نام و نام‌خانوادگی: پریسا عمادی

شماره دانشجویی: ۴۰۱۳۶۲۴۰۲۰

لینک گیت‌هاب: https://github.com/Dandelion07/BigData_HW8.git

:3

این کد برای کلاسترینگ داده‌های Uber در ۸ کلاستر با استفاده از الگوریتم KMeans استفاده می‌شود.

از پکیج pyspark.sql ، SparkSession و از پکیج pyspark.ml.clustering ، KMeansModelو KMeans و از پکیج pyspark.sql.functions ، count و col و از پکیج pyspark.ml.feature، VectorAssemblerو os ایمپورت می‌شوند.

مسیر HADOOP_HOME برای سیستم عامل مشخص شده و JAVA_HOME برای محیط جاوا مشخص شده است.

یک ابزار جلسه اسپارک (Spark Session) با نام UberKMeans ایجاد می‌شود.

داده‌های سفر Uber از فایل CSV به پایگاه داده PySpark خوانده می‌شود.

ستون‌های عرض جغرافیایی و طول جغرافیایی به نوع عددی تبدیل می‌شوند.

VectorAssembler ایجاد می‌شود تا ستون‌های عرض و طول جغرافیایی را به یک بردار ویژگی ترکیب کند.

تعداد کلاسترها به ۸ تنظیم می‌شود.

داده‌ها به دو مجموعه آموزش و تست با نسبت ۸۰ به ۲۰ تقسیم می‌شوند.

مدل KMeans برای داده‌های آموزش ساخته می‌شود.

داده‌های جدید برای کلاسترینگ خوانده می‌شوند.

برای داده‌های جدید، VectorAssembler ایجاد می‌شود تا ستون‌های عرض و طول جغرافیایی را به یک بردار ویژگی ترکیب کند.

با استفاده از مدل KMeans آموزش دیده شده، داده‌های جدید به کلاسترها تخصیص داده می‌شوند.

داده‌های کلاستر شده در یک پایگاه داده Cassandra ذخیره می‌شوند.

:3_1

ایمپورت کردن کلاس KMeans و SparkSession از پکیج‌های pyspark.ml و pyspark.sql به ترتیب

ایمپورت کردن کلاس ClusteringEvaluator از پکیج pyspark.ml.evaluation

ایمپورت کردن کلاس VectorAssembler از پکیج pyspark.ml.feature

ایمپورت کردن تابع col از پکیج pyspark.sql.functions

تعریف متغیر JAVA_HOME برای تنظیم مسیر جاوا

ساختن یک نمونه از SparkSession و نام‌گذاری برنامه

لود داده‌های سفر Uber به دیتافریم PySpark

تبدیل ستون‌های طول و عرض جغرافیایی به اعداد اعشاری

ایجاد یک VectorAssembler برای ترکیب دو ستون طول و عرض جغرافیایی به یک بردار ویژگی

تنظیم تعداد خوشه‌ها برای الگوریتم KMeans و جداسازی داده‌ها به دو بخش آموزش و تست

آموزش مدل KMeans با استفاده از داده‌های آموزش

لود داده‌های جدید برای خوشه‌بندی

ایجاد یک VectorAssembler برای ترکیب دو ستون طول و عرض جغرافیایی به یک بردار ویژگی برای داده‌های جدید

فقط ستون‌های "Lat" و "Lon" برای خوشه‌بندی به عنوان ورودی برای VectorAssembler انتخاب شده است. سپس، فقط ستون features برای هر داده‌ی جدید با استفاده از select انتخاب می‌شود.

در خط بعدی، clustered_data از مدل KMeans آموزش‌دیده با استفاده از new_data به دست می‌آید. این داده‌ها در بردار خوشه‌بندی شده است که شامل دو ستون "features" و "prediction" است.

سپس، با استفاده از groupBy وcount ، تعداد داده‌های هر خوشه شمارش شده و در cluster_counts قرار داده می‌شود. در نهایت با فراخوانیshow ، تعداد داده‌های هر خوشه در DataFrame نمایش داده می‌شود.

:3_2

این بخش کد، یک الگوریتم شناسایی مناطق پرتردد در شهر تحت تحلیل را پیاده‌سازی می‌کند.

در خط اول داده های اولیه به دو دسته داده های آموزشی و داده های تست تقسیم می‌شوند. ۸۰ درصد از داده ها به عنوان داده های آموزشی و ۲۰ درصد به عنوان داده های تست تعیین می‌شوند.

seed 42 به منظور تولید اعداد تصادفی استفاده شده است.

الگوریتم KMeans با k شامل ۸ شاخص را مقداردهی می‌کند. همچنین با استفاده ازsetSeed ، عدد تصادفی ثابت را تولید می‌کند.

داده جدید جهت تشخیص مناطق جدید بارگذاری می‌شود.

فیچر های داده های جدید تشکیل می‌شوند. این فیچرها شامل دو متغیر داده‌ای شامل طول جغرافیایی (Longitude) و عرض جغرافیایی (Latitude) هستند.

با استفاده از مدل آموزش دیده شده در قسمت قبل، داده های جدید خوشه‌بندی می‌شوند.

کوئری برای پیدا کردن خوشه بندی بیشترین فعالیت (peak cluster) ارائه شده است. برای این منظور، داده ها به ترتیب تعداد مرتب شده و سپس ماکزیمم مقدار تعداد در هر خوشه بررسی می‌شود. نام خوشه بیشترین فعالیت به عنوان peak cluster انتخاب می‌شود.

متغیر خوشه بندی را به داده های اولیه اضافه می‌کنیم.

سپس با استفاده از داده‌های فیلتر شده، اطلاعات هفته ها را با استفاده از ستون 'Date/Time' استخراج می‌کنیم. این کار با استفاده از تابع dt.week انجام می‌شود.

داده‌ها را بر اساس ستون‌های 'week' و 'Base' گروه‌بندی می‌کنیم.

تعداد سرویس‌های موجود در هر گروه (با ترکیب هر منطقه سرویس‌دهی و هر هفته) با استفاده از تابع size() محاسبه شده و به نام 'counts' ذخیره می‌شود.

:3_3

این قطعه کد به منظور پیدا کردن خوشه‌ای است که بیشترین فعالیت در طول هفته را داشته است.

ابتدا داده‌های جدید با استفاده از مدل KMeans خوشه‌بندی شده و نتایج حاصل از خوشه‌بندی در clustered_data ذخیره می‌شود.

سپس داده‌های خوشه‌بندی شده با توجه به شماره خوشه و تاریخ خوشه‌بندی شدن، گروه‌بندی می‌شوند و تعداد نمونه‌های هر گروه با تابع count شمرده می‌شوند و در ستون جدیدی با نام count در grouped_data ذخیره می‌شوند.

در ادامه داده‌ها بر اساس شماره خوشه گروه‌بندی می‌شوند و سپس تعداد نمونه‌های هر خوشه شمرده می‌شود و در ستون جدید total_count در cluster_totals ذخیره می‌شود.

سپس داده‌های مرتب شده cluster_totals بر اساس تعداد کل نمونه‌های هر خوشه به صورت نزولی مرتب می‌شوند.

خوشه با بیشترین تعداد نمونه‌های درون خود بر اساس اطلاعات مرتب‌شده از متغیر sorted_clusters انتخاب می‌شود و در متغیر peak_cluster ذخیره می‌شود.

در اخر شماره خوشه با بیشترین فعالیت کاری در طول هفته چاپ می‌شود.

:3_4

این قطعه کد شروع به پردازش داده‌های ۱۰ روز گذشته می‌کند و به دنبال داده‌هایی می‌گردد که مرکزگرایی بیشتری دارند.

در خط دوم، داده از فایل اکسل خوانده شده و به فیلتر می‌شود تا تاریخ آن با توجه به تفاوت با تاریخ امروز حداکثر ۱۰ روز باشد.

سپس، خطوط داده‌ای که بیشترین مرکزگرایی را دارند به دست می‌آیند. سپس داده‌هایی که در این مرکزگرا هستند، فیلتر می‌شوند. به طور مشابه، تعداد خطوطی که هر یک از آن‌ها مربوط به یک مختصات خاص از محدوده جغرافیایی هستند، شمارش می‌شوند. سپس به ترتیب نزولی، ده نقطه با بیشترین شماره شمارش انتخاب شده و در یک جدول جدید بازنشانی می‌شوند.