فایل سند و راهنمای پروژه تحلیل داده‌های دیجیتون

- **معرفی**:

اینجانب سید محمد مهدی هاشمی دانشجوی سال اخر ارشد مهندسی فناوری اطلاعات گرایش تجارت الکترونیک می‌باشد.

- **تاریخ تنظیم**:

این سند در مورخ ۲۲ تیر ماه ۱۴۰۳ به درخواست شرکت دیجیتون برای پروژه تحلیل و ذخیره سازی داده‌های مربوط به گزارشات nginx می‌باشد.

- **جزئیات**:

این سند بر اساس برنامه‌های توسعه داده شده در منبع https://github.com/SMMH1999/vigitoon می باشد.

بخش اول – Read and Parse the Provided Log File

در این قسمت برنامه با فراخوانی تابع read_and_parse_logs شروع می‌شود. در این تابع ابتدا یک لیست خالی با عنوان log_entries وجود دارد.

با استفاده از دستور open یک فایل متنی در برنامه بازگشایی می‌گردد. فایل باز شده با عنوان file در حافظه در دسترس قرار دارد.

در این بخش یک حلقه for برای پیمایش خط به خط این فایل فراخوانی می‌گردد. در این حلقه هر خط با عنوان line شناخته می‌شود. هر خط از فایل متنی به تابع parse_log ارسال شده و خروجی آن در متغیر parsed_line ذخیره می‌گردد. این متغیر شامل لیستی از تمامی مقادیر تفکیک شده از بر اساس الگوی تابع parse_log است.

- تابع parse_log یک تابع تفکیک کننده است که با استفاده از قوائد عبارات منظم یا RegEx کار می‌کند. در این تابع یک الگوی جداسازی بر اساس داده‌های موجود در فایل nginx_logs ساخته شده است تا ورودی‌های دریافتی را تفکیک نماید. در این تابع اگر ورودی بر اساس الگوی موجود وجود داشته باشد برگردانده می‌شود در غیر این صورت چیزی برگردانده نخواهد شد.

در صورتی که خروجی مطلوب در متغیر parsed_line وجود داشته باشد داده تفکیک شده در انتهای لیست log_entries اضافه می‌گردد در غیر این صورت پیغام خطای مناسب نمایش داده خواهد شد.

پس از اتمام حلقه و تکمیل لیست، لیست موجود با استفاده از کتابخانه pandas به یک DataFrame جهت سهولت در استفاده تبدیل می‌گردد. همچنین با استفاده از داده های مربوط به URL یک ستون با عنوان query_params به این داده‌ها اضافه می‌گردد.

داده‌های موجود در این مرحله در فایلparsed_log_step_1.csv ذخیره شده و به عنوان خروجی برگردانده می‌گردد.

بخش دوم – Clean the Parsed Data

این بخش با فراخوانی تابع clean_data اغاز می‌گردد.

در این تابع یک ورودی با عنوان dataدریافت می‌گردد. این داده ورودی باید از نوع DataFrame باشد.

داده‌های موجود در فایل‌های گزارش‌گیری معمولا دارای معایب بسیاری می‌باشند که بایست رفع شده و مرتب گردد. این فرایند در این تابع انجام می‌گردد که شامل بخش‌های مختلی است.

- مرحله اول تبدیل داده‌های متنی به قالب‌های مناسب

در این قسمت داده‌های مربوط به ستون query_params از قالب متنی به قالب متناسب تبدیل می‌گردد. (JSON)

- مرحله دوم حذف داده‌های تکراری

در این قسمت در صورتی که داده‌های تکراری در فایل دریافتی وجود داشته باشد شناسایی شده و حذف می‌گردد.

- مرحله سوم مدیریت داده‌های ناموجود

در این قسمت که یکی از چالش برانگیزترین مراحل تمیزکاری داده است، می‌توان راه‌حل های مختلفی را در پیش گرفت.

۱- حذف ردیف داده‌های ناقص

۲- حذف ستون داده‌های ناقص

۳- پرکردن داده‌های ناقص با یک مقدار مشخص

۴- پرکردن داده‌های ناقص به صورت خودکار با استفاده از کتابخانه‌های موجود همانند Pandas

در این تابع رویکرد اول انتخاب شده است و داده‌ها با مقدار None پر می‌گردد. نحوه شناسایی و تشخیص داده‌ها بر اساس بررسی موارد استثنا صورت گرفته است. در ادامه برای اطمینان از عملکرد درست برنامه تمامی ردیف‌های داده که مقادیر None را در خود داشته باشد حذف می‌گردد.

در انتهای تابع نتایج در فایل parsed_log_step_2.csv ذخیره شده و داده‌های تمیز شده به عنوان خروجی برگردانده می‌شود.

بخش سوم – Store the Data in a MySQL Database

در این بخش نحوه ارتباط و مدیریت پایگاه داده، ساخت پایگاه داده، ایجاد جدول و افزودن مقادیر در پایگاه داده مورد بررسی قرار خواهد گرفت.

در کلیه زبان‌های برنامه نویسی از جمله Python برای ارتباط با پایگاه داده‌های رابطه‌ای از زبان SQL و کتابخانه‌های مختلفی که وظیفه برقراری ارتباط را دارند استفاده می‌گردد. در این پروژه با توجه به بیان استفاده از پایگاه داده MySQL، از کتابخانه مربوط به این پایگاه داده استفاده شده است.

MySQL.py

در فایل MySQL.py یک برنامه پایتون برای ارتباط و مدیریت پایگاه داده توسعه داده شده است. این برنامه یک Query Executor است که دستورات پایگاه داده کاربر را اجرا می‌کند.

در این فایل تابع query_executor تعریف شده است که ورودی‌های مختلفی را دریافت می‌کند.

- dbName: یک رشته که بیانگر نام پایگاه داده جهت برقراری ارتباط است. در صورتی که پایگاه داده‌ای وجود ندارد باید "None" قرار داده شود.
- sql_command: یک رشته که شامل دستورات زبان SQL برای اجرا بروی پایگاه داده است.
- *args: شامل کلیه متغیرهای ورودی دیگر، متناسب با sql_command.

در این تابع مقادیر پیشفرض برای اتصال به پایگاه داده MySQL بر اساس اطلاعات کاربر تعبیه شده است. این تابع در صورتی که پایگاه داده‌ای برای اتصال وجود نداشته باشد، ابتدا یک اتصال به پایگاه داده ایجاد می‌کند و درصورت وجود پایگاه داده اسم آن را برای اتصال دریافت می‌کند.

در اینجا یک نشانگر از کتابخانه mysql به نام ___db ساخته می‌گردد تا در ادامه به عنوان پایگاه داده شناخته شود. همچنین دستورات پایگاه داده بر اساس این نشانگر اجرا خواهند شد.

در ادامه بحث مدیریت خطا و هماهنگی برای دریافت و مدیریت اطلاعات در این تابع پیاده‌سازی شده است.

Digitoon.py

در برنامه اصلی دو تابع setup_database و save_data_to_db برای مدیریت، ارتباط و دستکاری داده‌ها در پایگاه داده ایجاد شده است.

در تابع setup_database ابتدا دو دستور SQL برای اجرا به تابع query_executor ارسال می‌گردد.

- دستور اول:

```
CREATE DATABASE IF NOT EXISTS log_analysis
```

برای ساخت یک پایگاه داده با نام log_analysis در mysql. این دستور در صورتی که این پایگاه داده قبلا ساخته شده باشد اجرا نخواهد شد.

- دستور دوم:

```
CREATE TABLE IF NOT EXISTS logs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    ip VARCHAR(255),
    timestamp DATETIME,
    method VARCHAR(255),
    url TEXT,
    status INT,
    size INT,
    query_params JSON
)
```

که یک جدول با نام logs را در پایگاه داده log_analysis خواهد ساخت. این دستور در صورتی که جدولی با این نام در پایگاه داده وجود نداشته باشد ساخته خواهد شد.

در تابع save_data_to_db یک DataFrame به عنوان ورودی دریافت می‌گردد. که شامل داده‌های لازم برای ذخیره سازی بروی پایگاه داده می‌باشد.

در این تابع یک حلقه برای پیمایش سطرهای DataFrame ایجاد شده است. در این حلقه هر سطر با عنوان یک Series دریافت شده و مقادیر آن در متغیرهای مورد نیاز ذخیره می‌گردد. پس از تبدیل هر سطر به متغیرهای مطلوب، یک دستور SQL برای درج اطلاعات در جدول پایگاه داده به تابع query_executor ارسال می‌شود که شامل اطلاعات مورد نیاز برای افزون به پایگاه داده می‌باشد.

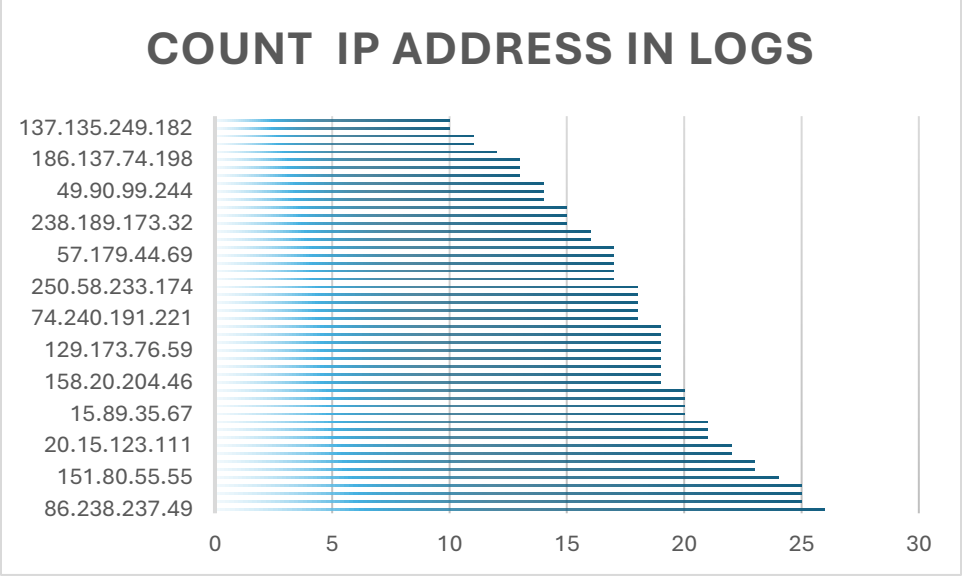
بخش چهارم – Generate Visualizations

در این قسمت مبحث مصورسازی و ایجاد مباحث مربوط به Bi مطرح می‌گردد.

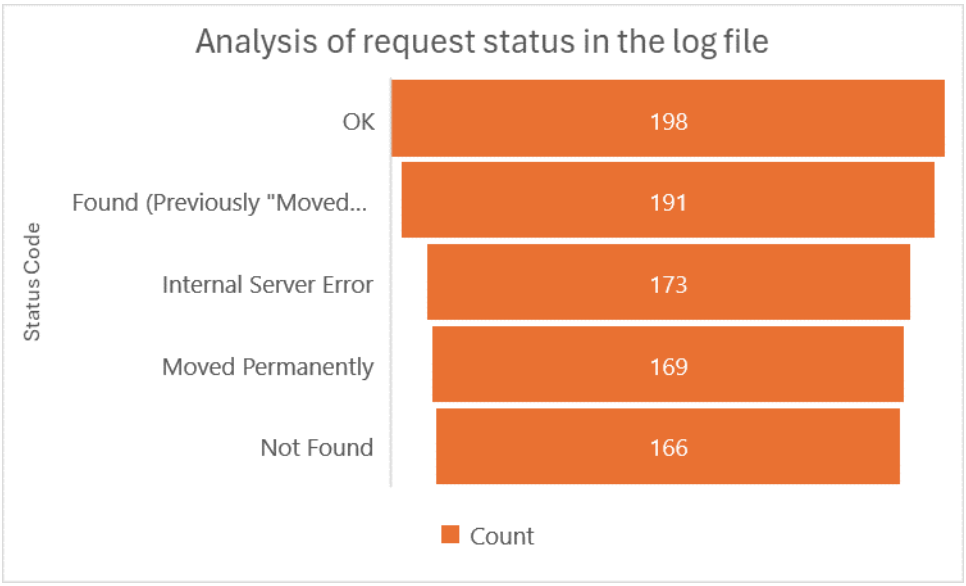
در این برنامه تلاش شده است برخی از توابع و مدلهای مصورسازی بیان شود ولیکن با توجه به تخصص بنده در زمینه Bi یا هوش تجاری پیشنهاد می‌شود از نرم‌افزارهای حرفه‌ای تر نظیر PowerBi, Tableau و Excel استفاده شود. بر این اساس برخی از دستور برای نشان دادن تسلط بر مباحث Visualization بیان شده است.

برخی از گزارشهایی که می‌توان از داده‌های موجود دریافت نمود.

- تعداد یکتا و تعداد درخواست‌های موجود بر اساس IP Address



- تجزیه و تحلیل وضعیت درخواست‌های سرور



ارائه گزارشات بر اساس داده‌های موجود می‌تواند در دسته بندی‌های مختلفی قرار گیرد که شامل:

- دسته بندی بر پایه تعداد درخواست‌ها
- دسته بندی بر پایه وضعیت درخواست‌ها
- دسته بندی بر پایه زمان درخواست
- دسته بندی بر پایه درخواست و درخواست کننده
- دسته بندی بر پایه مدلهای درخواست
- دسته بندی بر پایه شناسایی درخواست‌های مخرب
- دسته بندی بر پایه حجم مبادلات کاربر
- دسته بندی بر پایه فعالیت‌های صورت گرفته
- و دیگر مواردی که متناسب با معیارهای کارفرما قابل تنظیم است.