**דגשים**

על מנת להריץ את הפרויקט יש לכתוב בטרמינל:

streamlit run prediction.py

להרצת הפרויקט נדרש התקנה של הבאים דרך הטרמינל:

* pip install requests
* pip install streamlit
* pip3 install streamlit-lottie
* pip install streamlit-option-menu
* pip install sqlite3
* pip3 install fbprophet
* pip install pandas
* pip install pymongo
* pip install pymongo[srv]

**הסבר על החיבור ל-MongoDB**

התוכנה מתחברת למסד הנתונים MongoDB ולוקחת משם דגימה לבדיקת הנתונים הרלוונטים וליישום הפעולות המובנות על סמך התוצאות מהבדיקה.

* במידה ולא קיים המטבע שנבחר מהרשימה במסד, לשלוח בקשה לyahoo finance ומשם מעבירים את המידע כגיבוי לMongoDB.
* במידה וקיים המטבע אך התאריך הוא לא זהה להיום, לשלוח בקשה לyahoo finance ולמלא במסד הנתונים MongoDB את החסר מהיום האחרון במסד עד לרגע זה.
* במידה וקיים המטבע וגם התאריך מעודכן להיום, לקחת את הנתונים רק ממסד MongoDB ולהמשיך את התוכנית.

**פירוט פונקציות:**

def get\_MognoDatabase()

* מטרת הפונקציה היא לחבר אותה למסד הנתונים שלנו
* שומרת String שיש בו את המפתח להתחברות למסד הנתונים הרלוונטי של MongoDB
* מתחברת אליו דרך הפקודה MongoClient
* מחזירה את החיבור למסד בתור client עם collection ספציפי שהוא siteData

def checkIfCollectionExists(currency: String)

* מטרת הפונקציה היא לבדוק האם יש לנו מידע מאוחסן כלשהו על המטבע
* שמים את החיבור למסד שלנו בתוך משתנה mongoDB
* לוקחים מתוך המסד את כל הcollections לשם השוואה בין הString שקיבלנו לתוך הפונקציה לבין האם הcollection (כלומר המטבע) קיים במסד
* אם קיים, מחזירים .True אחרת False

def getLastDateFromDB(currency: String)

* מטרת הפונקציה היא להחזיר את המסמך האחרון שאספנו עם היום שהכי קרוב לעכשיו
* שמים את החיבור למסד שלנו בתוך משתנה mongoDB
* מחפשים בתוך הcollection שנקרא לפי הcurrency את התאריך האחרון שהכי קרוב להיום, בעזרת sort() עם 1- על המאפיין Date ונקביל אותו ל1 בעזרת limit(), לבסוף עוטפים את כל זה ב list כי אנו רוצים את זה בתוך מערך של רשימה כדי שתיהיה לנו אפשרות לגשת אחרכך, אחרת נקבל cursor אשר אין אפשרות לכניסתו ללא צורך בלולאה אשר מצביעה עליו
* ניגשים לlist שיצרנו בעזרת [0] (כיוון שזה תמיד יחזיר רק קובץ אחד בעזרת הlimit()) ונחזיר בreturn רק את המאפיין Date אשר נראה כך 01-01-99

def checkIfNeedUpdateDB(currency: String)

* מטרת הפונקציה היא לבדוק האם יש צורך בלעדכן מידע במסד הנתונים שלנו
* יוצרים את התאריך של היום בעזרת ספריית הdatetime לתוך המשתנה currentDay
* בעזרת הפונקציה getLastDateFromDB אנו מצליחים לייבא את התאריך האחרון (והכי קרוב לעכשיו) שנמצא בcollection שנבחר ושמורים אותו בתוך משתנה docDate
* בונים כעת String חדש אשר יכיל את התאריך מהמשתנה docDate ונוסיף לו שעה כדי שיוכל לעבור בcasting בין String לDate לשם השוואה אחר כך (כי אין אפשרות להשוות בין String ל Date, אלא רק Date ל Date)
* נוסיף פורמט אשר בעזרתו נגיד לString החדש שיצרנו איך להיראות נכון
* נדרוס את המשתנה docDate בעזרת ספריית datetime אשר נספק לה את המשתנהdocDate לפני דריסתו וגם את הפורמט שעיצבנו לו, והספרייה תמיר אותו לסוג Date
* כעת נשאר לבדוק האם המשתנה הדרוס docDate (שעכשיו הפך לסוג Date) קטן מהתאריך של היום (currentDay) ואם כן, סימן שיש פער בין מה שיש במסד ולבין היום ולכן נחזיר True אחרת False

Def createMongoDataTable(dataAcquired: Object, fromDatabase: Boolean)

* מטרת הפונקציה היא לבנות טבלה משלנו כדי שנוכל למיין אותה כרצוננו ולעלותה למסד שלנו
* הפונקציה מקבלת Boolean בשם fromDatabase אשר תפקידו הוא לספק את המידע האם הפונקציה הזו נקראה עם הנתונים מתוך מהסד או הנתונים מyahoo finance (שני הנתונים האלו ניגשים אליהם בצורות שונות ולכן יש צורך בהפרדה זו)
* באם המידע שקיבלנו מfromDatabase הוא True
  + בעזרת הפונקציה של ספריית pandas אנו יוצרים לעזמנו טבלת DataFrame עם אותו גודל של עמודות כמו הנתונים שקיבלנו במשתנה dataAcquired ובוחרים שם לכל עמודה ושמים את כל זה לתוך משתנה שנקרא data
  + כעת ממלאים את הטבלה הריקה(עם אותם שורות ועמודות כמו בdataAcquired) שקיבלנו מהפעולה הקודמת בעזרת לולאה לפי כל עמודה, כלומר עמודת Date של טבלת data מתמלאת בdataAcquired במיקום של ה i מהלולאה במאפיין date
  + לאחר סיום המליאה, אנו מעדכנים את הtype של כל נתון (כיוון שכולם הם string): מעבירים את הdate מסוג String לסוג date בעזרת הפונקציה של pandas יחד עם פורמט שהבאנו לו ואת שאר הנתונים אנו מעבירים אותם מסוג String ל Float בעזרת הפונקציה astype
  + מחזירים בעזרת return את הטבלה החדשה
* באם המידע שקיבלנו מfromDatabase הוא False
  + כל ליעיל קורה אך הפעם מילוי הטבלה קורה טיפה שונה, כיוון שמ-yahoo finance מקבלים את הdate בתור index אז למילוי הdate אנו עושים לפני מיקום ה-i של הindex ואת שאר הנתונים לפני המשתנה ואז את העמודה ורק אז לפי מיקום השורה של ה-i.

def turnDateToStr(String: String)

* מטרת הפונקציה היא להפוך את הdate שיכלול בתוכו רק מספרים
* המשתנה resolvedString הוא String בלי כלום והוא משתמש לאן נוכל לשמור את הString החדש שלנו
* בתוך לולאה אנו עוברים על כל הString שהתקבל מהפוקנציה כשקראו לה בודקים בעזרת isDigit() האם זה מספר או לא, אם כן אז להכניסו לresolvedString
* להחזיר את String שכולל בתוכו רק מספרים עכשיו (כלומר את התאריך)

def updateMongoDB(dataFetched: Object, currency:String)

* מטרת הפונקציה היא לעדכן את מסד הנתונים שלנו במידע החסר
* שמים את החיבור למסד שלנו בתוך משתנה mongoDB
* בעזרת הפונקציה createMongoDataTable אנו מעבירים את הנתונים דרך dataFetched יחד עם false כיוון שביצוע עדכון הנתונים יכול להיות רק מחוץ למסד הנתונים שלנו, פונקציה זו תחזיר לנו טבלה חדשה שאפשר לפעול איתה עם המסד בתוף המשתנה data
* לצורך ויזואלי בלבד יש את numC אשר יספור לנו כמה נתונים אנו מכניסים לתוך המסד
* כעת יש לולאה אשר תרוץ על כל המידע שהתקבל לנו בdata ולכל ריצה של שורה אנו יוצרים Object בשם oneRow אשר יכלול את כל המאפיינים הדרושים למודל המסד שלנו
* אך לפני כן אנו מכינים את המידע לכך בעזרת הפיכת הdate להיות כuniqueKey למסמך הנוכחי במקום הuniqueKey הrandomly של mongoDB וגם מוסיפים לשורת התאריך מקף – כדי שיהיה נוח ויזואלית כאשר מסתכלים במסד
* לאשר יצירת הObject בשם oneRow , יוצרים משתנה בשם exist ומגדירים אותו כFalse, זה ישמש אותנו לתהליך הבא שבו מחפשים אם ובמקרה קיים האתריך הזה (לשם מניעת שכפולויות במקרה וקורה מצב שכזה)
* הבדיקה נעשיית על ידי גישה לcollection עם השם שמקבלים מcurrency ומנסים למצוא id (כלומר הuniqueKey שלנו) אשר תואם לזה שבoneRow ואז הcursor שמוחזר לנו אנו עוטפים בlist כדי שנוכל לעבור עם לולאה ותבדוק בפנים האם זה קיים (כי cursor יכול עדיין להיות מוחזר למרות שיש בו 0 מסמכים)
* אם קיים מסמך בתוך הcursor, סימן שיש כבר את האתריך הזה במסך ואין צורך להוספתו, ולכן הופכים את הexist ל True
* מייד לאחר סיום הבדיקה הזו, המשתנה exist קובע האם יש להוסיף את הoneRow למסד, אם exist הוא False אז מוסיפים למסד דרך הפוקציה של ספריית pymongo שנקראת insert\_one, אחרת לא מוסיפים למסד
* שני הפעולות האלו מוסיפות לnumC ב 1+ (משומש רק לתצוגה ויזואלית)

def getMongoDBData(currency: String)

* מטרת הפונקציה להחזיר את כל המסמכים שנמצאים בתוך collection שהשם שלו סופק בcurrency
* שמים את החיבור למסד שלנו בתוך משתנה mongoDB
* ומחזירים טבלה חדשה דרך הפונקציה createMongoDataTable שהנתונים נשאבים מהמסד הנתונים עצמו עם ה collection בשם של currency ומסומן כTrue כיוון שהנתונים בפונקציה הזו תמיד יגיעו מהמסד הנתונים שלנו