מגישים: אריאל נובומינסקי, דוד פרנס, אילי ששון.
ת.ז. : 322228974, 322995358, 337977045

ETL:

Metadata:

לקחנו 5 מדינות במערב אירופה: גמרניה, צרפת, אנגליה, נורבגיה וספרד. מדינות הללו ביחד מהווים כ-110 מיליון שורות בקפקא.

Design:

מנתונים אלו יצרנו שלוש טבלאות אשר השתמשנו בהם לצרכים שונות:

1. time_df - StationId, Date, label
2. spatial_df - StationId, latitude, longitude, avg_prcp, max_temp, min_temp
3. model_df - StationId, label, avg_prcp, cluster_label, month, year

הטבלה time_df וspatial_df שימשו אותנו בניתוח ההשפעה של זמן ומיקום/אקלים על משקעים כהכנה למודל. הטבלה model_df משמש אותנו בהרצת מודל למידה על הנתונים והוא משלב בין time_df וspatial_df כדי לעשות זאת. בנוסף, החלוקה לtime_df וspatial_df נותן את האפשרות ליצור עוד design לעוד מודלים כמו מה שעשינו בבונוס. ככה שכל טבלה שימושית עבור המשימות שונות ורלוונטיות.

Description:

StationId - Name of the station (String)
Date - (DateType)
latitude & longitude - geographic location of each station
avg_prcp - average precipitation that per day for each station
max_temp - average of maximum TMAX per batch for each station
min_temp - average of minimum TMIN per batch for each station
cluster_label - results of clustering
month - month from date
year - year from date

Explanation & considerations:

בחירת פיצ'רים:

כפי שנראה בניתוח נתונים, משקעים הוא משתנה מאוד דינמי אשר אינו תופעה מחזורית במדינות שלקחנו. לכן, חתרנו לזהות משתנים אשר מצביעים על התנהגות של אזור. מצאנו אחרי השוואה עם מפות של האקלים באירופה שהמשתנים האגרגטיביים avg_prcp, max_temp, min_temp מגלים את האזורים כפי שמוגדר ע"י אפיון הKoppen-Geiger.

נציין שלא לקחנו ממוצע רגיל על טמפרטורה מקסימלית ומינימלית. זאת מכיוון שהטמפרטורות בחורף וקיץ יבטלו אחת את השנייה. אלא לקחנו מקסימום ומינימום בהתאמה על כל batch ועל זה ממוצע. ככה, המשתנים מביאים לידי ביטוי את טווח הטמפרטורות בכל תחנה. יחד עם זאת, ממוצע מאזן את ההשפעה של ימים יוצאים מן הכלל. (למשל, יכול להיות שבמקום מאוד קר היה פעם יום אחד חם, אבל זה רק מאפיין את המקום אם טמפרטורה כזאת מופיע לאורך השנים. ורק אם כך, ישפיע על הערך max_temp, כמו שצריך.)

מגישים: אריאל נובומינסקי, דוד פרנס, אילי ששון.
ת.ז. : 322228974, 322995358, 337977045

השתמשנו בתכונות המיקומיות האלה של avg_prcp, min_temp, max_temp כדי להבין את התנהגות האזור שהתחנה נמצאת בו. התוצאה היא cluster_label.

טעינת הנתונים:
הגדרנו datastream אשר מבצע סינון וטרנספורמציות על הנתונים שהיא קולטת. סיננו החוצה נתונים אשר לא עמדו ברמת איכות כלשהי, כלומר היה קיים להם ערך בq_flag. בניתוח ראשוני של הנתונים התגלו הרבה יוצאים מן הכלל לא הגיוניים בקבוצה זאת. לכן, הורדת רשומות עם q_flag משפר את איכות הנתונים.
בנוסף, לקחנו רק את חמשת השנים האחרונות בשביל בניית המודל שלנו. עקב השינויים באקלים אשר משנים דפוסי מזג אוויר בשנים האחרונות, נתונים ישנים אינם רלוונטיים למודל שצריך לנבא מזג אוויר עכשווי.
בנוסף, חילצנו את הנתונים מהפורמט של קפקא, התאמנו את טיפוס הנתונים לצרכים שלנו והורדנו משתנים שאינם רלוונטיים למשימה שלנו כמו s_flag.
עיבדנו את הנתונים בבאצ'ים של 500,000 נתונים. בכל באצ' נעשה שתי פעולות שונות. הראשונה, שומרים את כמות המשקעים שירד בכל תחנה בכל יום לשרת. ככה יוצרנו timeDF. השנייה, שומרים נתונים אגרגטיביים אשר מתעדכנים בכל באצ', בפרט avg_prcp, max_temp, min_temp. עושים זאת בעזרת קובץ parquet אשר מעדכנים בכל באצ'.
השיקול של גודל הבאצ' היה בין באצ' גדול אשר משמעותו לבצע פחות פעמים פעולות יקרות כמו join לבין באצ' קטן אשר כל אחד יותר מהיר ופעולות הjoin לוקחות פחות זמן כי הנתונים פזורים בפחות partitions.
השתמשנו בcaching כדי להאיץ את ביצועי כל באצ'.
לאחר הזרמת הנתונים, מבצעים את הדברים הבאים:
הראשון הוא קלאסטרינג. כיוון שמשתמשים בפרמטרים אגרגטיביים, avg_prcp, max_temp, min_temp, אי אפשר לעשות זאת תוך כדי הסטרימינג. במציאות הקלאסטרים האלה היו מוכרים על בסיס ממוצעים קודמים ולא היה צורך לחשב אותם בכל הזרמה. אלא רק אחרי פרק זמן מוגדר, כמו פעם בשנה. כיוון שהפרויקט רק מדמה מצב של סטרימינג, בנינו את הנתונים שהיה צריך בשביל הקלאסטרינג תוך כדי הזרמה וביצענו את הקלאסטרינג רק אחר כך. ככה יצרנו spatialDf.
בנוסף, מבצעים join אשר מוסיף לtimeDF שקיבלנו תוך כדי הstreaming גם נתונים אגרגטיביים כמו avg_prcp וcluster_label אשר רלוונטיים למודל. ככה עשינו join פעם אחת עם התוצאה הסופית, במקום בכל באצ' על ערך משתנה.
במציאות, נתונים של מזג אוויר היו מגיעים בבאצ' כל יום עם הנתונים מהיום האחרון. אם כן, הייתי מעביר את הjoin לבתוך הבאצ' כך שהוא יתעדכן כל יום. בדימוי של הזרמה שביצענו בשביל הפרויקט העלות של לעשות כך גובה מחיר לא סביר ביחס לפתרון הפשוט שעשינו. בנוסף, התמודדנו עם נתונים אגרגטיביים תוך הזרמה והראינו שאנו יודעים איך להתמודד איתו. לכן, לא ראינו צורך לחבל בביצועים שלנו כדי להראות זאת.
אחרון, אסביר למה דווקא בקלאסטרינג עברתי לpandas והשתמשתי בsklearn. עשיתי זאת רק כאשר יש שורה אחת לכל תחנה, כלמור גודל הטבלה חסומה וידועה מראש. לכן, אין סיכוי לfault עקב הגודל. בנוסף, זה סדר גודל יותר מהר. בנוסף, כפי שציינתי קלאסטרינג אינו חלק מהזרמה אלא הוא נתון שמתעדכן לפי פרק זמן קצוב מראש על בסיס נתונים סטטיים.