**پیش‌بینی ریزش مشتری در مخابرات با استفاده از یادگیری ماشین روی پلتفرم کلان‌داده**

**چکیده**

ریزش مشتری (Customer churn) یک مشکل اساسی و از مهم‌ترین نگرانی‌ها برای شرکت‌های بزرگ است. به دلیل تأثیر مستقیم آن بر درآمد شرکت‌ها، مخصوصاً در حوزه مخابرات، شرکت‌ها به دنبال توسعه روش‌هایی هستند تا مشتریانی را که احتمال دارد شرکت را ترک کنند پیش‌بینی کنند. بنابراین، شناسایی عواملی که ریزش مشتری را افزایش می‌دهند اهمیت دارد تا اقدامات لازم برای کاهش این ریزش انجام شود.

مشارکت (نوآوری) اصلی کار ما، توسعه یک مدل پیش‌بینی ریزش است که به اپراتورهای مخابراتی کمک می‌کند مشتریانی را که به احتمال زیاد در معرض ریزش هستند، پیش‌بینی کنند. مدلی که در این کار توسعه داده شده از تکنیک‌های یادگیری ماشین روی یک پلتفرم کلان‌داده استفاده می‌کند و یک شیوهٔ جدید برای مهندسی و انتخاب ویژگی ارائه می‌دهد.

برای سنجش عملکرد مدل، معیار استاندارد «مساحت زیر منحنی» (AUC) استفاده شده است و مقدار AUC به‌دست‌آمده برابر با ۹۳٫۳٪ است. یکی دیگر از مشارکت‌های اصلی، استفاده از شبکهٔ اجتماعی مشتریان در مدل پیش‌بینی است، از طریق استخراج ویژگی‌های «تحلیل شبکه اجتماعی» (Social Network Analysis یا SNA). استفاده از SNA عملکرد مدل را در معیار AUC از ۸۴٪ به ۹۳٫۳٪ بهبود داد.

مدل در محیط Spark آماده‌سازی و آزموده شد؛ با استفاده از یک مجموعه‌دادهٔ بزرگ که از تبدیل داده‌های خام عظیم ارائه‌شده توسط شرکت مخابراتی SyriaTel ساخته شد. این مجموعه‌داده شامل تمام اطلاعات مشتریان طی ۹ ماه بود و برای آموزش، آزمون و ارزیابی سامانه در SyriaTel به کار رفت. در این مدل چهار الگوریتم امتحان شدند: درخت تصمیم (Decision Tree)، جنگل تصادفی (Random Forest)، درخت گرادیان بوست‌شونده (Gradient Boosted Machine Tree یا GBM) و «بوستینگ گرادیان افراطی» (Extreme Gradient Boosting یا XGBOOST). بهترین نتایج با الگوریتم XGBOOST به‌دست آمد. این الگوریتم برای انجامِ دسته‌بندی در مدل پیش‌بینی ریزش ما به‌کار گرفته شد.

**واژه‌های کلیدی:**  
پیش‌بینی ریزش مشتری، ریزش در مخابرات، یادگیری ماشین، انتخاب ویژگی، دسته‌بندی، تحلیل شبکه اجتماعی موبایل، کلان‌داده

**مقدمه**

بخش مخابرات به یکی از صنایع اصلی در کشورهای توسعه‌یافته تبدیل شده است. پیشرفت فنی و افزایش تعداد اپراتورها سطح رقابت را بالا برده است [1]. شرکت‌ها برای بقا در این بازار رقابتی، با تکیه بر راهبردهای مختلف، تلاش زیادی می‌کنند. سه راهبرد اصلی برای افزایش درآمد پیشنهاد شده است [2]:

1. جذب مشتریان جدید،
2. فروش بیش‌تر به مشتریان فعلی (upsell)،
3. افزایش دورهٔ نگهداشت مشتری (retention period).

با این حال، مقایسهٔ این راهبردها با درنظرگرفتن بازگشت سرمایه (RoI) هرکدام نشان داده است که راهبرد سوم سودآورترین است [2]؛ این موضوع ثابت می‌کند که حفظ یک مشتری فعلی بسیار ارزان‌تر از جذب یک مشتری جدید است [3] و علاوه بر آن، عموماً بسیار ساده‌تر از راهبرد افزایش فروش (upselling) است [4].

برای اجرای راهبرد سوم، شرکت‌ها باید احتمال ریزش مشتری را کاهش دهند. ریزش مشتری به‌صورت «جابجایی مشتری از یک ارائه‌دهنده به ارائه‌دهندهٔ دیگر» تعریف می‌شود [5].

ریزش مشتری یک نگرانی جدی در صنایع خدماتیِ با رقابت شدید است. از طرف دیگر، پیش‌بینی مشتریانی که احتمال دارد شرکت را ترک کنند، اگر در مرحلهٔ اولیه انجام شود، می‌تواند یک منبع بالقوهٔ بسیار بزرگ برای درآمد اضافی باشد [3].

بسیاری از پژوهش‌ها تأیید کرده‌اند که فناوری یادگیری ماشین برای پیش‌بینی این وضعیت، بسیار کارآمد است. این تکنیک از طریق یادگیری از داده‌های گذشته به کار گرفته می‌شود [6، 7].

داده‌های استفاده‌شده در این پژوهش شامل تمام اطلاعات مشتریان در طول نه ماه پیش از «خط مبنا» (baseline) است. حجم این مجموعه‌داده حدود ۷۰ ترابایت روی HDFS («سیستم فایل توزیع‌شدهٔ هادوپ») است و شامل قالب‌های مختلف داده است: ساخت‌یافته، نیمه‌ساخت‌یافته و بدون ساختار. این داده‌ها همچنین با سرعت بالا تولید می‌شوند و به یک پلتفرم کلان‌دادهٔ مناسب برای مدیریت نیاز دارند. این مجموعه‌داده برای استخراج ویژگی‌های هر مشتری تجمیع شد.

ما شبکهٔ اجتماعی تمام مشتریان را ساختیم و ویژگی‌هایی مانند اندازه‌گیری‌های مرکزیت درجه (degree centrality)، مقادیر شباهت، و میزان اتصال شبکهٔ هر مشتری را برای هر مشتری محاسبه کردیم. ویژگی‌های SNA باعث بهبود محسوس در نتایج AUC شدند، و دلیل آن این است که این ویژگی‌ها اطلاعات متفاوت و اضافی دربارهٔ مشتریان فراهم می‌کنند.

ما روی ارزیابی و تحلیل عملکرد مجموعه‌ای از روش‌ها و الگوریتم‌های یادگیری ماشین مبتنی بر درخت، برای پیش‌بینی ریزش در شرکت‌های مخابراتی تمرکز کردیم. ما چند الگوریتم شامل درخت تصمیم، جنگل تصادفی، «درخت گرادیان بوست‌شونده» (GBM)، و «درخت XGBoost» را برای ساخت مدل پیش‌بینی ریزش مشتری پس از توسعهٔ آماده‌سازی داده، مهندسی ویژگی، و روش‌های انتخاب ویژگی امتحان کردیم.

در سوریه دو شرکت مخابراتی وجود دارد: SyriaTel و MTN. شرکت SyriaTel به این حوزهٔ تحقیقاتی علاقه‌مند بود زیرا جذب یک مشتری جدید شش برابر بیش‌تر از نگه‌داشتن یک مشتری در آستانهٔ ریزش هزینه دارد. مجموعه‌داده‌ای که SyriaTel فراهم کرد با چالش‌های زیادی همراه بود. یکی از این چالش‌ها نامتوازن بودن (unbalance) داده بود، جایی که کلاس مشتریان ریزشی خیلی کوچک‌تر از کلاس مشتریان فعال بود. ما برای برخورد با مسئلهٔ نامتوازن بودن، سه سناریو را امتحان کردیم: oversampling (بزرگ‌نمایی با تکرار کلاس اقلیت)، undersampling (کوچک‌کردن کلاس اکثریت) و بدون باز-متعادل‌سازی. برای ارزیابی از معیار «مساحت زیر منحنی مشخصه عملکرد گیرنده» (AUC یا Area Under ROC Curve) استفاده شد، چون این معیار عمومی است و در حالت مجموعه‌داده‌های نامتوازن به‌کار می‌رود [8].

تلاش‌های پیشین مختلفی با استفاده از سیستم «انبار داده» (Data Warehouse) برای کاهش نرخ ریزش در SyriaTel انجام شده بود. Data Warehouse بعضی انواع دادهٔ مخابراتی مثل دادهٔ صورتحساب، تماس‌ها/SMS/اینترنت و شکایات را تجمیع می‌کرد. سپس تکنیک‌های داده‌کاوی (Data Mining) روی Data Warehouse اعمال می‌شد، اما مدل با این داده‌ها نتوانست نتایج بالایی بدهد. در مقابل، منابع داده‌ای که خیلی بزرگ بودند نادیده گرفته شدند، چون کار با آن‌ها پیچیده بود. Data Warehouse نمی‌توانست به‌طور هم‌زمان حجم عظیم داده را دریافت، ذخیره و پردازش کند. علاوه بر این، منابع داده از انواع متفاوت بودند و ادغام آن‌ها در Data Warehouse یک فرایند بسیار دشوار بود، به‌طوری که افزودن ویژگی‌های جدید برای الگوریتم‌های داده‌کاوی زمان طولانی، توان پردازشی بالا، و فضای ذخیره‌سازی بیش‌تر لازم داشت. از طرف دیگر، تمام این فرایندهای دشوار، در یک پلتفرم کلان‌داده با پردازش توزیع‌شده، به‌سادگی قابل انجام است.

علاوه بر این، شبکه‌های اجتماعی بزرگ (مثل آنچه در SyriaTel وجود دارد) یکی از اجزای بنیادین گراف‌های کلان‌داده محسوب می‌شوند [9]. پیچیدگی محاسباتی شاخص‌های تحلیل شبکه اجتماعی (SNA) بسیار بالا است، چون محاسبات تکرارشونده روی یک گراف در مقیاس بزرگ انجام می‌شوند، همان‌طور که در روابط (۱) و (۲) اشاره شده است. تلاش‌های زیادی برای کاهش پیچیدگی محاسبهٔ شاخص‌های SNA انجام شده است. برای مثال، Barthelemy [10] یک الگوریتم جدید پیشنهاد کرد تا پیچیدگیِ محاسبهٔ «بینابینی بودن» (Betweenness centrality) را از مرتبهٔ O(n³) به O(n²) کاهش دهد. Elisabetta [11] نیز یک روش تقریب‌زنی پیشنهاد داد تا بینابینی با پیچیدگی کمتر محاسبه شود. با وجود این، سیستم سنتی Data Warehouse هنوز در محاسبهٔ شاخص‌های اساسی SNA روی شبکه‌های بزرگ‌مقیاس با کمبود و ناتوانی روبه‌رو است.

سامانهٔ کلان‌داده به شرکت SyriaTel اجازه داد تا داده را بدون توجه به حجم، تنوع و پیچیدگی آن به‌راحتی جمع‌آوری، ذخیره، پردازش و تجمیع کند. علاوه بر این، استخراج ویژگی‌های غنی‌تر و متنوع‌تری مثل ویژگی‌های SNA را ممکن کرد که اطلاعات اضافی برای بهبود مدل پیش‌بینی ریزش فراهم می‌کنند.

ما معتقدیم که کلان‌داده فرایند مهندسی ویژگی را ــ که یکی از سخت‌ترین و پیچیده‌ترین مراحل در ساخت مدل‌های پیش‌بینی است ــ ساده‌تر کرده است. با استفاده از پلتفرم کلان‌داده، ما این توان را به شرکت SyriaTel می‌دهیم که فراتر از منابع دادهٔ بزرگ برود. علاوه بر این، شرکت می‌تواند ویژگی‌های تحلیل شبکه اجتماعی را از یک گراف اجتماعی در مقیاس بزرگ استخراج کند؛ گرافی که از میلیاردها یال (تراکنش) ساخته شده که میلیون‌ها گره (مشتری) را به هم متصل می‌کنند. سخت‌افزار و طراحی پلتفرم کلان‌داده (که در بخش «روش پیشنهادی پیش‌بینی ریزش» توضیح داده شده) متناسب با نیاز محاسبهٔ این ویژگی‌هاست، بدون توجه به پیچیدگی‌شان روی این گراف بزرگ‌مقیاس.

مدل همچنین با استفاده از یک مجموعه‌دادهٔ جدید ارزیابی شد و اثر این سیستم بر تصمیمِ ریزش بررسی شد. مدل نتایج خوبی داد و در محیط عملیاتی (production) مستقر شد.

**کارهای مرتبط (Related work)**

رویکردهای زیادی برای پیش‌بینی ریزش در شرکت‌های مخابراتی اعمال شده است. بیشتر این رویکردها از یادگیری ماشین و داده‌کاوی استفاده کرده‌اند. عمدهٔ کارهای مرتبط یا فقط یک روش داده‌کاوی را برای استخراج دانش اعمال کرده‌اند، یا روی مقایسهٔ چند راهبرد برای پیش‌بینی ریزش تمرکز کرده‌اند.

Gavril و همکاران [12] یک روش پیشرفتهٔ داده‌کاوی برای پیش‌بینی ریزش مشتریان اعتباری (prepaid) ارائه کردند، با استفاده از یک مجموعه‌داده شامل جزئیات تماس (Call details) برای ۳۳۳۳ مشتری و ۲۱ ویژگی، و یک متغیر وابستهٔ ریزش با دو مقدار بله/خیر. بعضی ویژگی‌ها شامل اطلاعاتی دربارهٔ تعداد پیام‌های ورودی و خروجی و پیام‌صوتی (voicemail) هر مشتری بودند. نویسنده برای کاهش ابعاد داده از الگوریتم تحلیل مؤلفه‌های اصلی (PCA) استفاده کرد. برای پیش‌بینی عامل ریزش سه الگوریتم یادگیری ماشین به کار رفت: شبکه‌های عصبی، ماشین بردار پشتیبان، و شبکه‌های بیزی. برای سنجش عملکرد الگوریتم‌ها از AUC استفاده شد. مقادیر AUC به‌ترتیب برای شبکه‌های بیزی، شبکه‌های عصبی و ماشین بردار پشتیبان برابر با ۹۹٫۱۰٪، ۹۹٫۵۵٪ و ۹۹٫۷۰٪ بود. مجموعه‌دادهٔ استفاده‌شده در این مطالعه کوچک بود و هیچ مقدار گمشده‌ای نداشت.

He و همکاران [13] مدلی مبتنی بر الگوریتم شبکهٔ عصبی برای حل مشکل ریزش مشتری در یک شرکت مخابراتی بزرگ چینی با حدود ۵٫۲۳ میلیون مشتری پیشنهاد کردند. معیار دقت پیش‌بینی، «نرخ دقت کلی» بود و به مقدار ۹۱٫۱٪ رسید.

Idris [14] رویکردی مبتنی بر «برنامه‌نویسی ژنتیک همراه با AdaBoost» برای مدل‌سازی مسئلهٔ ریزش در مخابرات پیشنهاد کرد. مدل روی دو مجموعه‌دادهٔ استاندارد آزموده شد: یکی مربوط به شرکت Orange Telecom و دیگری مجموعهٔ cell2cell، با دقت ۸۹٪ روی cell2cell و ۶۳٪ روی مجموعهٔ دیگر.

Huang و همکاران [15] مسئلهٔ ریزش مشتری را روی یک پلتفرم کلان‌داده بررسی کردند. هدف پژوهشگران این بود که نشان بدهند کلان‌داده، با اتکا به حجم، تنوع و سرعت داده، فرایند پیش‌بینی ریزش را به‌شدت بهبود می‌دهد. کار با داده‌های بخش «پشتیبانی عملیات» و بخش «پشتیبانی کسب‌وکار» در بزرگ‌ترین شرکت مخابراتی چین، برای مهندسی ویژگی‌ها به یک پلتفرم کلان‌داده نیاز داشت. الگوریتم Random Forest استفاده شد و با AUC ارزیابی شد.

Makhtar و همکاران [16] مدلی برای پیش‌بینی ریزش با استفاده از «نظریه مجموعه زبر» (rough set theory) در مخابرات پیشنهاد دادند. طبق این مقاله، الگوریتم دسته‌بندی Rough Set عملکردی بهتر از الگوریتم‌های دیگر مانند رگرسیون خطی، درخت تصمیم، و «شبکه عصبی پرسپترون رأی‌گیری‌شده» داشته است.

پژوهش‌های مختلفی مسئلهٔ مجموعه‌داده‌های نامتوازن را بررسی کرده‌اند؛ جایی که کلاس مشتریان ریزشی کوچک‌تر از کلاس مشتریان فعال است. این موضوع یک چالش اصلی در مسئلهٔ پیش‌بینی ریزش است. Amin و همکاران [17] شش تکنیک مختلف نمونه‌برداری بیش‌ازحد (oversampling) را در زمینهٔ مسئلهٔ پیش‌بینی ریزش در مخابرات مقایسه کردند. نتایج نشان داد که الگوریتم‌های MTDF و تولید قانون مبتنی بر الگوریتم ژنتیک از دیگر الگوریتم‌های oversampling مقایسه‌شده بهتر عمل کردند.

Burez و Van den Poel [8] مسئلهٔ نامتوازن‌بودن مجموعه‌داده‌ها در مدل‌های پیش‌بینی ریزش را بررسی کردند و عملکرد «نمونه‌برداری تصادفی»، «نمونه‌برداری هشیارانهٔ پیشرفته» (Advanced Undersampling)، «مدل گرادیان بوستینگ»، و «جنگل تصادفی وزن‌دار» را مقایسه کردند. آنها از معیارهای AUC و Lift برای ارزیابی مدل استفاده کردند. نتیجه نشان داد که تکنیک undersampling از بقیهٔ تکنیک‌های آزموده‌شده بهتر عمل کرده است.

ما هیچ پژوهشی پیدا نکردیم که این مسئله را در یک شرکت مخابراتی در سوریه ثبت کرده باشد. بیشتر مقالات قبلی، مرحلهٔ مهندسی ویژگی را انجام نداده‌اند یا ویژگی‌ها را مستقیماً از دادهٔ خام نساخته‌اند؛ بلکه به ویژگی‌های آماده که یا توسط شرکت مخابراتی فراهم شده یا در اینترنت منتشر بوده تکیه کرده‌اند.

در این مقاله، مرحلهٔ مهندسی ویژگی جدی گرفته شده تا ویژگی‌های خودمان را برای استفاده در الگوریتم‌های یادگیری ماشین بسازیم. ما داده را با استفاده از یک پلتفرم کلان‌داده آماده‌سازی کردیم و نتایج چهار الگوریتم یادگیری ماشین مبتنی بر درخت را با هم مقایسه کردیم.

**مجموعه‌داده (Data set)**

در SyriaTel انواع متعددی از داده برای ساخت مدل ریزش استفاده شد. این انواع به شکل زیر دسته‌بندی می‌شوند:

1. **دادهٔ مشتری**  
   شامل تمام داده‌های مرتبط با خدمات مشتری و اطلاعات قرارداد است. همچنین شامل تمام طرح‌ها، بسته‌ها و سرویس‌هایی است که مشتری در آن‌ها عضو شده. علاوه بر این، اطلاعاتی که از سیستم CRM (مدیریت ارتباط با مشتری) تولید می‌شود را نیز دربر می‌گیرد، مثل:
   * تمام شماره‌های GSM مشتری،
   * نوع اشتراک،
   * تاریخ تولد،
   * جنسیت،
   * محل زندگی،  
     و غیره.
2. **پایگاه‌دادهٔ دکل‌ها و شکایات**  
   اطلاعات مربوط به محل انجام هر عمل به‌صورت عددی ثبت می‌شود. نگاشت این اعداد به پایگاه‌دادهٔ دکل‌ها (towers) محل آن تراکنش را به‌دست می‌دهد: طول و عرض جغرافیایی، زیربخش، بخش، شهر و استان.  
   پایگاه‌دادهٔ شکایات شامل تمام شکایات ثبت‌شده و استعلام‌های آماری مربوط به پوشش شبکه، مشکلات در طرح‌ها و بسته‌ها، و هر مشکل مرتبط با کسب‌وکار مخابراتی است.
3. **دادهٔ لاگ شبکه**  
   شامل نشست‌های داخلی مرتبط با اینترنت، تماس و پیامک برای هر تراکنش در اپراتور مخابراتی است؛ مثل مدت‌زمان لازم برای باز کردن یک نشست اینترنت، و وضعیت پایان تماس. این داده می‌تواند نشان بدهد آیا نشست به‌دلیل خطا در شبکهٔ داخلی قطع شده یا نه.
4. **رکوردهای جزئیات تماس (CDRs)**  
   شامل تمام اطلاعات شارژ مربوط به تماس‌ها، SMS، MMS و تراکنش‌های اینترنتی انجام‌شده توسط مشتریان است. این منبع داده به‌صورت فایل‌های متنی تولید می‌شود.
5. **اطلاعات IMEI موبایل**  
   شامل برند، مدل، نوع گوشی و این‌که دستگاه دو سیم‌کارته است یا تک‌سیم.

این داده‌ها بسیار حجیم‌اند و جزئیات فراوانی دارند. ما زمان زیادی صرف کردیم تا آن را بفهمیم و منابع و قالب ذخیره‌سازی آن را بشناسیم. علاوه بر این رکوردها، داده باید به داده‌های جزئی موجود در پایگاه‌داده‌های رابطه‌ای که اطلاعات دقیق دربارهٔ مشتری دارند متصل می‌شد. مجموعه‌دادهٔ مربوط به این ۹ ماه شامل حدود ده میلیون مشتری بود. تعداد کل ستون‌ها حدود ده‌هزار ستون بود.

**کاوش داده و چالش‌ها در مجموعه‌دادهٔ SyriaTel**

برای بررسی ساختار این مجموعه‌داده از موتور Spark استفاده شد. لازم بود این مرحلهٔ کاوش (Exploration) و آماده‌سازی اولیه انجام شود تا مجموعه‌داده برای الگوریتم‌های دسته‌بندی مناسب شود. بعد از بررسی داده‌ها، متوجه شدیم که حدود ۵۰٪ از همهٔ متغیرهای عددی فقط یک یا دو مقدار گسسته دارند، و تقریباً ۸۰٪ از همهٔ متغیرهای دسته‌ای (Categorical) کمتر از ۱۰ دسته دارند. ۱۵٪ از متغیرهای عددی و ۳۳٪ از متغیرهای دسته‌ای فقط یک مقدار دارند. بیشتر مقدارهای بعضی متغیرها نزدیک به صفر هستند. فهمیدیم که ۷۷٪ از متغیرهای عددی در بیش از ۹۷٪ سطرهایشان مقدار ۰ یا تهی (null) دارند. این نتایج نشان می‌دهد که تعداد زیادی از متغیرها را می‌توان حذف کرد چون این متغیرها ثابت‌اند یا تقریباً ثابت. این مجموعه‌داده با چند چالش روبه‌رو است که در ادامه می‌آید.

**۱. حجم داده**

چون از قبل معلوم نبود کدام ویژگی‌ها (Features) برای پیش‌بینی ریزش مفید هستند، مجبور بودیم روی تمام داده‌هایی که رفتار کلی مشتری را نشان می‌دهند کار کنیم. ما مجموعه‌داده‌های مربوط به تماس، SMS، MMS و اینترنت را با تمام اطلاعات مرتبط مثل شکایت‌ها، دادهٔ شبکه، IMEI، شارژ و... استفاده کردیم. داده شامل تراکنش‌های همهٔ مشتریان در ۹ ماه قبل از خط مبنا (Baseline prediction) بود. اندازهٔ این داده بیش از ۷۰ ترابایت بود و ما نمی‌توانستیم مرحلهٔ مهندسی ویژگی موردنیاز را با پایگاه‌داده‌های سنتی انجام دهیم.

**۲. تنوع داده**

دادهٔ استفاده‌شده در این پژوهش از چندین سیستم و پایگاه مختلف جمع‌آوری شده است. هر منبع داده را در نوع متفاوتی از فایل تولید می‌کند: ساخت‌یافته، نیمه‌ساخت‌یافته (XML یا JSON) یا بدون ساختار (CSV و متن خام). کار با این انواع داده بدون یک پلتفرم کلان‌داده خیلی سخت است، چون ما باید بتوانیم روی تمام این انواع داده، بدون نیاز به تغییر یا تبدیل اولیه، کار کنیم. با استفاده از پلتفرم کلان‌داده، دیگر اندازهٔ داده یا قالب داده برای ما مشکل نبود.

**۳. نامتوازن بودن مجموعه‌داده**

مجموعه‌دادهٔ ساخته‌شده نامتوازن (Imbalanced) بود. این یعنی توزیع کلاس‌ها همگن نیست. کلاسی که اکثریت است به‌عنوان کلاس اصلی شناخته می‌شود و کلاس دیگر کلاس فرعی است. اگر یکی از کلاس‌ها ۱۰٪ یا کمترِ کل نمونه‌ها باشد، مجموعه‌داده نامتوازن در نظر گرفته می‌شود [18].

الگوریتم‌های یادگیری ماشین معمولاً برای بهبود دقت کلی و کم‌کردن خطا طراحی شده‌اند، ولی همهٔ آنها توازن کلاس‌ها را در نظر نمی‌گیرند و همین می‌تواند نتایج بد تولید کند [18]. در حالت کلی، ما می‌خواهیم کلاس‌ها در آموزش از نظر وزن/اهمیت هم‌ارزش باشند.

فهمیدیم که مجموعه‌دادهٔ SyriaTel نامتوازن است: اندازهٔ کلاس فرعی، یعنی مشتریانی که «ریزش کرده‌اند»، حدود ۵٪ کل داده بود.

**۴. ویژگی‌های بسیار زیاد**

دادهٔ جمع‌آوری‌شده پر از ستون بود، چون برای هر خدمت، هر محصول و هر بستهٔ مربوط به تماس، SMS، MMS و اینترنت یک ستون وجود داشت، به‌علاوه ستون‌های اطلاعات شخصی و جمعیت‌شناختی. اگر قرار بود تمام این منابع داده را مستقیم استفاده کنیم، تعداد ستون‌ها برای هر مشتری (قبل از هرگونه پردازش) از ده‌هزار ستون بیشتر می‌شد.

**۵. مقادیر گمشده (Missing values)**

برای هر مشتری، نمایش جداگانه‌ای از هر خدمت و هر محصول وجود دارد. مقادیر گمشده ممکن است ظاهر شوند چون همهٔ مشتریان همهٔ سرویس‌ها را ندارند. بعضی مشتری‌ها تعدادی از سرویس‌ها را دارند و بقیه سرویس‌های دیگری. علاوه بر این، ستون‌هایی وجود دارند که فقط پیکربندی سیستم را بازتاب می‌دهند و این ستون‌ها برای تمام مشتریان فقط مقدار تهی (null) داشتند.

**روش پیشنهادی پیش‌بینی ریزش (Proposed churn method)**

برای ساخت سامانهٔ پیش‌بینی ریزش در SyriaTel، باید یک پلتفرم کلان‌داده نصب می‌شد. پلتفرم Hortonworks Data Platform (HDP) انتخاب شد چون رایگان و متن‌باز است و تحت مجوز Apache 2.0 عرضه می‌شود. HDP شامل مجموعه‌ای از سیستم‌ها و ابزارهای متن‌باز مرتبط با کلان‌داده است و این ابزارها با هم یکپارچه می‌شوند. در شکل ۱ (در مقالهٔ اصلی) اکوسیستم HDP نشان داده می‌شود که هر گروه از ابزارها زیر یک تخصص قرار می‌گیرند، مثل مدیریت داده، دسترسی به داده، امنیت، عملیات، و یکپارچه‌سازی حاکمیتی.

نصب چارچوب HDP به‌صورت سفارشی انجام شد تا فقط ابزارها و سیستم‌هایی که برای این کار لازم بود نصب شوند. این بستهٔ سفارشی از ابزارها و سیستم‌های نصب‌شده، SYTL-BD (چارچوب کلان‌دادهٔ SyriaTel) نام گرفت. ما اجزای زیر را نصب کردیم:

* سیستم فایل توزیع‌شدهٔ Hadoop (HDFS) برای ذخیرهٔ داده،
* موتور اجرای Spark برای پردازش داده،
* YARN برای مدیریت منابع،
* Zeppelin به‌عنوان رابط توسعه برای کاربران،
* Ambari برای مانیتور و نظارت سیستم،
* Ranger برای امنیت،
* Flume و Sqoop برای وارد کردن داده از بیرونِ چارچوب SYTL-BD به HDFS.

سخت‌افزار استفاده‌شده شامل ۱۲ نود، هرکدام با ۳۲ گیگابایت RAM، حافظهٔ ذخیره‌سازی ۱۰ ترابایت، و پردازندهٔ ۱۶ هسته‌ای بود. یک مجموعه‌دادهٔ ۹ماههٔ پیوسته جمع‌آوری شد. این مجموعه‌داده برای استخراج ویژگی‌های مدل پیش‌بینی ریزش استفاده شد. چرخهٔ حیات داده چند مرحله داشت (در شکل ۲ مقاله نشان داده شده).

موتور Spark در بیشتر مراحل مدل استفاده شد: پردازش داده، مهندسی ویژگی، آموزش و آزمون مدل. دلیلش این است که Spark پردازش را در RAM انجام می‌دهد و مزیت سرعت و مقیاس‌پذیری دارد. همچنین Spark مجموعهٔ بزرگی از کتابخانه‌ها برای تمام مراحل چرخهٔ یادگیری ماشین دارد.

**جمع‌آوری و ذخیره‌سازی داده**

اولین گام، انتقال داده از خارجِ SYTL-BD به HDFS بود. داده به سه نوع اصلی تقسیم می‌شد: ساخت‌یافته، نیمه‌ساخت‌یافته و بدون ساختار.

* **Apache Flume** یک سامانهٔ توزیع‌شده است که برای جمع‌آوری و انتقال فایل‌های بدون ساختار (CSV و متن) و نیمه‌ساخت‌یافته (JSON و XML) به HDFS استفاده می‌شود. در شکل ۳ مقاله معماری Flume در SYTL-BD نشان داده شده. Flume سه مؤلفهٔ اصلی دارد:
  + Source (منبع داده)،
  + Channel (مسیر عبور داده)،
  + Sink (مقصد نهایی داده).

در تنظیمات ما، Agentهای Flume فایل‌ها را از یک دایرکتوری مانیتور‌شونده (Spooling Directory Source) می‌خوانند و از طریق یک Channel منتقل می‌کنند. این Channel به‌صورت Memory Channel تعریف شد چون نسبت به سایر حالت‌ها در Flume عملکرد بهتری داشت. داده از کانال عبور می‌کند و در نهایت در Sink که همان HDFS است نوشته می‌شود. نوع قالب داده در HDFS همان قالب اصلی باقی می‌ماند.

* **Apache Sqoop** یک ابزار توزیع‌شده است برای انتقال حجم بالای داده بین HDFS و پایگاه‌داده‌های رابطه‌ای (یعنی دادهٔ ساخت‌یافته). با Sqoop تمام داده‌های موجود در پایگاه‌های رابطه‌ای به کمک Map jobها به HDFS منتقل شدند. شکل ۴ مقاله معماری فرایند import در Sqoop را نشان می‌دهد: مثلاً ۴ Mapper تعریف شده، و هر Map job بخشی از داده را انتخاب می‌کند و به HDFS می‌فرستد. داده پس از انتقال در HDFS به صورت CSV ذخیره شد.

بعد از اینکه تمام داده‌ها از منبع‌های مختلف وارد HDFS شد، باید قالب فایلی بهینه انتخاب می‌شد تا هم در زمان اجرا و هم در استفاده از فضا بهترین کارایی را بدهد. این آزمایش با Spark انجام شد: با کتابخانهٔ DataFrame، یک ترابایت دادهٔ CSV به دو نوع فایل Apache Parquet و Apache Avro تبدیل شد و سه سناریوی فشرده‌سازی هم بررسی شد.

نتیجه:

* فرمت Parquet انتخاب شد چون یک فرمت ستونی (columnar storage) است و کارایی خیلی خوبی دارد، مخصوصاً برای مهندسی ویژگی و کاوش داده.
* استفاده از فرمت Parquet همراه با فشرده‌سازی Snappy بهترین بهره‌وری فضایی را داشت.  
  (در شکل ۵ مقاله مقایسهٔ زمان اجرا و استفاده از فضا برای انواع فایل نشان داده شده است.)

**مهندسی ویژگی (Feature engineering)**

دادهٔ خام باید به ویژگی‌های قابل‌استفاده در الگوریتم‌های یادگیری ماشین تبدیل می‌شد. این طولانی‌ترین مرحله بود چون تعداد ستون‌ها بسیار زیاد بود.

ایدهٔ اولیه این بود:

* برای هر ستون عددی و برای هر مشتری، در هر ماه، مقادیری مثل میانگین، تعداد، جمع، حداکثر، حداقل و… را محاسبه کنیم.
* برای ستون‌های دسته‌ای (Categorical)، تعداد مقادیر متمایز را محاسبه کنیم.

دستهٔ دیگری از ویژگی‌ها بر اساس فعالیت اجتماعی مشتری‌ها از طریق SMS و تماس تلفنی محاسبه شد. Spark برای هر دو نوع ویژگی (آماری و اجتماعی) استفاده شد. برای ویژگی‌های SNA از کتابخانهٔ GraphFrame استفاده شد.

**الف) ویژگی‌های آماری**

این ویژگی‌ها از تمام انواع CDR (رکورد جزئیات تماس) ساخته شدند. مثلاً:

* میانگین تعداد تماس‌های انجام‌شده توسط مشتری در هر ماه،
* میانگین آپلود/دانلود اینترنت،
* تعداد بسته‌های فعال‌شده،
* درصد نوع دسترسی رادیویی (RAT) در هر سایت در هر ماه،
* نسبت تعداد تماس‌ها به تعداد SMS،
* و ویژگی‌های زیاد دیگر که با جمع‌زدن و تجمیع دادهٔ CDR ساخته شد.

چون دادهٔ مربوط به همهٔ اعمال مشتری در شبکه را داشتیم، داده‌های تماس، SMS، MMS و اینترنت را برای هر مشتری در بازهٔ روزانه، هفتگی و ماهانه، طی ۹ ماه گذشته جمع‌آوری و خلاصه کردیم. نتیجه این شد که تعداد ویژگی‌های ساخته‌شده بیش از سه برابر تعداد ستون‌ها شد.

همچنین ویژگی‌های مربوط به شکایت‌های ثبت‌شده هم وارد شد. مثل:

* تعداد شکایت‌ها،
* درصد شکایت‌های پوشش شبکه نسبت به کل شکایت‌ها،
* فاصلهٔ زمانی میان دو شکایت پی‌درپی (میانگین فاصلهٔ ساعتی)،
* زمان لازم تا بسته‌شدن شکایت،
* نتیجهٔ بسته‌شدن شکایت،
* و سایر مشخصه‌ها.

ویژگی‌های مربوط به IMEI هم استخراج شد، مثل:

* نوع گوشی،
* برند،
* تک‌سیم‌کارت یا دو‌سیم‌کارت بودن،
* و اینکه مشتری چند بار گوشی خودش را عوض کرده است.

برای طراحی ویژگی‌های با معنی بیزینسی، چند دور جلسهٔ طوفان فکری با نیروهای ارشد بخش بازاریابی انجام دادیم. علاوه بر چیزهایی که در مقالات قبلی ذکر شده بود، ویژگی‌های زیر را ساختیم:

* درصد تماس‌های ورودی/خروجی و SMS/MMS به/از اپراتور رقیب و خطوط ثابت،
* ویژگی‌های دودویی (بله/خیر) که نشان می‌داد آیا مشتری در سرویس خاصی عضو هست یا نه،
* نسبت استفاده از اینترنت بین 2G، 3G و 4G،
* تعداد دستگاه‌هایی که هر ماه استفاده شده،
* تعداد روزهایی که خارج از پوشش بوده،
* درصد دوستانی که در اپراتور رقیب هستند،
* و صدها ویژگی دیگر.

(در شکل‌های ۶ و ۷ مقاله، توزیع بعضی از ویژگی‌های مهم دسته‌ای و عددی بین کلاس ریزشی و غیرریزشی نشان داده شده است. رنگ قرمز = کلاس ریزش‌کرده، رنگ آبی = کلاس فعال.)

**ب) ویژگی‌های تحلیل شبکهٔ اجتماعی (SNA)**

برای استخراج این ویژگی‌ها، ما ارتباط بین هر دو مشتری را خلاصه کردیم و یک گراف شبکهٔ اجتماعی مبتنی بر داده‌های CDR چهار ماه آخر ساختیم. برای این کار از GraphFrame روی Spark استفاده شد. گراف اجتماعی شامل:

* **گره‌ها (Nodes):** شمارهٔ GSM مشترکین.
* **یال‌ها (Edges):** تعامل بین مشترکین (تماس، SMS و MMS).  
  یال‌ها جهت‌دار هستند (A→B با B→A فرق دارد).

در شکل ۸ مقاله یک نمونه از شبکهٔ اجتماعی سوریه نشان داده شده است.

* گره‌های قرمز: مشترکان SyriaTel
* گره‌های زرد: مشترکان MTN
* یال‌ها: تعامل بین آن‌ها

گراف کامل اجتماعی شامل حدود ۱۵ میلیون گره (SyriaTel، MTN و شماره‌های baseline) و بیش از ۲٫۵ میلیارد یال بود.

ویژگی‌های مبتنی بر گراف از این شبکه استخراج شدند. گراف جهت‌دار و وزن‌دار است. ما سه گراف مختلف ساختیم، بسته به اینکه وزن یال‌ها چه چیزی باشد. وزن یال تعریف شد به‌عنوان تعداد رویدادهای مشترک بین هر دو مشتری. سه نوع وزن داشتیم:

1. مدت زمان تماس نرمال‌سازی‌شده بین دو مشتری،
2. تعداد تماس/SMS/MMS نرمال‌سازی‌شده،
3. میانگین دو وزن قبلی.

نرمال‌سازی بسته به الگوریتم مورد استفاده فرق داشت. بر اساس این گراف‌های جهت‌دار، ما PageRank [19] و SenderRank [20] را محاسبه کردیم و برای هر گراف دو ویژگی تولید کردیم.

فرمول PageRank وزن‌دار برای یک مشتری m به‌شکل زیر است:

PR(m) = (1 - d) + d \* Σ\_{n ∈ N(m)} [ (W\_{n→m} / Σ\_{n' ∈ N(n)} W\_{n→n'}) \* PR(n) ]

و فرمول SenderRank وزن‌دار:

SR(m) = (1 - d) + d \* Σ\_{n ∈ N(m)} [ (W\_{m→n} / Σ\_{n' ∈ N(n)} W\_{n→n'}) \* SR(n) ]

توضیح:

* N(m): لیست دوستان مشتری m در شبکهٔ اجتماعی‌اش.
* W\_{n→m}: وزن یال جهت‌دار از n به m.
* d: فاکتور دمپینگ (damping factor).  
  در گراف‌های مخابراتی شما گره‌هایی دارید که فقط ورودی دارند و خروجی ندارند یا برعکس. این‌ها «گره‌های چاهک» (sink nodes) هستند. d باعث می‌شود این گره‌ها به‌صورت مصنوعی مقدار خیلی بزرگ نگیرند یا مقدار صفر مطلق نگیرند. d همچنین تضمین می‌کند مجموع PageRank یا SenderRank در هر دور به ۱ همگرا شود. در این کار d = 0.85 در نظر گرفته شد، که مقدار رایج در ادبیات [21, 22] است.

چون این معادلات مبتنی بر «راه‌رفتن تصادفی» تکرارشونده هستند، PR و SR بعد از چند تکرار پایدار می‌شوند. مقدار بالاتر PR(m) یا SR(m) یعنی آن مشتری در شبکهٔ اجتماعی مهم‌تر است.

ویژگی‌های SNA دیگر هم محاسبه شدند:

* درجهٔ ورودی (In-degree) و درجهٔ خروجی (Out-degree): تعداد دوستان متمایز در نقش دریافت‌کننده و در نقش فرستنده.
* پیوستگی همسایه (Neighbor Connectivity): میانگین درجهٔ همسایه‌ها برای هر مشتری:

NC(m) = ( Σ\_{k ∈ N(m)} |N(k)| ) / |N(m)|

* ضریب خوشه‌بندی محلی (Local Clustering Coefficient): این ویژگی نشان می‌دهد دوستان یک مشتری چقدر به هم وصل‌اند. یعنی تعداد ارتباط‌های موجود بین دوستان تقسیم بر تعداد تمام ارتباط‌های ممکن بین آن‌ها:

LC(m) = [ Σ\_{k ∈ N(m)} |N(m) ∩ N(k)| ] / [ |N(m)| \* (|N(m)| - 1) ]

همچنین از شبکه برای سنجش «شباهت» بین مشتری‌ها استفاده شد، بر اساس مفهوم «دوستان مشترک». برای هر مشتری دو نوع شاخص شباهت محاسبه شد: شباهت جاکارد (Jaccard) و شباهت کسینوسی (Cosine).

* شباهت جاکارد بین مشتری m و k:

J(m,k) = |N(m) ∩ N(k)| / |N(m) ∪ N(k)|

* شباهت کسینوسی (که در مقاله هم اشتباهاً با JS علامت‌گذاری شده ولی در واقع Cosine است):

Cos(m,k) = |N(m) ∩ N(k)| / sqrt( |N(m)| \* |N(k)| )

هدف این بود که ببینیم آیا مشتری دارد شبکهٔ خودش را روی اپراتور رقیب بازتولید می‌کند یا نه. وقتی کسی قصد مهاجرت دارد، معمولاً با خط جدید در اپراتور دیگر شروع می‌کند با همان دوستان کلیدی، فقط با یک لیست تماس کوچک‌تر. پس اگر شباهت کسینوسی با یک شمارهٔ MTN بالا می‌رود، یعنی این مشتری احتمالاً دارد مهاجرت می‌کند.

برای کم کردن هزینهٔ محاسباتی (چون این مقایسه‌ها برای هر جفت در گراف خیلی بزرگ است)، ما فقط زوج مشترک‌هایی را نگه داشتیم که حداقل یک دوست مشترک داشتند. سپس برای هر مشتری، بالاترین شباهت جاکارد و کسینوسی نسبت به یک مشتری SyriaTel و نسبت به یک مشتری MTN ذخیره شد (یعنی max cosine/jaccard با SyriaTel و با MTN).  
این ویژگی‌ها برای اولین‌بار در این کار برای پیش‌بینی ریزش به‌کار رفتند و اثر مثبت داشتند. در شکل ۹ مقاله توزیع بعضی از این ویژگی‌ها مثل شباهت کسینوسی با اپراتور دیگر، ضریب خوشه‌بندی محلی، و فاکتور توان اجتماعی (Social Power Factor که میانگین PageRank و SenderRank است) نشان داده شده است.

جدول ۱ مقاله، تعدادی از مهم‌ترین ویژگی‌های SNA را توضیح می‌دهد:

* In-Degree: تعداد دوستانی که با مشتری تماس می‌گیرند / به او پیام می‌دهند
* Out-Degree: تعداد دوستانی که مشتری با آن‌ها تماس/پیام دارد
* Max-Cosine-Sim-MTN: بیشترین شباهت کسینوسی با مشترکین اپراتور رقیب
* Max-Cosine-Sim-SyriaTel: بیشترین شباهت کسینوسی با مشترکین SyriaTel
* Max-Jaccard-Sim-MTN و Max-Jaccard-Sim-SyriaTel: مشابه بالا ولی با معیار جاکارد
* SR و PR: Sender Rank و PageRank وزن‌دار در گراف
* PF یا Social Power Factor: میانگین PR و SR
* Betweenness: تعداد کوتاه‌ترین مسیرهایی که از این مشتری رد می‌شوند
* LCC: ضریب خوشه‌بندی محلی
* NC: پیوستگی همسایگی (چند دوست و دوستِ‌دوست دارد)

**تبدیل و انتخاب ویژگی‌ها (Feature transformation and selection)**

برخی ویژگی‌ها مثل شناسهٔ قرارداد (Contract ID)، شمارهٔ MSISDN و سایر چیزهای یکتا که مشخصهٔ یک فرد خاص است، حذف شدند. چون اینها اثر مستقیم روی خروجی هدف دارند (هویت فردی) و ما نمی‌خواهیم مدل فقط مشتری را «حفظ» کند.

حذف کردیم:

* ویژگی‌های با مقدار تکراری یکسان یا مقدار تهی کامل،
* ویژگی‌های تکراری (Duplicate)،
* ویژگی‌هایی که فقط چند مقدار عددی محدود تکراری دارند.

فهمیدیم که بیش از نیمی از ویژگی‌ها در بیش از ۹۸٪ سطرها مقدار گمشده داشتند. ما ابتدا سعی کردیم هر ویژگی که حتی یک مقدار خالی دارد را حذف کنیم، ولی نتیجهٔ مدل بد شد.

در نهایت این استراتژی را پیاده کردیم:

* رکوردهایی که بیش از ۹۰٪ از ویژگی‌هایشان خالی بود حذف شدند.
* ویژگی‌هایی که بیش از ۷۰٪ مقدار خالی داشتند حذف شدند.
* در ویژگی‌های دسته‌ای، مقادیر گمشده را با یک دستهٔ جدید به نام «Other» جایگزین کردیم.
* مقادیر عددی گمشده را با میانگین آن ویژگی پر کردیم.
* برای ویژگی‌های دسته‌ای: ۷۸ ویژگی دسته‌ای داشتیم. برای هرکدام، ۳۱ دستهٔ پرتکرار اول را نگه داشتیم و بقیهٔ دسته‌ها را در یک دستهٔ جدید ادغام کردیم، تا در نهایت تعداد دسته‌ها ۳۲ تا باشد.

یک‌سری ویژگی‌ها از نظر ظاهری عددی بودند ولی فقط چند مقدار تکراری محدود داشتند، یعنی در عمل دسته‌ای بودند. ما آن‌ها را هم به‌صورت دسته‌ای در نظر گرفتیم ولی دیدیم مدل با آن‌ها بدتر عمل می‌کند، پس حذفشان کردیم.

همچنین همبستگی (Correlation) بین ویژگی‌های عددی با ضریب پیرسون محاسبه شد و ویژگی‌های به‌شدت همبسته حذف شدند. این حذف تأثیر منفی روی نتیجهٔ نهایی نداشت. روش‌های دیگر هم تست شد ولی این رویکرد بهترین عملکرد را در چهار الگوریتم داشت. در پایان، بعد از این مراحل هنوز بیش از ۲۰۰۰ ویژگی باقی مانده بود.

بعد، ما برای آموزش و تست باید برچسب (Label) داشته باشیم. با بخش بازاریابی صحبت کردیم تا یک نمونهٔ برچسب‌خورده از شماره‌ها (GSMها) بدهند. آن‌ها مشتریان اعتباری (prepaid) را که «در فاز غیرفعال شدن» بودند، دو ماه بعد از بازهٔ ۹ماهه، به‌عنوان «ریزش‌کرده» (churner) علامت زدند. بقیهٔ مشتریانی که ریزش نکرده بودند به‌عنوان «Active» (فعال) برچسب خوردند. مشتریانی که در ۴ ماه آخر تازه جذب شده بودند حذف شدند، چون تازه‌وارد هستند.

در نهایت:

* کل نمونه ≈ ۵ میلیون مشتری
* حدود ۳۰۰٬۰۰۰ مشتری ریزشی
* حدود ۴٫۷ میلیون مشتری فعال

در شکل ۱۰ مقاله بازهٔ دادهٔ تاریخی و بازهٔ آینده که طی آن مشتری ممکن است شرکت را ترک کند نمایش داده شده.  
بخش بازاریابی تصمیم گرفت ریزش را ۲ ماه قبل از رخ‌دادنش پیش‌بینی کنیم تا زمان کافی برای اقدام پیش‌گیرانه داشته باشند.

**دسته‌بندی (Classification)**

داده به دو بخش تقسیم شد:

* آموزش (۷۰٪)
* آزمون (۳۰٪)

گروه آموزش برای آموزش الگوریتم‌ها استفاده شد و گروه آزمون برای تست.  
فوق‌پارامترها (Hyperparameters) با Cross-validation نوع k-fold بهینه شدند، با k=10.

چون کلاس هدف نامتوازن بود، این می‌توانست اثر منفی بزرگی روی مدل بگذارد. ما این مشکل را با باز-متعادل‌سازی نمونهٔ آموزشی حل کردیم تا دو کلاس (ریزش و فعال) بالانس‌تر شوند [25].

سه سناریو امتحان شد:

1. Oversampling: تکرار کلاس اقلیت (ریزش) تا تعادل ایجاد شود.
2. Undersampling: کوچک‌کردن کلاس اکثریت تا به اقلیت نزدیک شود.
3. بدون هیچ باز-متعادل‌سازی.

روش undersampling (کم‌کردن کلاس بزرگ) در مقالات قبلی هم استفاده شده و در بعضی الگوریتم‌ها بهترین نتیجه را داد [8, 26]. بعد از undersampling، اندازهٔ نمونهٔ آموزشی حدود ۴۲۰٬۰۰۰ شد.

ما این الگوریتم‌ها را آموزش دادیم:

* **درخت تصمیم (Decision Tree):**  
  عمق درخت (depth) و حداکثر تعداد گره‌ها (max nodes) بهینه شد. بهترین حالت حدود ۳۹۸ گره و عمق ۲۰ بود.
* **جنگل تصادفی (Random Forest):**  
  تعداد درخت‌ها به‌عنوان فوق‌پارامتر بررسی شد (۱۰۰، ۲۰۰، ۳۰۰، ۴۰۰، ۵۰۰). بهترین نتیجه با ۲۰۰ درخت بود. بالاتر از ۲۰۰، بهبود مهمی دیده نشد.
* **گرادیان بوست‌شونده (Gradient Boosted Machine, GBM):**  
  دوباره تعداد درخت‌ها تا ۵۰۰ بررسی شد و بهترین مقدار حدود ۲۰۰ درخت بود. GBM بهتر از RF و DT عمل کرد.
* **XGBOOST:**  
  ما XGBoost را روی Spark 2.3 نصب و با کتابخانهٔ ML در Spark یکپارچه کردیم و همان روند را تکرار کردیم. بهترین تعداد درخت‌ها حدود ۱۸۰ بود.

**نتایج و بحث (Results and discussion)**

ما عملکرد الگوریتم‌ها را تحت اندازه‌های مختلف دادهٔ آموزشی و تحت سه حالت موازنه (oversampling، undersampling، بدون موازنه) بررسی کردیم.

سؤال اول: چه بازهٔ تاریخی برای استخراج ویژگی‌ها بهتر است؟ یعنی «طول پنجرهٔ لغزان» (sliding window) چقدر باشد؟

در شکل ۱۱ مقاله:

* M1 یعنی ماه اول قبل از baseline
* M9 یعنی ماه نهم قبل از baseline

برای هر N، ویژگی‌های ماه N یعنی تجمیع داده از ماه ۱ تا ماه N.

نتیجه:

* برای ویژگی‌های آماری: هرچه دادهٔ تاریخی بیشتری (ماه‌های اخیر تا ۶ ماه قبل) استفاده شد، عملکرد بهتر شد، ولی افزودن سه ماه خیلی قدیمی‌تر از آن دیگر بهبود خاصی ایجاد نکرد. وقتی فقط ویژگی‌های آماری استفاده شد، بالاترین AUC حدود ۸۴٪ بود. (شکل ۱۱a)
* برای ویژگی‌های SNA: بهترین بازه برای ساخت گراف اجتماعی و استخراج ویژگی‌های SNA فقط چهار ماه آخر بود. اضافه‌کردن ماه‌های قدیمی‌تر حتی بدتر هم کرد. وقتی فقط ویژگی‌های SNA استفاده شد، بالاترین AUC حدود ۷۵٫۳٪ بود. (شکل ۱۱b)

پس در ادامهٔ کار:

* برای ویژگی‌های آماری از ۶ ماه آخر استفاده شد.
* برای ویژگی‌های SNA از ۴ ماه آخر استفاده شد.  
  (همان‌طور که در شکل ۱۰ مقاله نشان داده شده است: بخش آبی = پنجرهٔ ویژگی‌های آماری، نارنجی = پنجرهٔ ویژگی‌های SNA، و خط قرمز = baseline)

وقتی هر دو نوع ویژگی (آماری + SNA) را با هم به الگوریتم‌ها دادیم، عملکرد به‌طور چشمگیری بهتر شد. طبق جدول ۲ و شکل ۱۱c، حداکثر مقدار AUC به ۹۳٫۳٪ رسید.

سؤال دوم: نامتوازن بودن کلاس‌ها.  
جدول ۳ مقاله نشان می‌دهد:

* برای XGBOOST و GBM، بهترین عملکرد با دادهٔ بدون هیچ باز-متعادل‌سازی (یعنی همون نسبت واقعی کلاس‌ها) به‌دست آمد.
* برای Random Forest و Decision Tree، بهترین عملکرد با undersampling بود.

در مجموع:

* XGBOOST بهترین بود با AUC = ۹۳٫۳٪
* GBM دوم بود با AUC ≈ ۹۰٫۹٪
* Random Forest و درخت تصمیم به‌ترتیب حدود ۸۷٫۷۶٪ و ۸۳٪ گرفتند.

(در شکل ۱۲ مقاله، منحنی ROC هر کدام از این الگوریتم‌ها نشان داده شده است.)

**ویژگی‌های مهم (Feature importance)**

ویژگی‌های مهم با معیار Gain [27] رتبه‌بندی شدند. Gain بالاتر یعنی آن ویژگی در پیش‌بینی ریزش نقش مهم‌تری دارد.

در شکل ۱۳ مقاله نشان داده شده:

* وقتی فقط ویژگی‌های آماری هست،
* وقتی فقط ویژگی‌های SNA هست،
* و وقتی هر دو ادغام شده‌اند.

بعد از ادغام، رتبهٔ اهمیت عوض شد. مهم‌ترین ویژگی شد «شباهت کسینوسی با MTN» (Max Cosine Similarity to MTN). یعنی مشتریانی که بیشترین شباهت شبکهٔ اجتماعی‌شان با یک شمارهٔ MTN است، بیشترین احتمال ریزش را داشتند؛ حتی اگر بقیهٔ شاخص‌ها مثل بالانس، مصرف اینترنت، تماس ورودی/خروجی هنوز عادی به‌نظر برسد. (توزیع این ویژگی در شکل ۹a مقاله آمده است.)

تحلیل: معمولاً مشتری یک خط جدید در اپراتور رقیب می‌گیرد و مدتی هر دو خط را فعال نگه می‌دارد. در این دوره سعی می‌کند مخاطبین مهمش را به خط جدید منتقل کند. بعد از اینکه مطمئن شد تماس‌های مهم به خط جدید می‌آیند، خط قدیمی را خاموش می‌کند. همچنین ممکن است فقط صبر کند تا شارژ باقی‌مانده را مصرف کند. شکل ۱۴ مقاله یک مشتری را در طول هفته‌های قبل از ریزش دنبال می‌کند: او در هفتهٔ ۷ سیم‌کارت MTN خریده، در هفتهٔ ۱۳ و ۱۴ دیگر در شبکهٔ SyriaTel پوشش ندارد، و در هفتهٔ ۱۴ سیم‌کارت SyriaTel را عملاً کنار گذاشته است.

ویژگی دوم از نظر اهمیت: «روز آخرین تراکنش خروجی» (Days of Last Outgoing Transaction). همان‌طور که در شکل ۷a مقاله دیده می‌شود، مشتریان ریزشی معمولاً برای مدت طولانی‌تری بدون هیچ تراکنش خروجی می‌مانند.

ویژگی سوم مهم: «موجودی کل» (Total Balance). شکل ۷c نشان می‌دهد مشتریان ریزشی معمولاً موجودی پایین‌تری دارند.

ویژگی چهارم: «میانگین نوع دسترسی رادیویی» (Average Radio Access Type). در شکل ۷b، مشتریان ریزشی بیشتر در 2G هستند تا 3G. کیفیت و سرعت 2G پایین‌تر است. اگر مشتری مصرف اینترنت بالایی دارد ولی در ناحیه‌ای است که 3G رقیب بهتر است، احتمال ریزش بالاتر است. تحلیل نشان داد ۶۸٪ از مشتریان ریزشی کاربر اینترنت بودند و ۶۵٪ از آنها مقدار پایین برای این ویژگی داشتند.

ویژگی پنجم مهم: «ضریب خوشه‌بندی محلی» (LCC). در شکل ۹b، مشتریانی با LCC خیلی پایین کمتر متمایل به ریزش‌اند. این به این شکل تفسیر شد: تعدادی از مشتریان از خط شخصی‌شان برای مقاصد تجاری (کافه، آرایشگاه، رستوران، کلینیک و...) استفاده می‌کنند. آن‌ها با افراد زیادی در تماس‌اند که همدیگر را نمی‌شناسند، پس شبکهٔ آن‌ها ساختار «ستاره‌ای» دارد و LCC خیلی پایین است. این افراد معمولاً چند سیم‌کارت دارند ولی مجبورند شماره‌شان را حفظ کنند چون مشتری از روی همان شماره به آنها دسترسی دارد. برای بررسی این فرضیه، با چند نفر از کسانی که LCC خیلی پایین داشتند تماس گرفته شد و مشخص شد اغلبشان صاحب یا کارمند کافه، رستوران، آرایشگاه، کتاب‌فروشی، گیم‌نت، کلینیک و… هستند.

ویژگی ششم: «درصد تراکنش‌ها به/از اپراتور دیگر» (Percentage of Transactions to/from other Operator). هرچه این درصد بیشتر باشد، احتمال ریزش بیشتر است. این نشان می‌دهد دوستان مهم فرد در اپراتور رقیب هستند و همین فشار اجتماعی ایجاد می‌کند که او هم مهاجرت کند. شکل ۷d این توزیع را نشان می‌دهد.

ویژگی بعدی: «سن مشتری». مشتریان جوان‌تر از ۳۲ سال احتمال ریزش بالاتری دارند. دلیل احتمالی: این گروه جوان مصرف بالاتر اینترنت و سرویس و در کل حساسیت به کیفیت/قیمت دارد و راحت‌تر اپراتور را عوض می‌کند. (در شکل ۶ توزیع این ویژگی نسبت به کلاس ریزش نشان داده شده است.)

«فاکتور توان اجتماعی» (Social Power Factor = میانگین PageRank و SenderRank) هم یکی از ویژگی‌های SNA مهم بود. در شکل ۹c، هرچه PF بالاتر باشد، احتمال ریزش کمتر است. و در شکل ۷e، می‌بینیم مشتریانی که خطاهای سیگنال و تماس‌های قطع‌شده (dropped calls) بیشتری داشتند، بیشتر متمایل به ریزش بودند.

در مجموع (همان‌طور که در شکل‌های ۱۱c و ۱۲c مقاله دیده می‌شود)، ویژگی‌های SNA نقش خیلی مهمی در بهبود عملکرد مدل پیش‌بینی ریزش داشتند چون یک بُعد اجتماعی از رفتار مشتری را وارد مدل می‌کردند که فقط با دادهٔ مصرف و شکایت و ترافیک قابل دیدن نبود.

**ارزیابی سامانه (System evaluation)**

سامانه با یک مجموعه‌دادهٔ جدید و به‌روز ارزیابی شد. این تست روی تمام مشترکین اعتباری SyriaTel انجام شد، یعنی روی ۷٫۵ میلیون مشتری، بدون اینکه بدانیم دو ماه بعد چه بر سرشان می‌آید.

ما همان مدل‌ها را روی این داده (بعد از همان نوع پردازش) اجرا کردیم. بعد، مشتریانی که مدل پیش‌بینی کرده بود «احتمال ریزش بالا» دارند، به دو گروه تقسیم شدند:

* گروه «Offer شده» (Offered): تیم بازاریابی به‌صورت فعال با آن‌ها تماس گرفت و پیشنهاد نگهداشت (Retention offer) داد.
* گروه «Offer نشده» (NotOffered): هیچ اقدامی انجام نشد.

بعد از دو ماه، وضعیت واقعی مشتریان در هر دو گروه بررسی شد و با پیش‌بینی مدل مقایسه شد.

نتیجه:

* بهترین AUC در این تست واقعی برای XGBOOST روی گروه «Offer نشده» حدود ۸۹٪ بود. یعنی حتی بدون دخالت بازاریابی، مدل تا حد خیلی خوبی درست حدس زده بود چه کسی می‌رود. (جدول ۴ در مقاله AUC هر چهار الگوریتم را روی این دادهٔ «Offer نشده» نشان می‌دهد: XGBOOST حدود ۸۹٪، GBM حدود ۸۵٫۵٪، جنگل تصادفی حدود ۸۳٫۴٪، درخت تصمیم حدود ۷۹٫۱٪.)
* در گروهی که بازاریابی مداخله کرد (Offered)، حدود ۴۷٪ از مشتریانی که پیش‌بینی شده بود می‌روند، در نهایت ماندند. یعنی تقریباً نصفشان بعد از دریافت پیشنهاد ویژه تصمیمشان را عوض کردند. نتیجه برای شرکت خیلی خوب بود: درآمد بالا رفت و نرخ ریزش حدود ۱٫۵٪ کاهش پیدا کرد.

**جمع‌بندی (Conclusion)**

اهمیت این نوع پژوهش در بازار مخابراتی این است که به شرکت‌ها کمک می‌کند سود بیش‌تری بسازند. پیش‌بینی ریزش الان یکی از مهم‌ترین منابع درآمد بالقوه برای اپراتورهای مخابراتی است.

این پژوهش با هدف ساخت یک سامانهٔ پیش‌بینی ریزش برای شرکت SyriaTel انجام شد. این مدل‌ها باید مقدار AUC بالایی به‌دست بیاورند. برای آموزش و تست، داده به ۷۰٪ (آموزش) و ۳۰٪ (تست) تقسیم شد و Cross-validation با ۱۰-فولد برای بهینه‌سازی فوق‌پارامترها انجام شد.

ما مهندسی ویژگی، تبدیل ویژگی و انتخاب ویژگی را با دقت انجام دادیم تا ویژگی‌ها آمادهٔ الگوریتم‌های یادگیری ماشین شوند. همچنین با مشکل نامتوازن بودن داده روبه‌رو بودیم (فقط حدود ۵٪ مشتریان در واقع ریزشی بودند). این مشکل با undersampling یا با استفاده از الگوریتم‌های درختی که نسبت به نامتوازن بودن مقاوم‌تر هستند مدیریت شد.

ما چهار الگوریتم مبتنی بر درخت را انتخاب کردیم چون هم متنوع‌اند و هم در این نوع پیش‌بینی کاربردی‌اند:

* درخت تصمیم (Decision Tree)،
* جنگل تصادفی (Random Forest)،
* گرادیان بوست‌شونده (GBM)،
* XGBOOST.

روش آماده‌سازی و انتخاب ویژگی‌ها، و مخصوصاً واردکردن ویژگی‌های شبکهٔ اجتماعی موبایل، بزرگ‌ترین تأثیر را روی موفقیت مدل داشت. در SyriaTel مقدار AUC به حدود ۹۳٫۳۰۱٪ رسید. مدل XGBOOST بهترین نتایج را داد. GBM دوم شد، و جنگل تصادفی و درخت تصمیم در رتبه‌های بعدی بودند.

ما مدل را همچنین روی یک مجموعه‌دادهٔ جدید که دورهٔ زمانی متفاوتی داشت و بدون دخالت بازاریابی آزمایش کردیم. XGBOOST باز هم بهترین عملکرد را داشت و AUC حدود ۸۹٪ به‌دست آمد. افت اندک دقت در این حالت احتمالاً به خاطر «ناایستایی در توزیع داده در طول زمان» (data non-stationarity) است ــ یعنی باید مدل را هر چند وقت یک‌بار دوباره آموزش داد.

نتیجهٔ کلی: استفاده از ویژگی‌های تحلیل شبکهٔ اجتماعی (SNA) دقت پیش‌بینی ریزش در مخابرات را به‌طور محسوسی بالا می‌برد.

**نویسندگان**

* AKA (عبدالرحیم کاسم احمد): مرور ادبیات، ساخت پلتفرم کلان‌داده، کار روی مدل پیشنهادی ریزش، انجام آزمایش‌ها و نگارش مقاله.
* AJ و KJ: نقش نظارتی داشتند و بر تکمیل کار نظارت کردند.  
  همهٔ نویسندگان نسخهٔ نهایی مقاله را خوانده و تأیید کردند.

**قدردانی**

از SyriaTel، خانم مدیرعامل Majda Sakr، آقایان Murid Atassi و Adham Troudi بابت پشتیبانی و انگیزه تشکر می‌شود. همچنین از آقایان Mhd Assaf، Nour Almulhem، William Soulaiman، Ammar Asaad، Soulaiman Moualla، Ahmad Ali و خانم Marwa Hanhoun برای همراهی و کمک تشکر می‌شود. از آقایان Kasem Jamil Ahmad و Fahmi Ammareen هم بابت بازبینی نسخهٔ نهایی مقاله سپاسگزاری می‌شود.

**تعارض منافع**

نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

**دسترسی به داده و مواد**

داده‌ها به شکل عمومی در دسترس نیستند چون تحت محدودیت SyriaTel هستند. داده فقط برای پژوهشگران داخل SyriaTel در دسترس است و فقط با اجازهٔ شرکت می‌تواند در اختیار دیگران قرار بگیرد.

**موافقت با انتشار**

نویسندگان با انتشار موافقت دارند.

**اخلاق و رضایت برای مشارکت**

همهٔ نویسندگان تأیید اخلاقی و رضایت برای مشارکت در ارسال و بازبینی را اعلام می‌کنند.

**تأمین مالی**

نویسندگان اعلام می‌کنند تأمین مالی خارجی نداشتند.

**یادداشت ناشر**

Springer Nature نسبت به دعاوی حقوقی مربوط به نقشه‌ها و وابستگی‌های سازمانی اعلام بی‌طرفی می‌کند.