

بِسْمِ اللَّهِ الرَّحْمَنِ

الرَّحِيمِ



گزارش کار پروژه‌ی داده کاوی

علی بدیعی گورتی

بهمن ماه ۱۴۰۲

فهرست

| | |
|---|----|
| چکیده | ۴ |
| مقدمه | ۵ |
| فصل اول: مقدمه به پیش نیازهای برنامه نویسی و علمی | ۷ |
| پیش نیازهای علمی برای انجام این پروژه | ۹ |
| فصل دوم: تحلیل، طراحی و اجرای فاز ۱ | ۱۰ |
| خروجی SQL | ۱۴ |
| فصل سوم: تحلیل، طراحی و اجرای فاز ۲ | ۱۷ |
| خروجی نمودار | ۲۱ |
| فصل چهارم: تحلیل، طراحی و اجرای فاز ۳ | ۲۳ |
| فصل پنجم: ارزیابی و بهبود نتایج | ۳۱ |
| مقایسه مدل ها با اعمال Optimization ها | ۳۵ |
| منابع | ۳۸ |

چکیده

پروژه داده کاوی: ارزیابی سختی سوالات با استفاده از مدل‌های متنوع

در این پروژه، از تنوع تکنیک‌های داده کاوی و یادگیری ماشین بهره گرفته می‌شود تا اطلاعات از یک سایت جمع‌آوری شده و ویژگی‌های متنوع و حیاتی از سوالات استخراج گردد. از الگوریتم‌های معتبر و متنوعی از جمله SVM، Logistic Regression، CART، SVC، Naive Bayes، XGBoost و MLP برای آموزش مدل‌های داده کاوی بهره می‌گیریم. به عنوان یک الگوریتم پیشرفته بردار پشتیبانی، توانمندی بسیاری در دسته‌بندی دارد و از آن برای تجزیه و تحلیل ویژگی‌ها بهره می‌بریم. XGBoost با قابلیت گرادین بوسستینگ و درخت تصمیم، در افزایش دقت و کارایی مدل‌ها به کار گرفته می‌شود. با ترکیب این الگوریتم‌ها و اجرای آموزش بر روی داده‌های آموزشی، دانشجو می‌تواند به تحلیل گسترده‌تری از عوامل موثر بر سختی سوالات دست پیدا کند. این پروژه فراهم کننده فرصتی مناسب برای توسعه مهارت‌های تحلیل داده، استفاده از تکنیک‌های پیشرفته داده کاوی، و ایجاد مدل‌های پیش‌بینی بر اساس داده‌های واقعی است. این تجربه به دانشجو این امکان را می‌دهد که به عنوان تحلیلگران داده، در حوزه تحلیل داده‌های پیچیده و متنوع تخصص کسب کنند.

مقدمه

در فصل اول، به بررسی پیش‌نیازهای برنامه‌نویسی و علمی می‌پردازیم.

در فصل دوم، به نحوه‌ی کارکرد و طراحی Query استفاده شده در <https://data.stackexchange.com/stackoverflow/query> می‌پردازیم. سپس خروجی‌های مورد انتظار از آن را دانلود کرده و با استفاده از روش‌های برنامه‌نویسی مشخص، سایر موارد مورد پرسش در متن پروژه را به دست می‌آوریم. سپس با انجام تغییرات جزئی در متن‌های ستون Body (حذف تگ‌های HTML و ...)، فایل نهایی از فاز ۱ را به دست می‌آوریم.

در فصل سوم، در خصوص انجام فاز ۲ صحبت می‌کنیم. در این فاز، ابتدا تمامی داده‌ها بر اساس تگ، عنوان سوال و متن سوال مورد ارزیابی قرار می‌گیرند. دو نفر از گروه سپس تک‌تک داده‌ها را بررسی کرده و بر اساس نظر خود عددی از ۱ تا ۴۷ را انتخاب می‌کنند. اگر هر دو نفر یک رای داشته باشند، آن عدد پذیرفته می‌شود، در غیر این صورت به فرد سوم انتقال می‌یابد. در صورت عدم توافق، رای اکثریت انتخاب می‌شود. اگر هر سه نفر رای متفاوتی داشته باشند، یکی از آن‌ها انتخاب می‌شود و در یکی از گروه‌های ساده (از ۱ تا ۱۴)، متوسط (از ۱۵ تا ۳۱) یا سخت (۳۲ به بالا) قرار می‌گیرد. برای

برچسب‌زدن از برنامه‌ای استفاده می‌شود که در هر مرحله یک سوال را نشان داده و منتظر گرفتن جواب آن باشد، که باعث افزایش سرعت برچسب‌زدن می‌شود. همچنین، طراحی برنامه به نحوی است که در صورت قطع برنامه یا اتمام عمل برچسب‌زدن، داده‌های لیبل‌زده شده از بین نرود. در نهایت، داده‌های لیبل‌زده شده به عنوان یک ستون جدید به داده‌ها افزوده می‌شود و خروجی فاز ۲ را تولید می‌کند. همچنین، نمودارها برای نمایش خروجی‌ها تعبیه شده‌اند.

در فصل چهارم، به اعمال انجام شده در فاز ۳ اشاره شده است. تمامی مراحل تبدیل خروجی فاز ۲ به موارد خواسته شده در متن توضیح داده شده و نحوه پردازش روی متن نیز توضیح داده شده است. سپس نحوه ساخت داده‌های Train و Test را تشریح کرده و در نهایت، پیاده‌سازی و آموزش مدل‌های Naïve Bayes، SVC، CART، Logistic، MLP، Regression و XGBoost را بیان کرده‌ایم. خروجی‌های این فاز برای تحلیل در فصل پنجم مورد استفاده قرار گرفته‌اند.

در فصل پنجم، نتایج مورد بررسی قرار گرفته و مدل‌ها با یکدیگر مقایسه می‌شوند.

فصل اول: مقدمه به پیش نیازهای برنامه‌نویسی و علمی

در این فصل، به بررسی و توضیح پیش‌نیازهایی که برنامه‌نویسان و علمای داده باید درک کنند، می‌پردازیم. این پیش‌نیازها از جمله مفاهیم اساسی برنامه‌نویسی، الگوریتم‌ها، و مهارت‌های علمی شامل تجزیه و تحلیل داده، استخراج اطلاعات مفید، و تفسیر نتایج آماری می‌شوند. هدف این فصل، فراهم کردن زمینه‌ای مناسب برای درک بهتر مفاهیم و ابزارهایی است که در فازهای بعدی پروژه به کار گرفته می‌شوند. به علاوه، توضیحاتی در مورد استانداردها و روش‌های مرتبط با برنامه‌نویسی و علم داده نیز ارائه خواهد شد.

پیش‌نیازهای برنامه‌نویسی برای انجام این پروژه:

۱. مهارت در زبان‌های برنامه‌نویسی:

برنامه‌نویسی یکی از مهارت‌های اساسی است که در این پروژه به طور فعال به کار خواهد رفت. مهارت در زبان‌هایی مانند Python یا R امکان اجرای کدها و پردازش داده‌ها را بهبود می‌بخشد. در این پروژه از زبان پایتون استفاده شده است.

۲. آشنایی با مفاهیم پایگاه داده:

اطلاعات مورد نیاز برای پروژه از پایگاه داده‌های Stack Exchange به دست می‌آید. بنابراین، آشنایی با مفاهیم مانند SQL و نحوه‌ی استخراج داده از پایگاه داده‌ها از اهمیت بالایی برخوردار است.

۳. توانایی در تحلیل داده:

تحلیل داده‌ها و استفاده از روش‌های مختلف برای استخراج الگوها و اطلاعات از داده‌های حاصل از پایگاه داده Stack Exchange ضروری است. مفاهیم مانند خوشه‌بندی، تجزیه و تحلیل آماری، و تصویرسازی داده می‌توانند مفید باشند.

۴. آشنایی با مفاهیم داده‌کاوی:

در فازهای مختلف پروژه، داده‌ها نیاز به برچسب‌زنی و دسته‌بندی دارند. آشنایی با مفاهیم داده‌کاوی و تکنیک‌های مختلف برچسب‌زنی و دسته‌بندی می‌تواند در اینجا مفید باشد.

۵. مهارت در استفاده از ابزارهای مرتبط:

استفاده از ابزارهایی مانند Jupyter Notebook و Pycharm برای اجرای کد، ایجاد گزارش‌های تحلیلی، و ایجاد نمودارها به اهمیت زیادی دارد. مهارت در استفاده از این ابزارها به بهبود فرآیند تحلیل و گزارش‌دهی کمک خواهد کرد.

با توجه به این پیش‌نیازها، برنامه‌نویسان می‌بایست با دقت و توجه به جزئیات این مراحل را پیش ببرند تا به بهترین نتایج در انجام پروژه دست یابند.

پیش‌نیازهای علمی برای انجام این پروژه:

۱. دانش در زمینه‌ی معیارهای ارزیابی:

در تحلیل داده‌ها و پیاده‌سازی مدل‌های یادگیری ماشین، شناخت دقیق از معیارهای ارزیابی از جمله F1 Score و Accuracy اساسی است. فهم درست از این معیارها در ارزیابی عملکرد مدل‌ها حائز اهمیت است.

۲. آشنایی با تکنیک‌های برچسب‌زنی و دسته‌بندی:

برچسب‌زنی داده‌ها به کمک مدل‌های یادگیری ماشین نیاز به فهم عمیق از تکنیک‌های برچسب‌زنی دارد. در اینجا، مفاهیمی مانند ماتریس درهم‌ریختگی (Confusion Matrix) می‌توانند به درک بهتری از عملکرد مدل‌ها کمک کند.

۳. آشنایی با تحلیل دقیق نتایج آماری:

در فازهای مختلف پروژه، نیاز به تحلیل دقیق نتایج آماری و اعتبارسنجی مدل‌ها وجود دارد. آشنایی با مفاهیم مانند انحراف معیار، p-value، و تفسیر نتایج آماری می‌تواند در اینجا مفید باشد.

۴. مهارت در ایجاد و بهینه‌سازی مدل‌های یادگیری ماشین:

توانایی در پیاده‌سازی و بهینه‌سازی مدل‌های یادگیری ماشین مهارتی حیاتی است. این شامل انتخاب و تنظیم پارامترها، تجزیه و تحلیل خطاها، و بهبود عملکرد مدل‌ها می‌شود.

۵. آشنایی با اصول علم داده:

اصول علم داده از قبیل تجزیه و تحلیل داده‌ها، استخراج ویژگی‌ها، و پیش‌پردازش داده‌ها در فازهای مختلف این پروژه اهمیت زیادی دارد. نیاز به درک اصول علم داده برای بهترین استفاده از داده‌ها و حصول اطلاعات مفید وجود دارد.

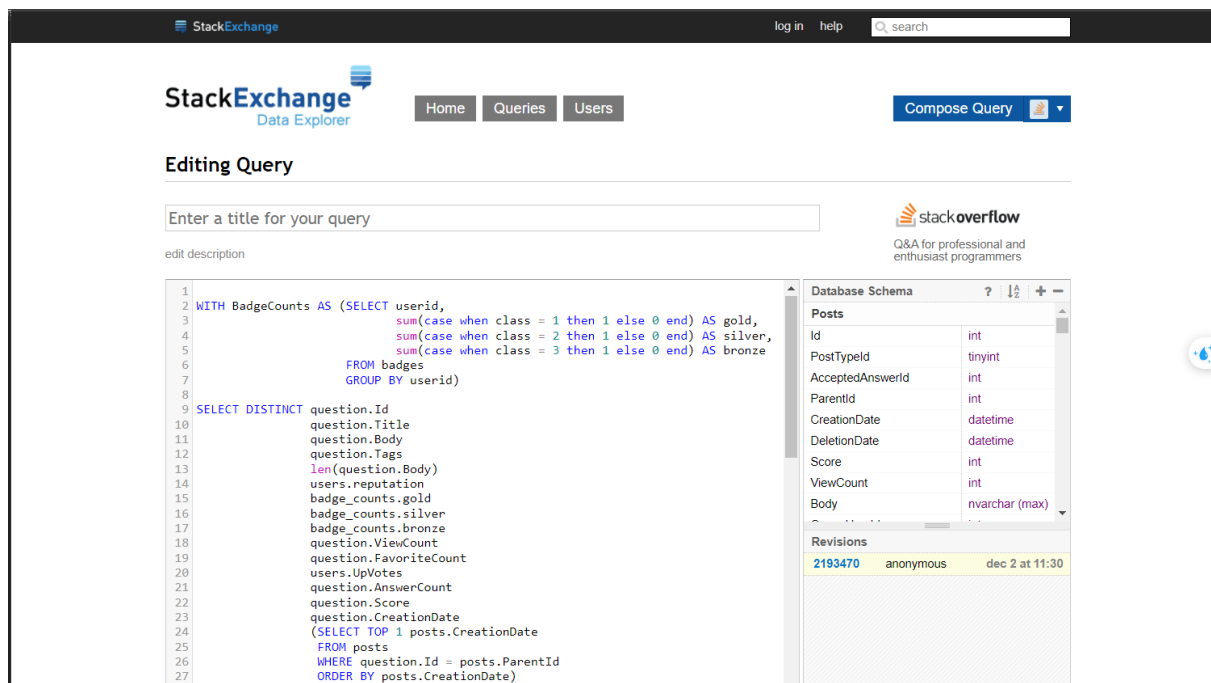
این پیش‌نیازها با همکاری موازی از برنامه‌نویسی و مفاهیم علم داده به طور کامل به انجام موثر پروژه کمک می‌کنند.

فصل دوم: تحلیل، طراحی و اجرای فاز ۱

در این فصل، به جزئیات فنی و طراحی Query برای ایجاد دیتاست مورد نیاز این پروژه می‌پردازیم. در ابتدا با نحوه کارکرد و ویژگی‌های مهم سامانه‌ی <https://data.stackexchange.com/stackoverflow/query> آشنا می‌شویم که به ما امکان اجرای پرس‌وجوهای پیچیده را در پایگاه داده Stack Exchange فراهم می‌کند.

سپس، Query مناسبی طراحی می‌کنیم تا اطلاعات مورد نیاز از پایگاه داده را به دقت استخراج کنیم. در این مرحله، توجه به انتخاب ویژگی‌های مهم از سوالات و جواب‌ها، تاریخچه‌ی فعالیت کاربران، و سایر جزئیات مهم است. همچنین، بررسی نحوه‌ی ارتباط و ارتباطات بین داده‌های مختلف در سامانه، کلیدی برای ساخت یک دیتاست جامع و قابل استفاده است.

پس از طراحی Query، به تفصیل به تحلیل خروجی‌های مورد نظر می‌پردازیم. این شامل دانلود و بررسی خروجی‌ها، اعمال تغییرات لازم در متون سوالات و جواب‌ها، و ذخیره‌سازی داده‌ها به منظور افزودن لیبل‌های مربوط به فاز اول پروژه می‌شود. این فصل نقطه‌ی شروعی جزئیات‌محور برای اجرای موثر پروژه فراهم می‌کند و اساساً پله اول در مسیر تجزیه و تحلیل دقیق داده‌ها و آماده‌سازی آن‌ها برای مراحل بعدی پروژه می‌باشد.



سایت StackExchange

در ابتدا، به توضیح کوئری مورد استفاده برای گرفتن داده‌ها می‌پردازیم. این کوئری به منظور ایجاد یک دیتاست جامع از سوالات مرتبط با زبان برنامه‌نویسی SQL در بازه زمانی مشخص (از ۲۰۱۸-۰۱-۰۱ تا ۲۰۲۰-۱۲-۳۱) طراحی شده است. توضیحات مربوط به اجزای مختلف کوئری عبارتند از:

۱. BadgeCounts Subquery:

این زیر کوئری به شمارش تعداد بلیط‌های طلایی، نقره‌ای و برنزی بر اساس کلاس‌های مختلف، برای هر کاربر در جدول Badges می‌پردازد:

```
WITH BadgeCounts AS (SELECT userid,
                           sum(case when class = 1 then 1 else 0 end) AS gold,
                           sum(case when class = 2 then 1 else 0 end) AS silver,
                           sum(case when class = 3 then 1 else 0 end) AS bronze
                        FROM badges
                        GROUP BY userid)
```

۲. اطلاعات اصلی (Main Query):

- اطلاعات اصلی از جدول Posts برای سوالات انتخاب می شوند.
- اطلاعات مربوط به برچسب ها از جدول PostTags و Tags استخراج می شوند.
- اطلاعات مربوط به بلیط ها از BadgeCounts Subquery استفاده می کنند.
- اطلاعات مربوط به کاربران از جدول Users به همراه اطلاعات بلیط ها انتخاب می شوند.
- اطلاعات مربوط به نظرات کاربران از جدول Comments نیز در نظر گرفته می شوند.

۳. شرایط فیلترینگ:

- سوالات انتخاب شده باید از نوع سوال (PostTypeId = 1) باشند.
- تاریخ ایجاد سوال باید در بازه ی زمانی مشخص شده (از ۲۰۱۸-۰۱-۰۱ تا ۲۰۲۰-۱۲-۳۱) باشد.
- حاوی برچسب SQL باشند.

برچسب SQL داشتن یک شرط ویژه برای گروه ما بود. به این صورت که ما ملزم به جمع آوری پست ها (سوالات) مرتبط با تگ SQL (جامعه SQL) بودیم.

```

SELECT DISTINCT question.Id AS QuestionId,
question.Title AS QuestionTitle,
question.Body AS QuestionBody,
question.Tags AS QuestionTags,
len(question.Body) AS QuestionBodyLength,
users.reputation AS UserReputation,
badge_counts.gold AS GoldBadges,
badge_counts.silver AS SilverBadges,
badge_counts.bronze AS BronzeBadges,
question.ViewCount AS QuestionViewCount,
question.FavoriteCount AS QuestionFavoriteCount,
users.UpVotes AS UserUpVotes,
question.AnswerCount AS AnswerCount,
question.Score AS QuestionScore,
question.CreationDate AS QuestionCreationDate,
(SELECT TOP 1 posts.CreationDate
FROM posts
WHERE question.Id = posts.ParentId
ORDER BY posts.CreationDate) AS FirstAnswerCreationDate,
(SELECT TOP 1 posts.CreationDate
FROM posts
WHERE question.AcceptedAnswerId = posts.Id) AS AcceptedAnswerCreationDate,
DATEDIFF(day, question.CreationDate, (SELECT TOP 1 posts.CreationDate
FROM posts
WHERE question.Id = posts.ParentId
ORDER BY posts.CreationDate)) as FirstAnswerIntervalDays,
DATEDIFF(day, question.CreationDate, (SELECT TOP 1 posts.CreationDate
FROM posts
WHERE question.AcceptedAnswerId = posts.Id)) as AcceptedAnswerIntervalDays

```

ادامه ی فایل SQL

۴. گروه‌بندی و مرتب‌سازی:

– سوالات بر اساس امتیاز Score به ترتیب نزولی مرتب شده و از جدول اول ۱۰۰۰ سطر انتخاب می‌شوند.

```

FROM Posts question
    LEFT JOIN
    PostTags post_tag ON question.Id = post_tag.PostId
    LEFT JOIN
    Tags tags ON post_tag.TagId = tags.Id
    JOIN
    BadgeCounts badge_counts ON question.owneruserid = badge_counts.userid
    JOIN
    users users ON question.owneruserid = users.id
    JOIN
    Comments comments ON question.Id = comments.PostId

WHERE question.CreationDate ≥ '2018-01-01'
    AND question.CreationDate ≤ '2020-12-31'
    AND question.PostTypeId = 1
    AND question.tags LIKE '%sql%'
GROUP BY question.Id,
    question.Title,
    question.Body,
    question.Tags,
    users.reputation,
    badge_counts.gold,
    badge_counts.silver,
    badge_counts.bronze,
    question.ViewCount,
    question.FavoriteCount,
    question.AnswerCount,
    question.Score,
    question.CreationDate,
    question.AcceptedAnswerId,
    users.UpVotes
ORDER BY question.Score DESC
OFFSET 0 ROWS FETCH NEXT 1000 ROWS ONLY;

```

انتهای فایل SQL

این کوئری اطلاعات مهمی را از سوالات مرتبط با SQL جمع آوری کرده و آماده سازی می کند تا در مراحل بعدی پروژه برای تحلیل و مدل سازی استفاده شود.

1.QueryResults.csv - Excel

File Home Insert Draw Page Layout Formulas Data Review View Help Foxt PDF Tell me what you want to do

Clipboard Font Alignment Number

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

QuestionId

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|----|------------|------------------------|-------------|------------|------------|------------|-----------|-------------|------------|------------|------------|-----------|------------|------------|------------|------------|------------|
| 1 | QuestionId | QuestionTi | QuestionBc | QuestionTe | QuestionBk | UserReputi | UserGoldB | UserSilverE | UserBronze | QuestionVi | QuestionFe | UserUpVot | QuestionAi | QuestionSc | QuestionCr | FirstAnswe | AcceptedAl |
| 2 | 50093144 | MySQL 8.0 <p>I can't | <mysql><n | | 2789 | 7655 | 3 | 10 | 14 | 1015467 | 0 | 7 | 35 | 755 | ##### | ##### | ##### |
| 3 | 49194719 | Authentica <p>I am | <mysql><nr | | 355 | 17755 | 9 | 53 | 82 | 1043189 | 0 | 715 | 38 | 671 | ##### | ##### | ##### |
| 4 | 50379839 | Connector <p>I try to | <java><my | | 2204 | 4667 | 3 | 9 | 10 | 670288 | 0 | 17 | 25 | 450 | ##### | ##### | ##### |
| 5 | 65456814 | Docker (Ap <p>I'm | <mysql><d | | 903 | 5470 | 4 | 31 | 52 | 405092 | 0 | 34 | 24 | 445 | ##### | ##### | ##### |
| 6 | 50177216 | How to gra <p>Tried< | <mysql><nr | | 753 | 2820 | 3 | 8 | 8 | 720528 | 0 | 7 | 24 | 282 | ##### | ##### | ##### |
| 7 | 50690076 | phpMyAdn <p>I have | <php><mys | | 620 | 5235 | 7 | 21 | 35 | 474164 | 0 | 9 | 22 | 272 | ##### | ##### | ##### |
| 8 | 52364415 | PHP with N <p>I'm | <php><mys | | 542 | 2311 | 2 | 12 | 16 | 420053 | 0 | 1 | 8 | 228 | ##### | ##### | ##### |
| 9 | 55038942 | FATAL: pas <p>I | <postgres | | 306 | 1849 | 2 | 8 | 9 | 589784 | 0 | 0 | 18 | 183 | ##### | ##### | ##### |
| 10 | 50557234 | Authentica <p>I am | <python><i | | 768 | 4649 | 13 | 43 | 75 | 307207 | 0 | 215 | 29 | 172 | ##### | ##### | ##### |
| 11 | 50026939 | php mysql <p>I am | <php><mys | | 1401 | 2568 | 2 | 16 | 20 | 310409 | 0 | 91 | 18 | 162 | ##### | ##### | ##### |
| 12 | 61523447 | Skipping ac <p>Good | <linux><po | | 672 | 1601 | 2 | 6 | 3 | 108093 | 0 | 0 | 4 | 160 | ##### | ##### | ##### |
| 13 | 55300370 | PostgreSQL <p>To | <postgres | | 614 | 36202 | 11 | 45 | 69 | 79782 | 0 | 1709 | 1 | 150 | ##### | ##### | ##### |
| 14 | 57879150 | How can I s <p>I want | <node.js><i | | 9671 | 12524 | 73 | 249 | 449 | 296271 | 0 | 463 | 28 | 144 | ##### | ##### | ##### |
| 15 | 48218065 | Objects cre <p>I can't | <python><i | | 1064 | 1293 | 2 | 7 | 5 | 143516 | 0 | 0 | 11 | 128 | ##### | ##### | ##### |
| 16 | 58461178 | How to fix <p>Iâ€™m | <postgres | | 613 | 1393 | 2 | 7 | 8 | 116287 | 0 | 5 | 15 | 126 | ##### | ##### | ##### |
| 17 | 57665645 | Server retu <p>My | <mysql><sr | | 277 | 1240 | 2 | 6 | 7 | 88473 | 0 | 54 | 5 | 113 | ##### | ##### | ##### |
| 18 | 54513450 | A strange c | <sql><sql-s | | 1122 | 839 | 1 | 6 | 7 | 8047 | 0 | 1 | 3 | 105 | ##### | ##### | ##### |
| 19 | 60864367 | #1030 - Go <p>Create | <mysql><cp | | 355 | 1003 | 2 | 8 | 7 | 156659 | 0 | 4 | 6 | 100 | ##### | ##### | ##### |
| 20 | 52032739 | Loading cla <p>This is | <mysql><jc | | 335 | 1021 | 2 | 7 | 7 | 278424 | 0 | 0 | 26 | 100 | ##### | ##### | ##### |
| 21 | 59993844 | ERROR: Loc <p>I don't | <mysql><ccl | | 2344 | 1121 | 1 | 6 | 4 | 257829 | 0 | 0 | 15 | 99 | ##### | ##### | ##### |
| 22 | 51335298 | Concepts o <p>Can | <python><i | | 1052 | 1009 | 2 | 10 | 11 | 34417 | 0 | 0 | 2 | 97 | ##### | ##### | ##### |
| 23 | 52423595 | mysqldum <p>I want | <mysql> | | 631 | 14737 | 6 | 67 | 65 | 61214 | 0 | 64 | 3 | 93 | ##### | ##### | ##### |

1.QueryResults

Ready Accessibility: Unavailable

فایل خام گرفته شده از سایت

پس از جمع آوری و ذخیره‌ی اطلاعات از وب‌سایت، امکان بهبود و توسعه‌ی دیگری برای افزودن اطلاعات مورد نیاز وجود دارد. در این مرحله، با بهره‌گیری از مهارت‌های برنامه‌نویسی، ابتدا متون سوالات را مورد بررسی قرار می‌دهیم و عملیات‌هایی چون حذف تگ‌های HTML، محاسبه تعداد خطوط کد یا LOC (Lines of Code)، شمارش تعداد URL‌ها و تصاویر استفاده شده در متن سوال، محاسبه تعداد کاراکترهای سوالات، و در نهایت، محاسبه AcceptRate را اجرا می‌کنیم.

در ادامه این فرآیند، می‌توانیم اطلاعات جدیدی را به دست آوریم. در نهایت، با اتمام این فرآیند، یک فایل نهایی حاصل از تمامی اطلاعات جمع‌آوری‌شده و افزوده‌شده تهیه می‌شود که برای مراحل بعدی پروژه به عنوان پایه و اطلاعات پردازش‌شده قابل استفاده است.

2.PythonResults.csv - Excel

File Home Insert Draw Page Layout Formulas Data Review View Help Foxt PDF Tell me what you want to do Share

Paste Font Alignment Number Conditional Formatting Format as Table Cell Styles Insert Delete Format Sort & Find & Filter Select

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

A1 QuestionId

| | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
|----|-----------|-----|-------|------------|----------|----------|----------|---------|------------|------|---------|---------|----------------|-----------------|------------------|------------------|-----------------|--------------|
| 1 | URLImageC | LOC | | UserReputi | UserGolg | UserSilv | UserBror | Questic | QuestionVi | Ques | UserUp\ | Questic | Questi | QuestionCreatio | FirstAnswerCreat | AcceptedAnswerCr | FirstAnswerInte | AcceptedAnsw |
| 2 | 4 | 51 | 7655 | 3 | 10 | 14 | 64 | 1015467 | 0 | 7 | 35 | 755 | 18/04/30 2:04 | 18/05/02 9:58 | 18/05/02 9:58 | | 2 | 2 |
| 3 | 0 | 0 | 17755 | 9 | 53 | 82 | 61 | 1043189 | 0 | 715 | 38 | 671 | 18/03/09 13:19 | 18/03/12 5:08 | 18/03/12 5:08 | | 3 | 3 |
| 4 | 0 | 37 | 4667 | 3 | 9 | 10 | 73 | 670288 | 0 | 17 | 25 | 450 | 18/05/16 21:02 | 18/05/20 19:57 | 18/05/20 19:57 | | 4 | 4 |
| 5 | 1 | 16 | 5470 | 4 | 31 | 52 | 54 | 405092 | 0 | 34 | 24 | 445 | 20/12/26 13:20 | 21/01/06 9:04 | 21/01/06 9:04 | | 11 | 11 |
| 6 | 0 | 2 | 2820 | 3 | 8 | 8 | 53 | 720528 | 0 | 7 | 24 | 282 | 18/05/04 14:23 | 18/05/06 8:17 | | | 2 | |
| 7 | 0 | 2 | 5235 | 7 | 21 | 35 | 63 | 474164 | 0 | 9 | 22 | 272 | 18/06/04 23:15 | 18/06/07 16:58 | 18/06/07 16:58 | | 3 | 3 |
| 8 | 0 | 0 | 2311 | 2 | 12 | 16 | 67 | 420053 | 0 | 1 | 8 | 228 | 18/09/17 9:17 | 18/09/17 9:18 | | | 0 | |
| 9 | 0 | 0 | 1849 | 2 | 8 | 9 | 71 | 589784 | 0 | 0 | 18 | 183 | 19/03/07 8:12 | 19/03/07 8:41 | 19/03/07 8:41 | | 0 | 0 |
| 10 | 1 | 8 | 4649 | 13 | 43 | 75 | 42 | 307207 | 0 | 215 | 29 | 172 | 18/05/27 22:53 | 18/05/27 23:05 | 18/05/27 23:05 | | 0 | 0 |
| 11 | 1 | 22 | 2568 | 2 | 16 | 20 | 51 | 310409 | 0 | 91 | 18 | 162 | 18/04/25 16:15 | 18/04/25 16:54 | 18/04/25 16:54 | | 0 | 0 |
| 12 | 3 | 3 | 1601 | 2 | 6 | 3 | 50 | 108093 | 0 | 0 | 4 | 160 | 20/04/30 12:31 | 20/05/01 20:08 | | | 1 | |
| 13 | 2 | 4 | 36202 | 11 | 45 | 69 | 71 | 79782 | 0 | 1709 | 1 | 150 | 19/03/22 13:10 | 19/03/22 13:31 | 19/03/22 13:31 | | 0 | 0 |
| 14 | 4 | 106 | 12524 | 73 | 249 | 449 | 65 | 296271 | 0 | 463 | 28 | 144 | 19/09/10 22:59 | 20/01/23 16:12 | | | 135 | |
| 15 | 0 | 15 | 1293 | 2 | 7 | 5 | 52 | 143516 | 0 | 0 | 11 | 128 | 18/01/12 0:52 | 18/01/12 1:17 | 18/01/12 1:17 | | 0 | 0 |
| 16 | 0 | 2 | 1393 | 2 | 7 | 8 | 71 | 116287 | 0 | 5 | 15 | 126 | 19/10/19 6:39 | 19/10/19 7:19 | | | 0 | |
| 17 | 0 | 0 | 1240 | 2 | 6 | 7 | 81 | 88473 | 0 | 54 | 5 | 113 | 19/08/26 22:29 | 19/08/27 2:27 | 19/08/27 2:27 | | 1 | 1 |
| 18 | 1 | 13 | 839 | 1 | 6 | 7 | 61 | 8047 | 0 | 1 | 3 | 105 | 19/02/04 9:43 | 19/02/04 11:02 | | | 0 | |
| 19 | 1 | 0 | 1003 | 2 | 8 | 7 | 44 | 156659 | 0 | 4 | 6 | 100 | 20/03/26 9:36 | 20/04/30 8:52 | 20/05/17 8:56 | | 35 | 52 |
| 20 | 0 | 4 | 1021 | 2 | 7 | 7 | 70 | 278424 | 0 | 0 | 26 | 100 | 18/08/27 4:52 | 18/09/19 12:23 | | | 23 | |
| 21 | 0 | 38 | 1121 | 1 | 6 | 4 | 38 | 257829 | 0 | 0 | 15 | 99 | 20/01/30 20:15 | 20/02/02 15:45 | | | 3 | |
| 22 | 0 | 14 | 1009 | 2 | 10 | 11 | 68 | 34417 | 0 | 0 | 2 | 97 | 18/07/14 4:44 | 20/01/26 17:15 | | | 561 | |
| 23 | 0 | 1 | 14737 | 6 | 67 | 65 | 68 | 61214 | 0 | 64 | 3 | 93 | 18/09/20 11:00 | 18/09/20 11:00 | 18/09/20 11:00 | | 0 | 0 |

2.PythonResults

Ready Accessibility: Unavailable

فایل نهایی فاز ۱

فصل سوم: تحلیل، طراحی و اجرای فاز ۲

در فصل قبلی، ما با مراحل اولیه‌ی جمع‌آوری داده‌ها از وب‌سایت مرتبط با Stack Overflow آشنا شدیم و دیتاست اولیه‌ای را به دست آوردیم. حالا اطلاعات حاصل از فصل قبلی را با مشارکت گروهی از افراد برجسب‌زنی می‌نماییم. این مرحله از اهمیت بسیار زیادی برخوردار است چرا که اعتبار لیبل‌ها بر کیفیت مدل نهایی تأثیر گذار است. ما به صورت خودکار و هماهنگ با افراد برجسب‌زن، داده‌ها را به گروه‌های مختلفی تقسیم بندی می‌کنیم.

در پایان این فصل، با داشتن داده‌های پاکسازی‌شده و برجسب‌زده شده، به سمت آماده‌سازی داده‌های آموزش و آزمون برای مدل‌های یادگیری ماشین متجه خواهیم شد. این مرحله از پروژه اساسی است و مؤثر بر کیفیت و قدرت پیش‌بینی مدل‌های آموزش یافته است.

در این مرحله، تمام داده‌ها بر اساس تگ، عنوان سوال و متن سوال، جهت ارزیابی قرار می‌گیرند. دو اعضای گروه به عنوان ارزیابان، هر داده را بررسی نموده و بر اساس نظر شخصی خود، یک عدد از مجموعه اعداد ۱ تا ۴۷ را انتخاب می‌نمایند. در صورت تطابق نظر دو افراد، آن عدد به عنوان انتخاب قطعی در نظر گرفته می‌شود؛ در غیر این صورت، به نظر دیگری انتقال می‌یابد. در موارد عدم توافق، رای اکثریت اعضا تصمیم‌گیری می‌شود. اگر سه نفر به نتایج متفاوت دست یابند، یکی از آن‌ها انتخاب شده و در یکی از گروه‌های ساده (از ۱ تا ۱۴)، متوسط (از ۱۵ تا ۳۱) یا سخت (۳۲ به بالا) قرار می‌گیرد. این اعداد برچسب زنی از جدول زیر به دست می‌آیند:

| | | |
|--------------|---|---|
| Intermediate | | Difference between two packages |
| | Questions stating the answer of the problem but still inquires about more efficient answer | Looking for Appropriate way |
| | | Analyzing different alternatives |
| | Questions about a system's computational cost, space utilization, or other resource usages | Efficient way |
| | | Performance, optimization, accuracy |
| | | Memory related |
| | Questions requiring conceptual thinking in response to any programming structure, API, or design principle | Reverse programming |
| | | Underlying philosophy of any programming construction |
| | | Design pattern |
| | | Feasibility study |
| | Required Testing Related Knowledge | Question about built-in documentation in details |
| | | Automated testing related problem |
| Advanced | Questions about critical challenges that require in-depth technical expertise or logical reasoning to solve | Requires knowledge of Testing |
| | | Critical problems where solution needs in-depth programming knowledge or conceptual thinking. |
| | | Multiple question, Solution needs in-depth programming knowledge or logical thinking. |
| | Questions that require advanced in-depth knowledge of internal language structure | Multiple question and in depth knowledge needed |
| | | In-depth knowledge of internal language structure. |
| | Questions that deals with infrequently/rarely used framework/API | In-depth knowledge of packages |
| | | Deals with infrequently/rarely used framework |
| | Related to real life scenario | Deals with deprecated framework |
| | | Efficiency related Question in real life scenario |
| | | Optimization in Real life scenario |

منبع اصلی برچسب زنی

برای انجام این برچسب‌زنی، از برنامه‌ای استفاده می‌شود که در هر مرحله یک سوال را نمایش می‌دهد و منتظر دریافت جواب از ارزیابان می‌ماند؛ این امر منجر به افزایش سرعت برچسب‌زنی می‌گردد. این روش به صورت موثری به سرعت و دقت در فرآیند برچسب‌زنی کمک می‌کند.

```

C:\Windows\System32\cmd.e x + v
<----- Title ----->
SQL Server join where not exist on other table
<----- Body ----->
+-----+ +-----+ +-----+
| Service | | Asset | | AssetService |
+-----+ +-----+ +-----+
| Id | Name | | Id | Name | | AssetId | ServiceId |
+-----+ +-----+ +-----+
| 1 | Service 1 | | 1 | Asset 1 | | 1 | 1 |
| 2 | Service 2 | | 2 | Asset 2 | | 1 | 2 |
| 3 | Service 3 | | 3 | Asset 3 | | 2 | 2 |
| | | | | | | 2 | 3 |
+-----+ +-----+ +-----+

So I have these tables. I want to get the Services that is not on AssetService where AssetId = 1
Like this:
+-----+
| Service |
| Id | Name |
+-----+
| 3 | Service 3 |
+-----+

Is this possible with just inner/left/right join? because I already tried different combinations of inner join but it's
not working, like this inner join Asset a on a.Id != as.AssetId. I event tried left and right join.
Can somebody help me?
Thanks.

<----- ID ----->

[Question: 48646568]
[Previous Question: None]

<----- Index ----->

What is index 1-47 ? [1/1] > 1

```

نمونه نمایش سوال برای برچسب زدن

همچنین، طراحی برنامه به گونه‌ای است که در صورت ایجاد قطعیت یا اتمام عملیات برچسب‌زنی، داده‌های لیبل زده شده از دست نرود. این فایل، که با نام “2.5.keeper.pickle” شناخته می‌شود، به این امر کمک می‌کند. به این صورت که بعد از هر ۵ بار برچسب‌زدن، یک بار تمامی داده‌ها در این فایل ذخیره می‌شود. در صورتی که برنامه به هر دلیلی بسته شود یا به خطایی برخورد کند، تمامی اطلاعات گذشته در این فایل موجود هستند و داده‌ها از بین نمی‌روند. این اقدام امنیت و اطمینان از ادامه‌ی فرآیند برچسب‌زنی را تضمین می‌کند.

```

# Save the labels after every 5th question to avoid memory issues
if x % 5 == 0:
    with open(temp_keeper_name, 'wb') as file:
        dump(keeper, file)

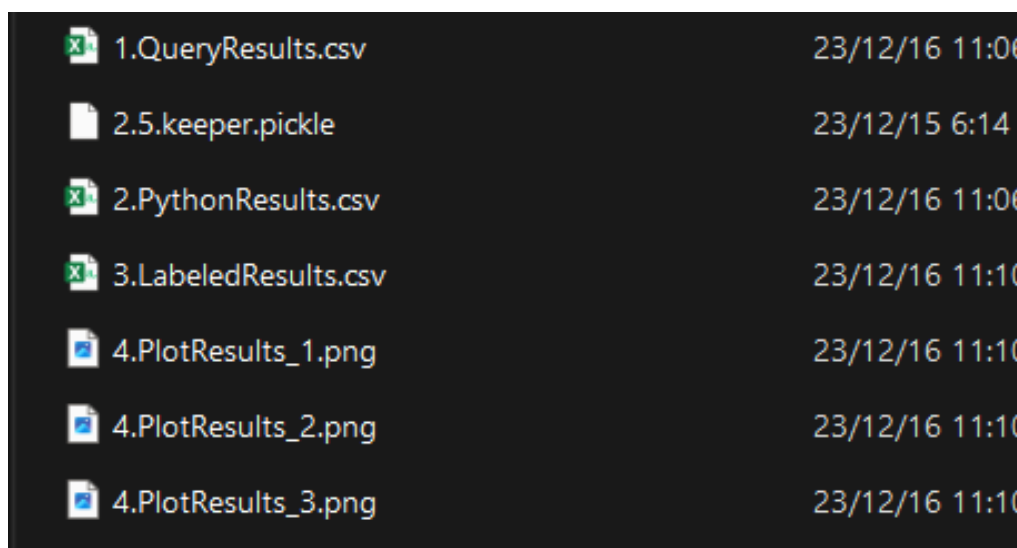
```

کد مربوط به این بخش

در پایان، داده‌های لیبل زده شده به عنوان دو ستون جدید به داده‌ها افزوده می‌شوند و خروجی فاز دو تولید می‌شود (دو ستون آخر).

| | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|----|-----------|-----------|-------------|-----------|-----------|-----------|-----------|-----------|-----------|------------|------------|-------------|-----------|-------------|-----------|------------|---------------|------------|---|
| 1 | UserReput | UserGoldB | UserSilverF | UserBronz | QuestionA | QuestionV | QuestionF | UserUpVot | QuestionA | QuestionSc | QuestionCr | FirstAnswer | AcceptedA | FirstAnswer | AcceptedA | QuestionLa | QuestionLabel | Definition | |
| 2 | 7655 | 3 | 10 | 14 | 64 | 1015467 | 0 | 7 | 35 | 755 | ##### | ##### | ##### | 2 | 2 | Advanced | 32 | | |
| 3 | 17755 | 9 | 53 | 82 | 61 | 1043189 | 0 | 715 | 38 | 671 | ##### | ##### | ##### | 3 | 3 | Advanced | 32 | | |
| 4 | 4667 | 3 | 9 | 10 | 73 | 670288 | 0 | 17 | 25 | 450 | ##### | ##### | ##### | 4 | 4 | Intermedia | 31 | | |
| 5 | 5470 | 4 | 31 | 52 | 54 | 405092 | 0 | 34 | 24 | 445 | ##### | ##### | ##### | 11 | 11 | Advanced | 37 | | |
| 6 | 2820 | 3 | 8 | 8 | 53 | 720528 | 0 | 7 | 24 | 282 | ##### | ##### | ##### | 2 | | Basic | 9 | | |
| 7 | 5235 | 7 | 21 | 35 | 63 | 474164 | 0 | 9 | 22 | 272 | ##### | ##### | ##### | 3 | 3 | Basic | 14 | | |
| 8 | 2311 | 2 | 12 | 16 | 67 | 420053 | 0 | 1 | 8 | 228 | ##### | ##### | ##### | 0 | | Basic | 3 | | |
| 9 | 1849 | 2 | 8 | 9 | 71 | 589784 | 0 | 0 | 18 | 183 | ##### | ##### | ##### | 0 | 0 | Basic | 13 | | |
| 10 | 4649 | 13 | 43 | 75 | 42 | 307207 | 0 | 215 | 29 | 172 | ##### | ##### | ##### | 0 | 0 | Advanced | 44 | | |
| 11 | 2568 | 2 | 16 | 20 | 51 | 310409 | 0 | 91 | 18 | 162 | ##### | ##### | ##### | 0 | 0 | Advanced | 44 | | |
| 12 | 1601 | 2 | 6 | 3 | 50 | 108093 | 0 | 0 | 4 | 160 | ##### | ##### | ##### | 1 | | Basic | 14 | | |
| 13 | 36202 | 11 | 45 | 69 | 71 | 79782 | 0 | 1709 | 1 | 150 | ##### | ##### | ##### | 0 | 0 | Intermedia | 19 | | |
| 14 | 12524 | 73 | 249 | 449 | 65 | 296271 | 0 | 463 | 28 | 144 | ##### | ##### | ##### | 135 | | Intermedia | 30 | | |
| 15 | 1293 | 2 | 7 | 5 | 52 | 143516 | 0 | 0 | 11 | 128 | ##### | ##### | ##### | 0 | 0 | Basic | 6 | | |
| 16 | 1393 | 2 | 7 | 8 | 71 | 116287 | 0 | 5 | 15 | 126 | ##### | ##### | ##### | 0 | | Basic | 9 | | |
| 17 | 1240 | 2 | 6 | 7 | 81 | 88473 | 0 | 54 | 5 | 113 | ##### | ##### | ##### | 1 | 1 | Basic | 14 | | |
| 18 | 839 | 1 | 6 | 7 | 61 | 8047 | 0 | 1 | 3 | 105 | ##### | ##### | ##### | 0 | | Basic | 2 | | |
| 19 | 1003 | 2 | 8 | 7 | 44 | 156659 | 0 | 4 | 6 | 100 | ##### | ##### | ##### | 35 | 52 | Advanced | 37 | | |
| 20 | 1021 | 2 | 7 | 7 | 70 | 278424 | 0 | 0 | 26 | 100 | ##### | ##### | ##### | 23 | | Advanced | 38 | | |
| 21 | 1121 | 1 | 6 | 4 | 38 | 257829 | 0 | 0 | 15 | 99 | ##### | ##### | ##### | 3 | | Basic | 3 | | |
| 22 | 1009 | 2 | 10 | 11 | 68 | 34417 | 0 | 0 | 2 | 97 | ##### | ##### | ##### | 561 | | Intermedia | 27 | | |
| 23 | 14737 | 6 | 67 | 65 | 68 | 61214 | 0 | 64 | 3 | 93 | ##### | ##### | ##### | 0 | 0 | Advanced | 32 | | |

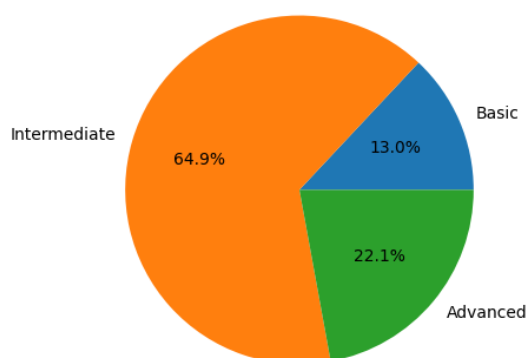
خروجی فایل فاز ۲



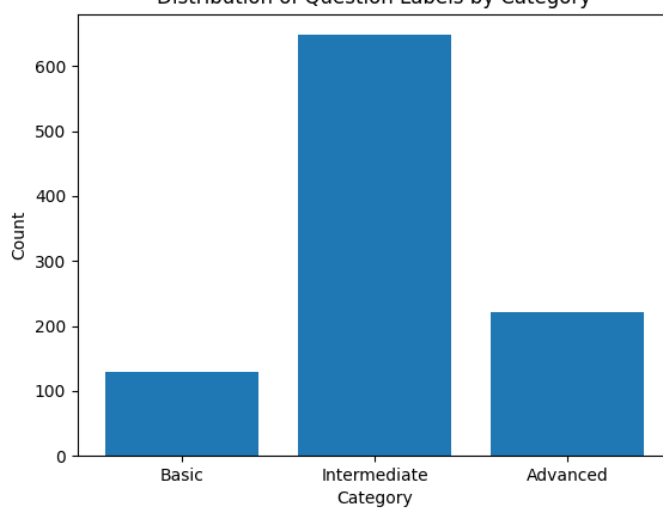
تمامی فایل های ساخته شده تا پایان فاز ۲

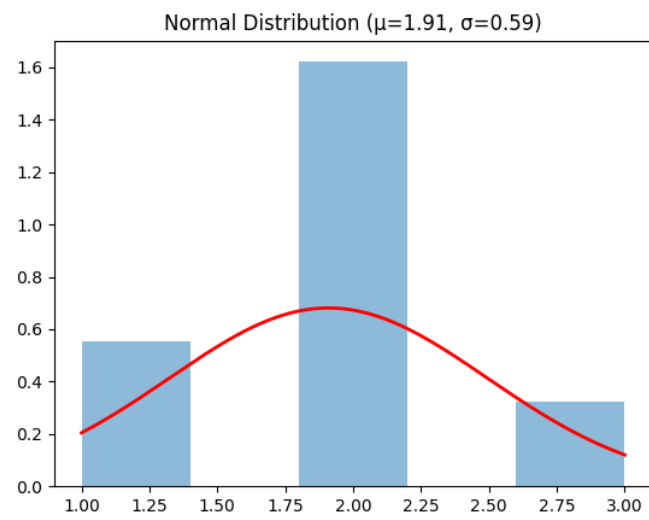
به علاوه، نمودارهایی برای نمایش بهتر خروجی ها در نظر گرفته شده اند:

Distribution of Question Labels by Category



Distribution of Question Labels by Category





فصل چهارم: تحلیل، طراحی و اجرای فاز ۳

در فصل‌های گذشته، ما به مرحله‌های جمع‌آوری داده‌ها، پیش‌پردازش آن‌ها و برجسب‌زنی اطلاعات پرداختیم. حالا، با دستیابی به یک دیتاست کامل و آماده، در فصل چهارم به تحلیل و پردازش داده‌ها به صورت جزئیات می‌پردازیم. در این فصل، ابتدا به تبدیل خروجی فاز دو به موارد خواسته شده در متن پروژه می‌پردازیم. این مرحله شامل تحلیل و استخراج ویژگی‌های مهم از داده‌ها، اطلاعات توافق و عدم توافق در ارزیابی‌ها، و سایر موارد تحلیلی است.

پس از تبدیل خروجی فاز دو، به پردازش متن سوالات می‌پردازیم و تحلیل روی آن‌ها انجام می‌دهیم. از تکنیک‌های پردازش متن استفاده می‌کنیم تا الگوها و اطلاعات ارزشمندی را از متون سوالات استخراج کنیم.

یکی از بخش‌های حیاتی این فصل، تبدیل متون توضیحات به نمودارهای عددی و آماده‌سازی داده‌ها برای آموزش مدل‌های یادگیری ماشین است. با استفاده از توابعی چون Logistic، CART، SVC، Naïve Bayes، MLP، Regression و XGBoost، به آموزش مدل‌ها می‌پردازیم.

نهایتاً، خروجی‌های این فصل را برای مقایسه مدل‌ها و تفسیر نتایج به فصل بعد منتقل می‌کنیم.

در این مرحله از پروژه، نیاز داریم یک متن واحد از محتوای موجود در ستون‌های QuestionBody، QuestionTitle و QuestionTags به دست آورده و در یک ستون جدید با نام MergedText ذخیره کنیم:

```
# Merge QuestionTags, QuestionTitle and QuestionBody fields into a new feature 'MergedText'
df['MergedText'] = (df['QuestionTags'].astype(str) + ', ' + df['QuestionTitle'].astype(str) + ', ' +
                    df['QuestionBody'].astype(str))
```

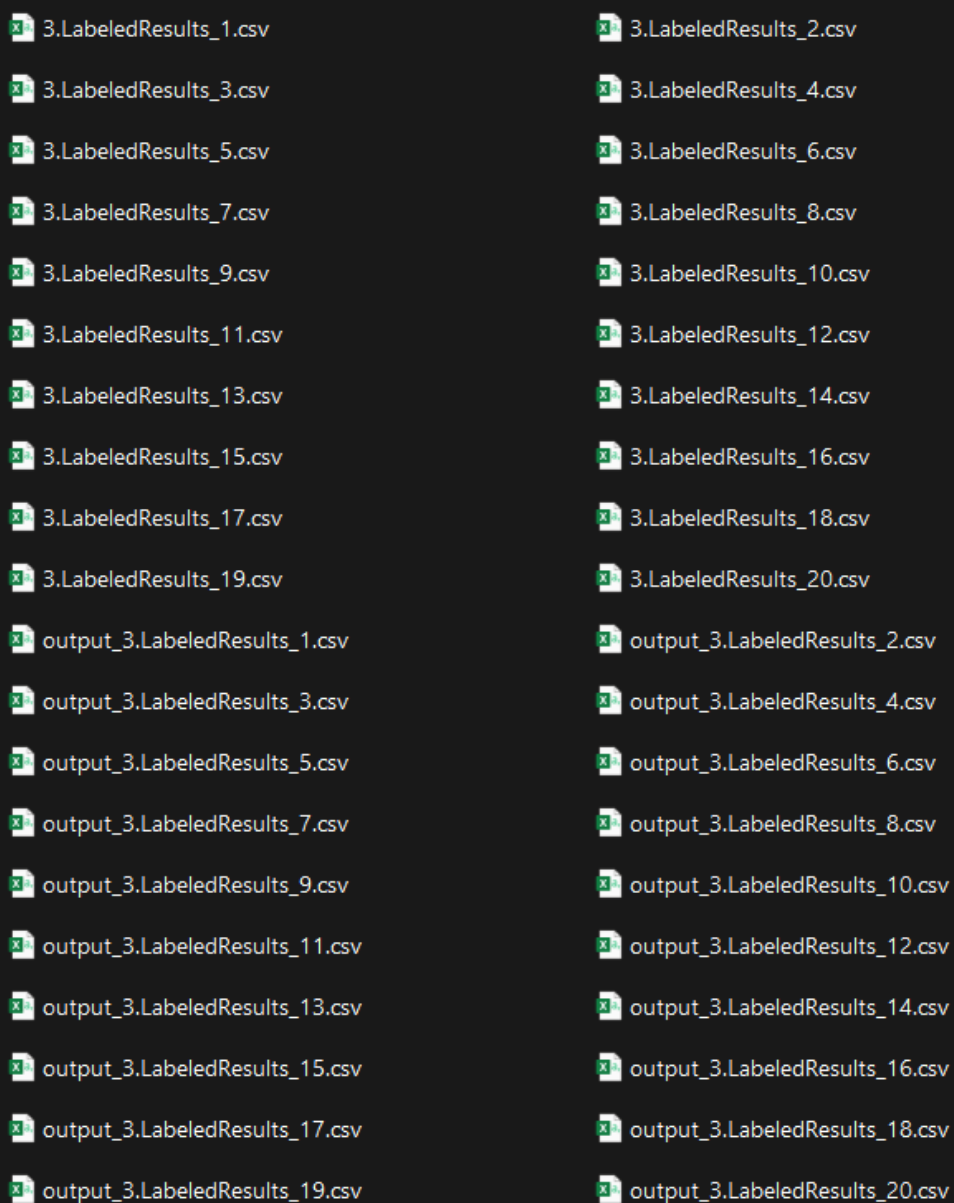
سپس، برای هر داده در ستون MergedText، عملیات preprocess را روی آن اعمال کرده و نتیجه را در یک ستون جدید به نام ProcessedText ذخیره می‌کنیم:

```
# Apply text preprocessing to the 'MergedText' column
df['ProcessedText'] = tqdm(
    df['MergedText'].progress_apply(self.preprocess_given_text, stop_words=stop_words, ps=ps),
    total=len(df))
```

نکته:

در ابتدای پروژه، از دیتاست تستی استفاده میشد. در انتها امکان عوض کردن دیتاست اصلی با دیتاست تستی ممکن نبود. این به دلیل این بود که دیتاست تستی تنها دارای ۱۰ داده بود ولی دیتاست اصلی شامل ۱۰۰۰ داده بود. انجام عملیات preprocessing روی تعداد زیادی داده به طور مستقیم زمان بسیار زیادی را می‌طلبد و همچنین حجم قابل توجهی از حافظه RAM را اشغال می‌کرد. برای حل این مشکلات، توابعی به برنامه افزوده شدند که با استفاده از تکنیک‌های برنامه‌نویسی، این مشکلات را حل کنند. یکی از این تکنیک‌ها، استفاده از روش تقسیم و غلبه بود که فایل CSV اصلی را به چندین بخش تقسیم کرد. هر بخش تعداد داده یکسانی را شامل می‌شد. سپس عملیات preprocessing بر روی هر بخش اعمال شد و در نهایت، تمام بخش‌ها با یکدیگر ادغام شدند. این کار با استفاده از توابع split_csv و merge_csv انجام شد. همچنین برای بهبود قابلیت‌ها و زیبایی توابع، شمارنده و progressbar به آن‌ها افزوده شد. در این روند، فایل اصلی ابتدا به ۲۰ فایل فرعی (در دایرکتوری TextProcessedFiles) تقسیم شد. سپس، هر یک از این فایل‌ها به ترتیب بررسی شد و خروجی‌ها در فایل‌های جدیدی ذخیره شدند.

استفاده از فایل‌های فرعی در اینجا به این معناست که هر فایل به طور مستقل بررسی و پردازش می‌شود. این رویکرد این امکان را فراهم می‌کند که در صورت بروز هر گونه خطا یا بسته شدن ناگهانی برنامه، تنها فایل در حال پردازش از دست برود و داده‌های دیگر از بین نرود. این مزیت به اجرای مطمئن‌تر و ادامه‌پذیرتر عملیات preprocessing کمک می‌کند.



3.LabeledResults_1.csv
3.LabeledResults_2.csv
3.LabeledResults_3.csv
3.LabeledResults_4.csv
3.LabeledResults_5.csv
3.LabeledResults_6.csv
3.LabeledResults_7.csv
3.LabeledResults_8.csv
3.LabeledResults_9.csv
3.LabeledResults_10.csv
3.LabeledResults_11.csv
3.LabeledResults_12.csv
3.LabeledResults_13.csv
3.LabeledResults_14.csv
3.LabeledResults_15.csv
3.LabeledResults_16.csv
3.LabeledResults_17.csv
3.LabeledResults_18.csv
3.LabeledResults_19.csv
3.LabeledResults_20.csv
output_3.LabeledResults_1.csv
output_3.LabeledResults_2.csv
output_3.LabeledResults_3.csv
output_3.LabeledResults_4.csv
output_3.LabeledResults_5.csv
output_3.LabeledResults_6.csv
output_3.LabeledResults_7.csv
output_3.LabeledResults_8.csv
output_3.LabeledResults_9.csv
output_3.LabeledResults_10.csv
output_3.LabeledResults_11.csv
output_3.LabeledResults_12.csv
output_3.LabeledResults_13.csv
output_3.LabeledResults_14.csv
output_3.LabeledResults_15.csv
output_3.LabeledResults_16.csv
output_3.LabeledResults_17.csv
output_3.LabeledResults_18.csv
output_3.LabeledResults_19.csv
output_3.LabeledResults_20.csv

فایل‌های ساخته شده در این مرحله (خروجی‌ها با output شروع میشوند)

```
# Remove stop words
words = [word for word in text.lower().split() if word.lower() not in stop_words]

# Perform stemming
stemmed_words = [ps.stem(word) for word in words]

# Correct spellings using TextBlob
corrected_text = ' '.join([str(TextBlob(word).correct()) for word in stemmed_words])
```

پیش پردازش متن (شامل حذف stop words، تصحیح غلط املائی و ...)

```
D:\SKU\Term 7\7. Basics Of Data Mining\Homeworks\Homework3 - Project2>python main.py
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\Master\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

Start splitting the SCV !
Finish splitting the SCV !
Start preprocessing on each file >>
(1/40) 3.LabeledResults_1.csv:
6% | ██████████ 3/49 [00:24<06:12, 8.09s/it]
```

نمونه اجرای برنامه با اضافه شدن progressbar

سپس بردار های عددی آن به دست می آید، که در شکل زیر تمامی آن همراه بخش های قبلی نشان داده شده است:

| Q | R | S | T | U | V | W | X | Y | Z | AA | AB | AC | AD | AE | AF | AG |
|----------|----|----------|-------|-----------|-----------|-----------|-----------|---------------------|-----------------------------|---------------|----|----|----|----|----|----|
| Question | Sc | Question | Q | FirstAnsw | AcceptedA | FirstAnsw | AcceptedA | QuestionL | MergedText | ProcessedText | | | | | | |
| 1 | 13 | ##### | ##### | ##### | 0 | 0 | 14 | <postgres><jdbc> | <postgres><job><beaver | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 25 | ##### | ##### | ##### | 13 | 13 | 20 | <google-cloud> | <goose-cloud><sal><goose | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 16 | ##### | ##### | ##### | 1101 | | 10 | <mysql><amazon> | <myself><amazon><rd>, a r | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 15 | ##### | ##### | ##### | 0 | 0 | 10 | <sql><postgres> | <sal><postgres>, select | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 30 | ##### | ##### | ##### | 1 | 1 | 17 | <google-cloud> | <goose-cloud><platform><ci | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 47 | ##### | ##### | ##### | 0 | 0 | 8 | <sql><postgres> | <sal><postgres>, differ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 21 | ##### | ##### | ##### | 1 | 1 | 9 | <postgres> | <postgres>, postgresql | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 60 | ##### | ##### | ##### | 0 | 0 | 13 | <django><python> | <django><sal><pymysql> | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 13 | ##### | ##### | ##### | 364 | | 9 | <python><apache> | <patron><django><postgr | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 40 | ##### | ##### | ##### | 68 | | 13 | <python><django><p | <patron><django><postgr | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 20 | ##### | ##### | ##### | 97 | | 10 | <postgres><docker> | <postgres><doctor><go | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | ##### | ##### | ##### | 907 | | 18 | <cf><sql> | <cf><sal><server><net><stai | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 22 | ##### | ##### | ##### | 0 | 0 | 2 | <sql><amazon> | <sal><amazon><redshift>, c | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 12 | ##### | ##### | ##### | 3 | 3 | 33 | <cf><net> | <cf><net><core><entity><fr | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 33 | ##### | ##### | ##### | 1 | 6 | 14 | <postgres><pgadm | <postgres><pgadmin><q | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 18 | ##### | ##### | ##### | 0 | 0 | 2 | <apache> | <apache><spark><spark><ca | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 24 | ##### | ##### | ##### | | | 32 | <sqlite><poco> | <quite><pock>, known res | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 13 | ##### | ##### | ##### | 5 | 5 | 14 | <python><ssl><pymy | <patron><sal><pymysql> | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 14 | ##### | ##### | ##### | 0 | 0 | 20 | <database><postgre | <database><postgres> | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 16 | ##### | ##### | ##### | 0 | | 10 | <mysql><node.js><se | <myself><node.js><seque | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 51 | ##### | ##### | ##### | 23 | | 14 | <mysql><mysql> | <myself><myself><workber | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 14 | ##### | ##### | ##### | 4 | 9 | 23 | <scala><apache> | <scala><apache><spark><aj | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

داده های جدید (شامل بردار عددی، متن پردازش شده و متن سرهم شده)

```
# Vectorization using CountVectorizer
vectorizer = CountVectorizer(ngram_range=(1, 2))
X_count = vectorizer.fit_transform(df['ProcessedText'])

# Vectorization using TF-IDF
tfidf_vectorizer = TfidfVectorizer(ngram_range=(1, 2))
X_tfidf = tfidf_vectorizer.fit_transform(df['ProcessedText'])

# Add n-gram and TF-IDF features to the DataFrame
df_count = pd.DataFrame(X_count.toarray(), columns=vectorizer.get_feature_names_out())
df_tfidf = pd.DataFrame(X_tfidf.toarray(), columns=tfidf_vectorizer.get_feature_names_out())
```

کد استفاده شده برای محاسبه بردارهای عددی

نکته:

فایل حاصل در این مرحله دارای حجم بسیار بزرگ است، به بیش از ۶۰۰ مگابایت می‌رسد. به عبارت دیگر، داده‌ها تا این نقطه به صورت زیر فراهم شده‌اند:

| | | | |
|-------------------------|-------------------|------------------------|------------|
| TextProcessedFiles | 24/01/27 4:02 AM | File folder | |
| ~\$Report.docx | 24/01/27 6:38 PM | Microsoft Word Doc... | 1 KB |
| 1.QueryResults.csv | 23/12/16 11:06 AM | Microsoft Excel Com... | 2,092 KB |
| 2.5.keeper.pickle | 23/12/15 6:14 PM | PICKLE File | 13 KB |
| 2.PythonResults.csv | 23/12/16 11:06 AM | Microsoft Excel Com... | 1,875 KB |
| 3.LabeledResults.csv | 23/12/16 11:10 AM | Microsoft Excel Com... | 1,885 KB |
| 4.1.PlotResults.png | 23/12/16 11:10 AM | PNG File | 24 KB |
| 4.2.PlotResults.png | 23/12/16 11:10 AM | PNG File | 20 KB |
| 4.3.PlotResults.png | 23/12/16 11:10 AM | PNG File | 24 KB |
| 5.TextProcessedData.csv | 24/01/27 4:02 AM | Microsoft Excel Com... | 5,061 KB |
| 6.VectorizedData.csv | 24/01/27 4:10 AM | Microsoft Excel Com... | 651,318 KB |

فایل‌های موجود تا این مرحله

برای بخش بعدی از پروژه، این داده‌ها قابل استفاده نیستند، زیرا حاوی حجم بسیار بالایی هستند. به منظور مدیریت بهتر حافظه RAM و کنترل مقدار ورودی، از تکنیک‌های برنامه‌نویسی استفاده می‌شود. به عنوان مثال، برای خواندن فایل CSV از ورودی، پارامتر "low_memory" به آن اضافه می‌شود:

```
# Load the CSV file into a DataFrame
df = pd.read_csv(input_file, low_memory=False)
```

در بخش دوم این فاز، داده‌های به دست آمده تا این مرحله در قالب یک فایل موجود است. سپس نیاز است دو فایل "train" و "test" را از روی آن‌ها بسازیم. پس از اعمال تابع مورد نظر روی داده‌ها، دو بخش "x" و "y" را از روی ستون "QuestionLabel" می‌سازیم. سپس آن‌ها را در فایل ذخیره می‌کنیم:

```
# Assume 'QuestionLabel' is the column you want to predict, and 'test_size' is the proportion of the dataset to
# include in the test split
X_train, X_test, y_train, y_test = train_test_split(*arrays: df.drop(labels='QuestionLabel', axis=1),
df['QuestionLabel'], test_size=0.2, random_state=42)
```

تقسیم بندی فایل به دو بخش آموزشی و آزمون

سپس فایل‌های زیر به تمامی فایل‌های قبلی اضافه میشوند:

| | |
|---|------------|
|  7.1.TrainData.csv | 24/01/2020 |
|  7.2.TestData.csv | 24/01/2020 |

در بخش سوم این فاز، فایل‌های آموزشی و آزمون از ورودی خوانده می‌شوند. سپس داده‌های تکراری، خالی و غیرقابل استفاده از آن حذف می‌شود. سپس "x" و "y" طبق ستون‌های "ProcessedText" و "QuestionLabel" جدا شده و از آن‌ها داده‌های آزمون و تست خوانده می‌شوند:

```
# Separate features and labels
x_train, y_train = train_df['ProcessedText'], train_df['QuestionLabel']
x_test, y_test = test_df['ProcessedText'], test_df['QuestionLabel']
```

برای بعضی از مدل‌ها، نیاز هست که متن توسط TF-IDF وکتورایز شود:

```
# Vectorize the text data using TF-IDF
vectorizer = TfidfVectorizer()
x_train_tfidf = vectorizer.fit_transform(x_train)
x_test_tfidf = vectorizer.transform(x_test)
```

سپس در این مرحله، مدل‌ها را تعریف می‌کنیم و برای هر کدام، داده‌ها را روی آن‌ها اعمال کرده و آموزش می‌دهیم. سپس مجموعه آزمون را روی آن تست می‌کنیم. پس از این مرحله، با استفاده از توابع آماده کتابخانه‌های مربوط، موارد خواسته شده از جمله دقت، صحت، F1 Score و Recall را محاسبه می‌کنیم:

```
# Initialize the classifiers
models = {
    'Naive Bayes': MultinomialNB(),
    'SVC': SVC(),
    'CART': DecisionTreeClassifier(),
    'Logistic Regression': LogisticRegression(),
    'MLP': MLPClassifier(max_iter=500),
    'XGBoost': xgb.XGBClassifier()
}
```

سپس نتایج را در یک دیکشنری ذخیره می‌کنیم. از دیکشنری ساخته‌شده، یک دیتافریم جدید ایجاد می‌کنیم. سپس با استفاده از کد مربوطه، ماتریس ابهام (confusion matrix) را برای هر کدام از مدل‌ها می‌سازیم و آن‌ها را کنار یکدیگر در یک فایل ذخیره می‌کنیم:

```
# Train and evaluate each model
results = {'Model': [], 'Accuracy': [], 'Precision': [], 'Recall': [], 'F1 Score': []}

for model_name, model in models.items():...

# Display the results
results_df = pd.DataFrame(results)
```

اعمال مدل‌ها

```
# Plotting confusion matrix
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(15, 15))

for ax, (model_name, model) in zip(axes.flatten(), models.items()):...

plt.tight_layout()
plt.savefig(confusion_matrix_name)
plt.show()
```

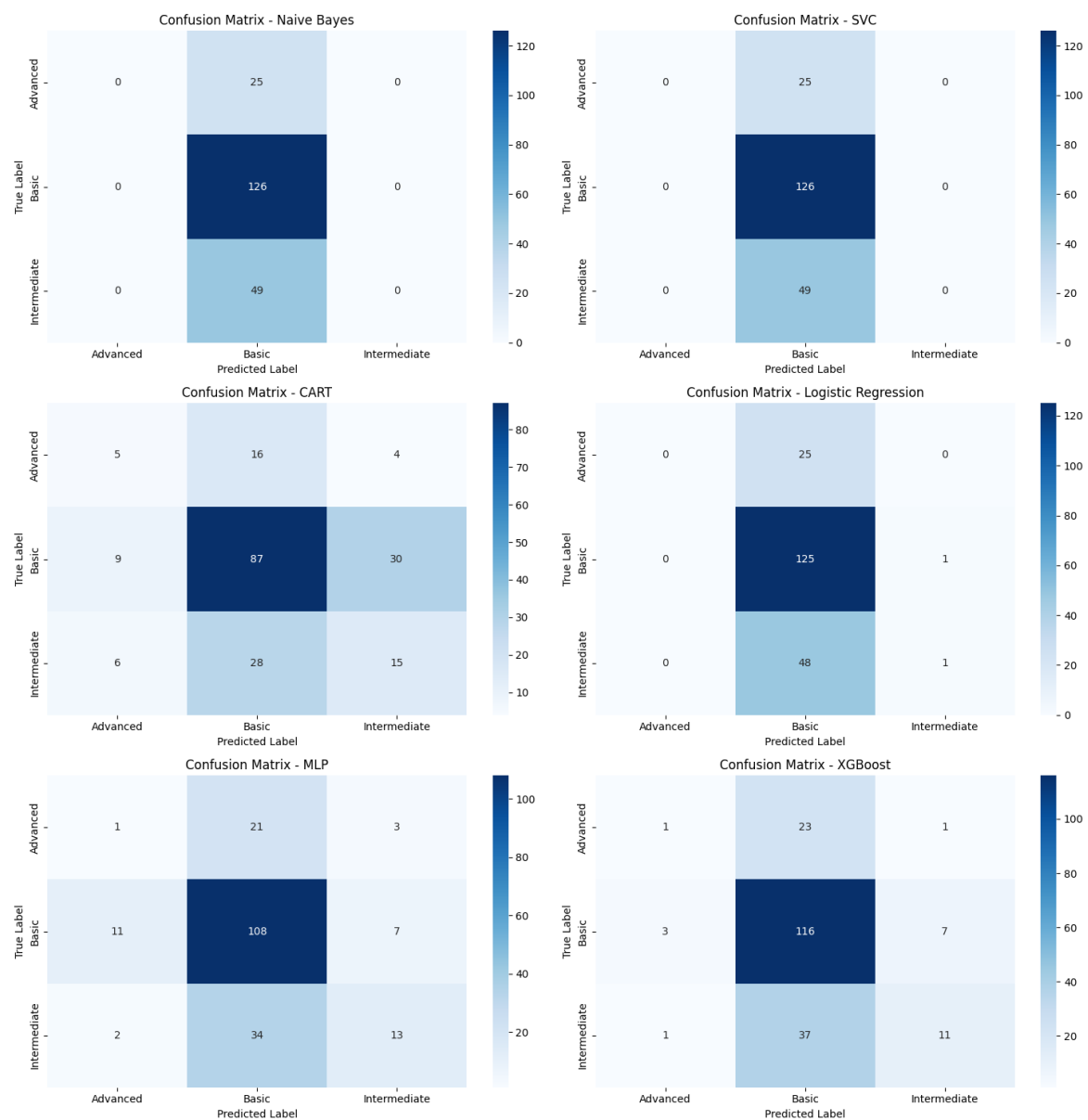
ماتریس confusion

سپس نمودارهای مقایسه برای هر کدام از ویژگی‌های گفته‌شده در بالا، ساخته و آن‌ها را ذخیره می‌کنیم.

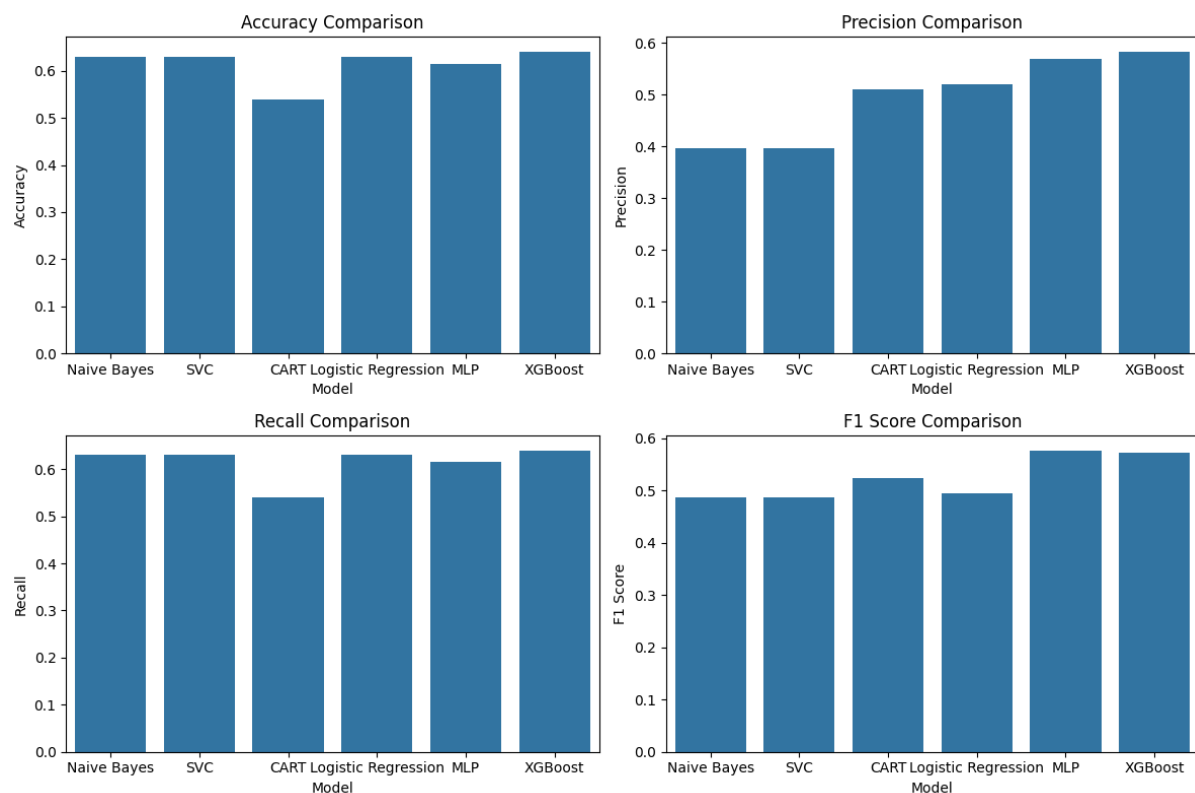
فصل پنجم: ارزیابی و بهبود نتایج

در فصل پنجم این پروژه، به مرحله‌ی ارزیابی و بهینه‌سازی مدل‌های آموزش دیده بر پایه دیتاهای آزمون می‌پردازیم. در این فاز، ابتدا مدل‌های پیش‌بینی طراحی شده را به عنوان یک مجموعه از دیتاهای آزمون مورد ارزیابی قرار می‌دهیم. سپس با توجه به نتایج به دست آمده، تلاش می‌کنیم مدل‌ها را بهینه‌سازی کرده و کارایی آن‌ها را افزایش دهیم. این فصل به عنوان یک مرحله حیاتی در گام‌های پیشروی پروژه می‌تواند به ما کمک کند تا مدل‌های پیش‌بینی خود را بهبود بخشیم و در نهایت نتایج دقیق‌تری در پروژه حاصل کنیم. از متداول‌ترین روش‌ها برای بهینه‌سازی مدل‌ها، تنظیم پارامترها، انتخاب ویژگی‌ها، و استفاده از تکنیک‌های متنوعی از جمله افزایش حجم داده، استفاده از مدل‌های پیچیده‌تر، و تغییر الگوریتم‌های آموزش است. در این فصل، ما به دنبال بهترین راهبردها برای دستیابی به یک مدل پیش‌بینی عالی و دقیق هستیم تا در نهایت به نتایج کاربردی و مفیدی دست پیدا کنیم.

داده‌های به دست آمده از فصل قبل به صورت نمودارهای زیر است:



ماتریس های confusion



مقایسه متدهای مختلف

:Accuracy

در دیتاهای آموزشی ما، هیچ یک از مدل‌ها دقت بالایی بیش از ۶۳ درصد ندارند. در مقایسه با مدل‌های دیگر، XGBoost به عنوان بهترین مدل با دقت عالی درخشانده و بدترین عملکرد به مدل CART تعلق دارد.

:Precision

در دیتاهای آموزشی ما، هیچ یک از مدل‌ها دقت بالایی بیش از ۵۷ درصد ندارند. XGBoost نیز از نظر دقت برترین مدل است و دارای بدترین عملکرد به مدل SVC تعلق دارد.

:Recall

در دیتاهای آموزشی ما، هیچ یک از مدل‌ها دقت بالایی بیش از ۶۵ درصد ندارند. XGBoost نیز از نظر recall بهترین عملکرد را ارائه می‌دهد و بدترین عملکرد متعلق به مدل CART است.

F1 Score

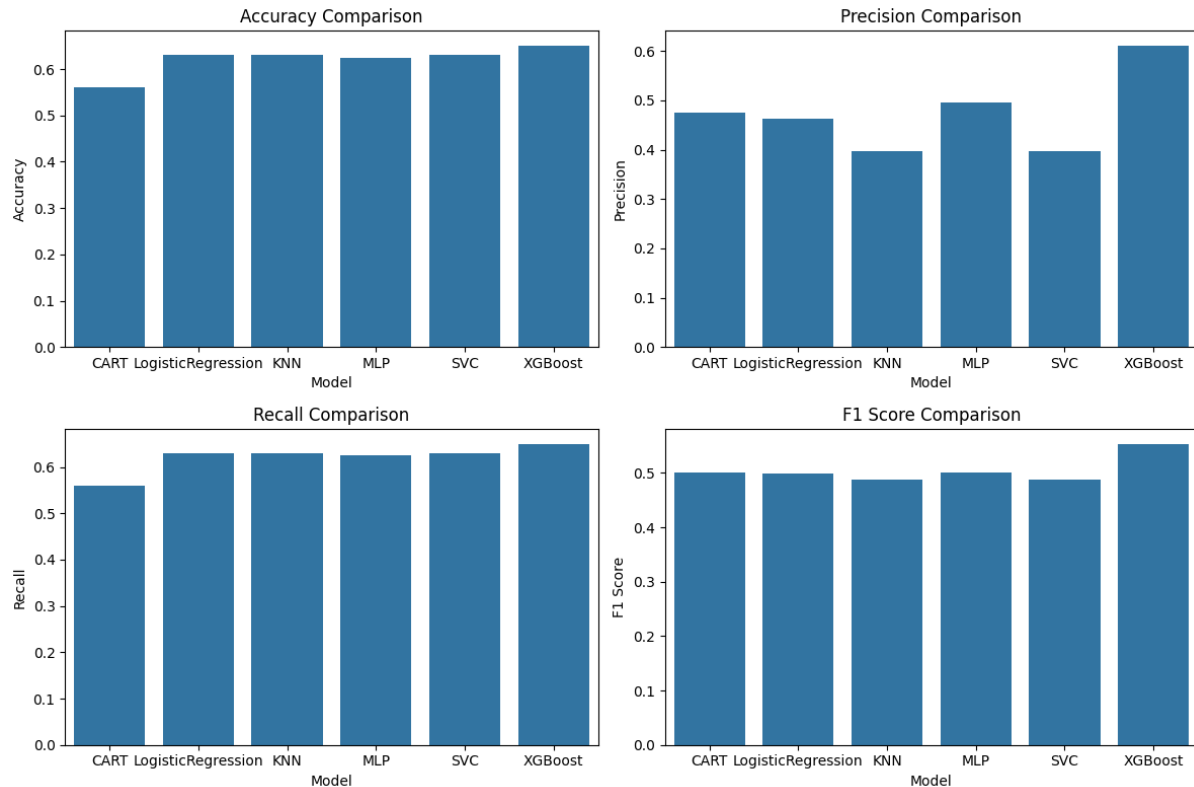
در دیتاهای آموزشی ما، هیچ یک از مدل‌ها F1 Score بالایی بیش از ۵۸ درصد ندارند. MLP به عنوان بهترین مدل از نظر F1 Score شناخته می‌شود و Naïve Bayes دارای بدترین عملکرد در این مورد است.

پیشرفت‌های به دست آمده در این پژوهش نشان می‌دهد که استفاده از مدل XGBoost به عنوان مدل اصلی تاثیر بسزایی در بهبود دقت و عملکرد کلی دارد. با اینکه مدل‌های دیگر نیز در مراحل مختلف مقایسه شدند، اما اینکه XGBoost به عنوان بهترین گزینه برجسته شود نشان‌دهنده قابلیت بالای این الگوریتم در مدیریت داده‌های پیچیده و گسترده می‌باشد.

بررسی معیارهای مختلف ارزیابی، از جمله دقت (Accuracy)، دقت مثبت (Precision)، بازخوانی (Recall) و اسکور F1 (F1 Score)، نشان می‌دهد که XGBoost به طور متوسط در هر یک از این معیارها بهترین عملکرد را از خود نشان می‌دهد. این تجزیه و تحلیل ارتقاء قابلیت پیش‌بینی و دقت در تصمیم‌گیری‌های آتی را از این مدل تایید می‌کند.

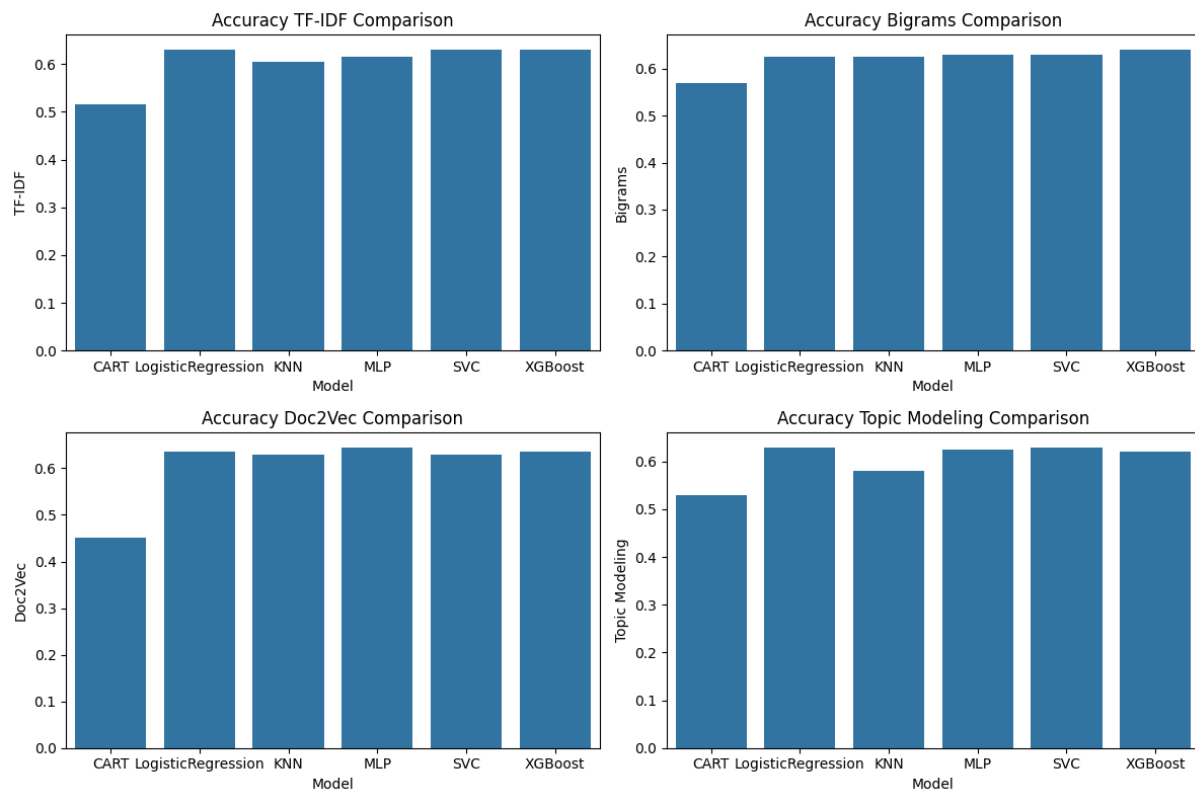
مقایسه مدل ها با اعمال Optimization ها:

- مقایسه و بهینه سازی مدل ها با اعمال بهینه سازی Hstacking Text:



در این بررسی بهینه سازی، اعمال بهینه سازی Hstacking Text نتایج قابل توجهی در تاثیر مدل ها به ویژه بر دقت (accuracy) داشته است، که در تصویر نیز به وضوح مشهود است. تمام مدل ها به جز CART با افزایش دقت مواجه شده اند، به خصوص می توان تاثیر بسزایی را بر XGBoost مشاهده کرد. در حالی که این بهینه سازی بر دقت اثر مثبت گذاشته، تاثیر آن بر دقت تخصیصی (precision) تا حدی منفی بوده است، به جز برای مدل XGBoost که به نظر می رسد این بهینه سازی در آن تاثیر مناسبی نداشته باشد. همچنین، بر روی بازخوانی (recall) تاثیر زیادی نداشته و مدل ها تقریباً به همان ترتیب قبلی خود باقی مانده اند. در مورد امتیاز F1، بهینه سازی Hstacking Text تاثیر مثبتی داشته و موجب ارتقاء امتیاز F1 برای مدل های XGBoost و MLP شده است. این نتایج نشان دهنده این است که استفاده از بهینه سازی Hstacking Text می تواند بهبود قابل توجهی در عملکرد مدل ها، به ویژه در معیارهای دقت و امتیاز F1، ایجاد کند.

- مقایسه و بهینه‌سازی مدل‌ها با اعمال بهینه‌سازی Set Hyper Parameters:



مقایسه و بهینه‌سازی مدل‌ها با اعمال بهینه‌سازی تنظیم پارامترهای ابر:

در این تجزیه و تحلیل بهینه‌سازی، اعمال بهینه‌سازی تنظیم پارامترهای ابر نتایج چشم‌گیری در دقت مدل‌ها، به‌ویژه در جنبه دقت (accuracy)، به‌همراه داشته است. در زمینه دقت، بهینه‌سازی با استفاده از Bigram به مدل‌ها افزایش قابل توجهی داده است. این به‌ویژه در تصویر مشهود است که Bigram توانسته است دقت مدل‌ها را به حداکثر برساند. از سوی دیگر، در استفاده از مدل Doc2Vec برای CART، تأثیر منفی مشهود بوده و دقت این مدل را کاهش داده است.

در تحلیل دیگر مقایسه‌ها نیز اثرات متفاوتی مشاهده می‌شود که در تصویر به خوبی قابل مشاهده هستند. این تغییرات نشان‌دهنده این است که اعمال بهینه‌سازی تنظیم پارامترهای ابر به مدل‌ها به تعادل و بهبود در عملکرد آن‌ها منجر شده و این امر به تناسب با نوع مدل و استفاده از ویژگی‌های مختلف، نتایج متنوعی را به دنبال دارد.

از نظر زمانی، در اجرای کدها، مشاهده شد که بیشترین زمان مربوط به اجرای مدل MLP در هر دو حالت بهینه‌سازی بوده است. به وضوح مشاهده شد که عملیات اجرای MLP به علت پیچیدگی بالا و نیاز به محاسبات متعدد، زمان اجرای بیشتری را اشغال می‌کند.

علاوه بر این، در ترتیب زمانی دیگر نیز مشاهده شد که پس از MLP، مدل KNN و سپس SVC زمان کمتری برای اجرا به خود اختصاص داده‌اند. این نتایج نشان‌دهنده تفاوت‌های قابل توجه در زمان اجرا بین مدل‌هاست، که از اهمیت آن در انتخاب و استفاده از مدل‌های مختلف در محاسبات زمان‌بر برنامه‌ها و پروژه‌های مختلف خبر می‌دهد.

به‌طور کلی، در نظر گرفتن نتایج زمانی می‌تواند در انتخاب مدل مناسب بر اساس نیازهای پروژه و محدودیت‌های زمانی کمک مؤثری کند.

1. <https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/>
2. <https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/>