تا ۲۵۰۰۰ دلار است.

۲-۳- چهارچوب کلی مدل سازی

مسائلی مانند مسئله عددی ارائه شده نیاز به چهارچوبی دارند که بتواند به خوبی در دل روش‌های بهینه‌سازی از مدل‌های پیش‌بینی کننده استفاده کند و یک تجویز مناسب را برای مسئله مورد بررسی ارائه دهد.

مطالعات موجود در مورد ترکیب تحلیل‌های پیش‌بینی کننده و تجویزی را می‌توان در قالب جدول ۱ خلاصه‌سازی کرد.

جدول ۱- خلاصه مرور ادبیات

شرح	روش	پژوهش
زمانی که شمارش کامل یا شمارش جزئی مجموعه راه‌حل‌های امکان‌پذیر عملی باشد، امکان‌پذیر است	پیش‌بینی‌های ثابت به عنوان پارامتر در یک مدل بهینه سازی	Ferreira et al. (2015) Huang et al. (2019)
عدم ارتباط مستقیم با موضوع مورد بررسی	استفاده از یادگیری ماشین برای بهبود بهینه‌سازی و برعکس	Boyd et al. (2011), Koh et al. (2007), Cozad et al. (2014), Bienstock et al. (2018) Cappart et al. (2018), Khalil et al. (2017), Bengio et al. (2018), Nazari et al. (2018), Lemos et al. (2019)
در نظر گرفتن عدم قطعیت در پیش‌بینی‌ها	برآورد با مدل‌های یادگیری ماشین به طوری که پشیمانی از تصمیم‌ها را به حداقل می‌رساند	Elmachtoub and Grigas (2017), Demirovic et al. (2019), Mandi et al. (2019), Wilder et al. (2019), Bertsimas and Kallus (2020)
-	بهینه سازی بر روی مدل‌های یادگیری نظارت شده	Paul et al. (2018), Misić (2017), Biggs et al. (2017), Aouad et al. (2019), Feldman et al. (2019)
جستجو در مجموعه کوچک از مقادیر ممکن کارا نبودن الگوریتم ابتکاری ارائه شده	بهینه‌سازی بر ساختار شبکه‌های عصبی	Lam et al. (2000)
پیچیدگی محاسباتی بالا	مسئله بهینه سازی سراسری با یک شبکه عصبی تعبیه شده با تابع فعال‌سازی tanh	Schweidtmann and Mitsos (2019)

درواقع تا پیش از چهارچوب ارائه شده در این مقاله، هیچ راه حل کارایی (به صورت یک چهارچوب) در این زمینه پیاده سازی و ارائه نشده است.

مجدداً به مسئله ارائه شده در بخش ۲-۲ باز می گردیم. برای حل این مسئله بهینه سازی با استفاده از JANOS، می توان مدلی را برای پیش بینی احتمال پذیرش یک پیشنهاد با توجه به SAT، GPA و بورس تحصیلی برای یک دانشجو توسعه داد. متغیری که باید برای آن تصمیم گیری شود، سومین ویژگی برای هر دانشجو است: میزان بورسیه ای که به هر دانشجو ارائه می شود.

مسئله را می توانیم به صورت زیر مدل می کنیم:

$$\begin{aligned} \max \sum_{i=1}^N y_i & \quad (STUDENT - ENROLL) \\ s.t. \sum_{i=1}^N x_i & \leq BUDGET & (1) \\ y_i = g(s_i, g_i, x_i; \theta) & \quad \forall i \in \{1, \dots, N\} & (2) \\ 0 \leq x_i \leq 25000 & \quad \forall i \in \{1, \dots, N\} & (3) \end{aligned}$$

که در آن

- x_i متغیر تصمیم است، یعنی مقدار بورسیه اختصاص داده شده به هر متقاضی
- s_i امتیاز SAT متقاضی i است
- g_i نمره معدل متقاضی i است
- y_i یک متغیر پیش بینی شده برای هر دانشجوی متقاضی است که نتیجه یک مدل پیش بینی شده است و نشان دهنده احتمال پذیرش پیشنهاد توسط متقاضی است.
- g یک مدل از پیش آموزش داده شده است که برای پیش بینی احتمال پذیرش یک پیشنهاد توسط متقاضیان از قبل آموزش داده شده است. پارامترهای s_i ، g_i و x_i ورودی های مدل پیش بینی کننده هستند و بردار θ پارامترهای یاد گرفته شده توسط این مدل را نشان می دهد (که فرض می کنیم برای هر متقاضی یکسان است). تابع g می تواند هر یک از مدل های پیش بینی مجاز باشد که در ادامه درباره آن ها توضیحاتی ارائه خواهد شد.

به صورت کلی تر JANOS به دنبال حل یک مسئله فرموله شده به عنوان (PROBLEM-ORI):

$$\max_x \sum_{j=1}^{n_1} c_j x_j + \sum_{k=1}^{n_2} d_k y_k \quad (PROBLEM - ORI)$$

S.t.

$$\sum_{j=1}^{n_1} a_{ij}^i x_j \leq b_i \quad \forall i \in \{1, \dots, m\} \quad (4)$$

$$y_k = g_k(\alpha_1^k, \dots, \alpha_{p_k}^k; \theta_k) \quad \forall k \in \{1, \dots, n_2\} \quad (5)$$

$$\alpha_l^k \text{ is given} \quad \begin{matrix} \forall l \in \{1, \dots, q_k\} \\ \forall k \in \{1, \dots, n_2\} \end{matrix} \quad (6)$$

$$\alpha_l^k = e_l^k \cdot x \quad \begin{matrix} \forall l \in \{q_k + 1, \dots, p_k\} \\ \forall k \in \{1, \dots, n_2\} \end{matrix} \quad (7)$$

$$x_j \in X_j \quad \forall j \in \{1, \dots, n_1\} \quad (8)$$

متغیرهای $x = (x_1, \dots, x_{n_1})$ متغیرهای منظم^۳ و $y = (y_1, \dots, y_{n_2})$ متغیرهای پیش بینی شده هستند. هر متغیر x_j به یک مجموعه محدود یا پیوسته X_j تعلق دارد و از طریق نابرابری های خطی محدود می شود. هر متغیر پیش بینی شده y_k با یک مدل پیش بینی کننده g_k ، با ویژگی های $(\alpha_1^k, \dots, \alpha_{p_k}^k)$ همراه است. هر g_k یک مدل پیش بینی کننده از قبل آموزش دیده (رگرسیون خطی، رگرسیون لجستیک، یا شبکه عصبی با تابع فعال سازی خطی اصلاح شده) است و فرض می شود که پارامترهای θ_k قبل از بهینه سازی توسط کاربر به طور مناسب انتخاب شده اند. q_k ویژگی اول مدل از پیش آموزش داده شده، ثابت و داده شده هستند؛ در حالی که هر یک از ویژگی های باقی مانده $(p_k - q_k)$ ویژگی هایی هستند که از طریق تابع e_l^k ، به متغیر منظم مرتبط، پیوند داده شده اند. توجه داشته باشید که در تعریف فوق می توان از یک مدل از پیش آموزش داده شده g بجای چند g_k استفاده شود.

۲-۴- جزئیات الگوریتمی

۲-۴-۱- مقدمه

در حال حاضر چهارچوب JANOS از ۳ الگوریتم رگرسیون خطی، رگرسیون لجستیک و شبکه عصبی با تابع فعال ساز ReLU^4 برای مدل های پیش بینی کننده، پشتیبانی می کند.

³ Regular

⁴ Rectified Linear Unit (ReLU)

شبکه‌های عصبی ابزار قدرتمندی هستند که می‌توانند الگوهای پیچیده داده‌ها را با تکرار زیاد کشف کنند و در صورت وجود قدرت محاسباتی کافی و حجم مناسب داده بهترین گزینه برای مدل‌های پیش‌بینی کننده هستند. با توجه به این نکته و نتایج عددی مقاله که این موضوع را تایید می‌کند، ما فقط نحوه استفاده از این الگوریتم را در ادامه تشریح خواهیم کرد.

۲-۴-۲- بهینه‌سازی روی شبکه‌های عصبی

اگر \mathcal{V} مقدار پیش‌بینی شده یک شبکه عصبی باشد، این مقدار از یک مدل جریان شبکه به دست می‌آید.

برای هر متغیر پیش‌بینی شده \mathcal{V} که توسط یک شبکه عصبی تعیین می‌شود، یک مدل متمایز جریان شبکه را مرتبط می‌کنیم؛ درواقع، یک شبکه عصبی را می‌توان به عنوان یک گراف جهت‌دار لایه‌ای غیرحلقوی تک-ترمینال چند منبع، مشاهده کرد $N(V, A)$ که مجموعه رئوس آن V ، به l زیر مجموعه که هر زیر مجموعه نشان دهنده یک لایه در شبکه است، افراز شده است.

یک تابع یک به یک α میان ورودی‌ها و رئوس لایه اول $v \in V_1$ وجود دارد. برای هر $v \in V_1$ ، ورودی مرتبط با آن را با نماد $\alpha(v)$ نمایش خواهیم داد. V_l متشکل از راس ترمینال است که آن را t می‌نامیم.

برای $j \in \{2, \dots, l\}$ ، هر راسی $u \in V_j$ در حین آموزش، یک مقدار بایاس $B(u)$ دارد.

هر یال $a = (u, v) \in A$ از راس u در لایه V_j به راس v در لایه V_{j+1} به ازای بعضی از $j \in \{1, \dots, l-1\}$ متصل شده‌اند. هر یال یک وزن $w(a)$ دارد که در حین آموزش، آن مقدار برای آن تعیین شده است.

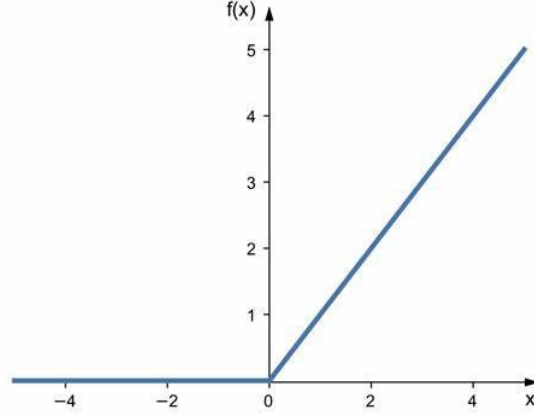
با توجه به مقادیر $\alpha(v)$ برای همه گره‌های $v \in V_1$ ، پیش‌بینی یک شبکه عصبی با تابع فعال‌سازی $ReLU$ به صورت بازگشتی با اختصاص یک مقدار F_v به همه گره‌ها در شبکه عصبی از طریق روش تکراری زیر، F_t محاسبه می‌شود:

$$\begin{aligned} F_v &= G_v = \alpha(v) & \forall v \in V_1 \\ G_v &= \sum_{u \in V_{j-1}} w((u, v))F_u + B(v), & \forall j \in \{2, \dots, l-1\}, \quad \forall v \in V_j \\ F_v &= \max\{0, G_v\} \\ F_v &= G_v = \sum_{u \in V_{l-1}} (w((u, v))F_u + B(u)) \end{aligned}$$

⁵ Rectified Linear Unit

در اینجا G_v ورودی تابع $ReLU$ و F_v خروجی تابع $ReLU$ است و ضابطه این تابع به صورت زیر است:

$$f(x) = \max(0, x)$$



شکل ۱- نمودار تابع $ReLU$

$\forall u \in V$ ، z_u را به عنوان یک متغیر باینری تعریف می کنیم که هرگاه مقدار $G_u > 0$ ، مقدار ۱ بگیرد. با این تفسیر، می توان مدل زیر ($MOD - NN$) را برای محاسبه y بر اساس ورودی های $\alpha(v)$ ، $\forall v \in V_1$ ، مدل کرد که هرکدام ویژگی هایی هستند که می توانند ثابت یا تابعی از متغیرهای تصمیم گیری باشند.

$$y = F_t \quad (MOD - NN)$$

$$G_v = \sum_{u \in V_{j-1}} w((u, v)) \cdot F_u + B(u) \quad \forall v \in V_1 \cup V_l \quad (9)$$

$$G_v = \alpha(v) \quad \forall v \in V_1 \quad (10)$$

$$\forall j \in \{2, \dots, l\}, \forall v \in V_1 \quad (11)$$

$$-M \cdot (1 - z_v) \leq G_v \leq M \cdot z_v \quad \forall v \in \bigcup_{i=2}^{l-1} V_i \quad (12)$$

$$G_v - M \cdot (1 - z_v) \leq F_v \leq G_v + M \cdot z_v \quad \forall v \in \bigcup_{i=2}^{l-1} V_i \quad (13)$$

$$0 \leq F_v \leq M \cdot z_v \quad \forall v \in V \quad (14)$$

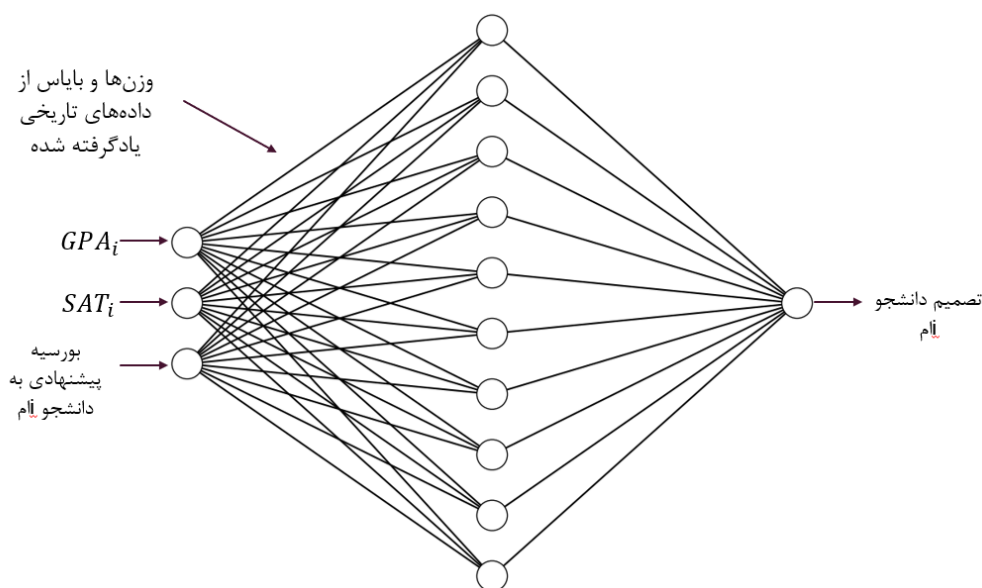
$$z_v \in \{0, 1\} \quad \forall v \in V \quad (15)$$

$$G_v \text{ unconstrained} \quad \forall v \in V \quad (16)$$

$$F_v \text{ unconstrained} \quad \forall v \in V \quad (17)$$

محدودیت (۹) تضمین می‌کنند که تابع فعال‌سازی $ReLU$ در لایه‌های ورودی و خروجی اعمال نمی‌شود. محدودیت (۱۰) اجبار می‌کند که مقادیر گره‌ها در لایه ورودی، مقادیر هر متغیر ورودی مدل پیش‌بینی‌کننده است، این مقادیر یا ثابت هستند یا مقداری هستند که توسط مدل بهینه‌سازی برای یک متغیر مسئله تعیین می‌شوند. قید (۱۱) ورودی تابع $ReLU$ را برای هر گره‌ای که در لایه اول نیست محاسبه می‌کند. محدودیت‌های (۱۲)–(۱۴) تابع فعال‌سازی $ReLU$ را در هر لایه پنهان اعمال می‌کنند.

در $(PROBLEM - ORI)$ ، $\alpha(v)$ می‌تواند یک متغیر منظم یا یک ثابت باشد و یک فرمول جریان شبکه جداگانه برای هر یک از متغیرهای پیش‌بینی شده که نتایج شبکه‌های عصبی هستند وجود خواهد داشت. برای فهم این موضوع مجدداً به مثال بخش ۲-۲ باز می‌گردیم. به ازای هر دانشجو ما یک جریان شبکه مثل شکل ۲ خواهیم داشت:



شکل ۲- شبکه مربوط به دانشجو نام

مجموع بودجه پیشنهادی به دانشجویان باید در محدودیت بودجه صدق کند (رجوع شود به بخش ۲-۳ گزارش مدل Student-Enrollment) و هدف حداکثر کردن امید ریاضی تعداد دانشجویان است. در ادامه به نحوه پیاده‌سازی و نتایج عددی می‌پردازیم.

۳- پیاده‌سازی و بررسی نتایج

۳-۱- چهارچوب JANOS

مدل‌های پیش‌بینی در Python3.6.9 با استفاده از scikit-learn 0.22.2 ساخته شدند و تمام مدل‌های بهینه‌سازی با Gurobi Optimizer v9.0.2 حل شدند. داده‌ها و نتایج در مخزن^۶ گیت‌هاب این چهارچوب در دسترس هستند.^۷ دریافت کتابخانه مربوط به چهارچوب به سادگی از pypi قابل انجام است. کد مربوط به این بخش به همراه توضیحات مربوط به آن، در نوت‌بوک ضمیمه شده در بخش JANOS قابل مشاهده است.

این مقاله جدول زیر را به عنوان نتایج ارائه کرده است:

جدول ۲- نتایج ارائه شده در مقاله

تعداد داوطلبان	Heuristic	JANOS_Discrete	JANOS_Continuous	Expected % Reduction in Declination	
				JANOS_Discrete vs. Heuristic	JANOS_Continuous vs. JANOS_Discrete
500	243.28	299.67	311.86	21.97 %	6.08 %
1000	467.44	580.66	606.57	21.26 %	6.18 %

با توجه به اینکه Solver استفاده شده برای توسعه این مدل، برای $N > 25$ نیاز به لایسنس دارد و دریافت لایسنس تحصیلی آن در ایران ممکن نیست مسئله را مجدداً با Pyomo مدل کردیم تا بتوانیم آن را با Solverهای متن باز حل کنیم. در بخش ۳-۴-۳ یکسان بودن نتایج حاصل از نسخه پیاده‌سازی شده را با نسخه اصلی کتابخانه بررسی کردیم.

۳-۲- پیاده‌سازی در Pyomo

Pyomo یک کتابخانه متن باز مدل‌سازی است که در زبان پایتون توسعه داده شده است. با کمک این کتابخانه می‌توانیم مدل ریاضی مسائل را در زبان پایتون پیاده‌سازی کنیم سپس با هر solver دلخواهی مسئله را حل کنیم. برای حل مدل ریاضی از COIN-OR Branch-and-Cut solver استفاده کردیم که در موارد تجربی نشان داده شده از glpk سریعتر عمل می‌کند.^۸

^۶ Repository

^۷ <https://github.com/INFORMSJoC/2020.1023>

^۸ <https://github.com/google/or-tools/issues/1522>

کد و جزئیات مربوط به آن در در نوتبوک ضمیمه شده در بخش Implemented JANOS قابل مشاهده است.

اگر ساختار این مسئله را در نظر بگیریم تعداد متغیرهای این مسئله از رابطه زیر پیروی می کند:

$$N \times 3(3 + l \times \#neuron_l + 1)$$

برای N بزرگ حل این مسئله $MILP$ بسیار زمانبر خواهد شد.

۳-۳- الگوریتم ابتکاری

نویسندگان مقاله برای اینکه نشان بدهند عملکرد مدل آن ها کارا است یک الگوریتم ابتکاری ارائه کردند و نشان دادند مدل آن ها از الگوریتم ابتکاری بهتر عمل می کند.

الگوریتم ابتکاری بدین صورت عمل می کرد:

۱. محاسبه مقدار $g(s_i, g_i, 25000) - g(s_i, g_i, 0)$ را برای همه ی دانشجویان محاسبه کنید و آن ها را به صورت غیر صعودی مرتب کنید.

۲. شروع به اختصاص ۲۵۰۰۰ دلار به افراد این لیست کنید تا جایی که بودجه اجازه می دهد.

این الگوریتم در واقع بورسیه را به افرادی که بیشترین حساسیت را نسبت به بورسیه دریافتی دارند، اختصاص می دهد.

پیاده سازی این الگوریتم را در قسمت Heuristic نوتبوک ضمیمه شده می توانید مشاهده کنید.

۳-۴- بررسی نتایج عددی

۳-۴-۱- مجموعه داده

داده از مخزن گیت هاب مقاله دریافت شده و در مقاله نوشته شده این مجموعه داده به صورت تصادفی تولید شده است. به همین جهت نیازی به فاز پیش پردازش نبوده و صرفا همانطور که در مقاله هم نوشته شده، SAT و GPA با استفاده از رابطه زیر استاندارد سازی شده اند.

$$x'_{ij} = \frac{x_{ij} - \bar{x}_{.j}}{\sigma_{x.j}} \quad j \in \{SAT, GPA\}$$

مقدار merit که نشان دهنده مقدار بورسیه است نیز به صورت پیش فرض با روش اعشاری استاندارد سازی شده است (مقادیر اولیه بر 10^4 تقسیم شده است).

در مقاله اشاره شده پس از آزمایش کردن روش های مختلف به این ترکیب روش ها دست یافته اند.

مجموعه داده آموزش ۲۰۰۰۰ ردیف و مجموعه آزمون ۶۰۰۰ ردیف داده است.

۳-۴-۲- آزمون برابری میانگین‌ها

باتوجه به اینکه در هربار نمونه‌ای به اندازه N از مجموعه داده آزمون گرفته می‌شود، نتایج هربار اجرا تغییر می‌کند. در واقع ما باید بررسی کنیم این تغییر ناشی از randomness است و یا اختلاف معنی‌دار است. این مسئله در آمار سالیان سال بررسی شده است و برای حالت‌های مختلف آزمون‌های مختلفی وجود دارد. در اینجا ما دو نمونه داریم که تعداد آن‌ها الزاماً برابر نیست و واریانس جوامع هم الزاماً برابر نمی‌باشد. در چنین مواردی باید از $Welch's t$ -test استفاده کرد که درواقع یک نسخه توسعه یافته از t -test عادی برای چنین مواردی است. برای اطلاعات بیشتر به (Welch, 1951) مراجعه کنید.

۳-۴-۳- بررسی یکسان بودن نتیجه مدل پیاده‌سازی شده و مدل اصلی

کد مربوط به این آزمون را می‌توانید در بخش Testها مشاهده کنید. مقدار p-value این آزمون برابر ۰.۸۲ شد که بدین معناست که فرض صفر را در سطوح معنی داری مرسوم نمی‌توان رد کرد. فرض صفر تست برابری میانگین جوامع است. پس درواقع می‌توانیم فرض کنیم این دو مدل یکسان اند.

۳-۴-۴- تاثیر solverهای مختلف شبکه عصبی

برای حالتی که بیش از ۲ جامعه داشته باشیم نمی‌توانیم از تست ۳-۴-۲ استفاده کنیم. اگر بخواهیم برابری میانگین آن‌ها را آزمون کنیم می‌توانیم از آنالیز واریانس یکطرفه استفاده کنیم. برای اینکه نیاز نباشد مفروضات کلاسیک آنالیز واریانس را بررسی کنیم از آزمون ناپامتری Kruskal-Wallis استفاده می‌کنیم (برای این آزمون صرفاً کفایت نمونه به اندازه کافی بزرگ باشد). مقدار p-value این آزمون برای حالت‌هایی که از Adam تنظیم شده، Adam با تنظیمات پیش فرض، SGD و lbfgs برای آموزش شبکه عصبی استفاده شود، برابر ۰.۰۰۰۱ شد که بدین معناست که فرض صفر (برابری میانگین جوامع) در سطوح معنی دار مرسوم، ۰.۰۰۰۱ رد می‌شود. اگر میزان صحت تست^۹ مدل این حالت‌ها را بررسی کنیم، میزان صحت SGD کمتر از بقیه است. با حذف SGD از جوامع مورد بررسی p-value به ۰.۷ می‌رسد و نمی‌توانیم فرض صفر را در این حالت رد کنیم.

با توجه به این مشاهده بسیار مهم است که وضعیت صحت نتایج شبکه عصبی پیش از استفاده در مسئله بهینه‌سازی به خوبی بررسی شود.

۳-۴-۵- صحت سنجی نتایج

تصمیم هر دانشجو را می‌توان با یک متغیر تصادفی برنولی مدل کرد:

^۹ Test Accuracy

$$Y_i \sim \text{Bern}(y_i)$$

امید ریاضی تعداد دانشجویان را می‌توان به صورت زیر ساده کرد:

$$E\left[\sum_{i=1}^N Y_i\right] = \sum_{i=1}^N E[Y_i] = \sum_{i=1}^N y_i$$

در ساده‌سازی این رابطه از خاصیت خطی بودن امید ریاضی استفاده شده است.

طبق اصل اول احتمال باید $0 \leq y_i \leq 1$ در حالی که هیچ محدودیتی بر راس نهایی اعمال نشده و ممکن است مقادیر بدست آمده برای y_i در این اصل صدق نکنند. شکل ۳ نیز تایید عددی این مورد را نشان می‌دهد.

```
'predicted': [1.0,
1.02,
1.0,
0.0,
1.0,
0.96,
1.0,
1.0,
0.99,
1.0,
-0.0,
0.0,
0.01,
1.0,
0.01,
0.96,
0.0,
1.0,
-0.0,
0.01]]
```

شکل ۳- نمونه نتیجه پیش‌بینی شده توسط چهارچوب

در واقع برای مسائل کلاس‌بندی باید محدودیتی بر راس ترمینال اعمال شود که در صورتی که مقدار آن از ۱ بزرگتر شد به ۱ و در صورتی که مقدار آن کوچکتر از صفر شد به صفر تبدیل شود.

در خصوص حالت کوچکتر از صفر می‌توانیم از محدودیت‌های ۱۲ تا ۱۴ استفاده کنیم و این محدودیت‌ها را به لایه‌ی آخر هم بسط دهیم. در خصوص حالت بزرگتر از یک باید مدلسازی به نحوی انجام شود که همچنان مسئله خطی باقی بماند در غیر اینصورت این چهارچوب ارزش عملی ندارد.

پی‌نوشت

- نسخه ویندوزی cbc solver هنوز پایدار نیست به همین جهت به صورت محلی این کد روی ویندوز اجرا نمی‌شود پس ترجیحاً کد را در گوگل کولب ویا سیستم عامل‌های بر پایه لینوکس اجرا کنید.
- دیتاست مربوط به پروژه به صورت پیش‌فرض با اجرای کد مربوط به آن دانلود و لود می‌شود در صورتی که لینک‌ها از کار افتاد از دیتاست‌های ضمیمه شده استفاده کنید.
- تمامی فایل‌های پروژه در ریپازیتوری ایجاد شده آن وجود دارد و به زودی نسخه کامل این مدل با قابلیت استفاده solverهای متن باز به صورت یک پکیج در پایتون منتشر خواهد شد.^{۱۰}

¹⁰ Repo Link: <https://github.com/MastersMasterM/DataDrivenModelingProject>