Cryptocurrency Forecasting

פרויקט גמר תכנון ותכנות מערכות

מגיש: ליאם ברינקר
תעודת זהות: 213215205
בית הספר: מקיף י"א ראשונים ראשון לציון
כיתה: י"ב 4
מורה: דינה קראוס
תאריך הגשה: 20.6.2021

תוכן עניינים

מבוא

בשנים האחרונות, טכנולוגיית למידת מכונה (Machine Learning) הפכה לתחום בלתי נפרד מחיינו. ניתן לראות זאת ביישומים רבים של הטכנולוגיה בתחומים שונים, החל במיון אימיילים וכלה במכוניות אוטונומיות. המשותף לכל אלו הוא שהם מדמים באופן מלאכותי את הליך הלמידה האנושית. לעתים, התהליך יהיה פשוט ובמקרים אחרים ייתכן שלא וכדי שהמכונה "תבין" כיצד לפתור את הבעיה העומדת לפניה, יש לפרקה לתתי משימות למשל מכונית אוטונומית צריכה לדעת לזהות תמרורים ועצמים שונים וגם לדעת לווסת את מהירותה בהתאם לתנאי הדרך. יש לציין שגם המידע שמקבלת המכונה משתנה בהתאם למשימה הנדרשת.

בפרויקט שלי בחרתי לעסוק בחיזוי הערך העתידי של המטבע המבוזר הנפוץ ביותר – ביטקוין (Bitcoin). הביטקוין, המטבע המבוזר הראשון, הוא למעשה אמצעי תשלום דיגיטלי לחלוטין, שלא כמו כרטיסי אשראי, מאפשר למשתמשים בו להשתמש בכספם באנונימיות, ללא כל פיקוח וניטור של הכספים מצד גורמים שונים. התומכים במטבע המבוזר ינמקו את תמיכתם בעצם השמירה על הפרטיות שמספק המטבע שהרי כל נתון על עסקה או העברת כספים שמתבצעת אינו נאגר, נשמר ומנותח וכך לא יכול להיות מנוצל על ידי אחרים. לא פעם ארע שמידע אישי ורגיש של משתמשים נוצל באופן מקומם, כפי שאירע לדוגמה בשערוריית הפרטיות של קיימברידג' אנליטיקה. הרעיון החדשני של המטבעות המבוזרים היה אהוד בקרב אנשים רבים והביטקוין ספציפית הפך לכל כך פופולרי כך שעד סוף שנת 2017 שווי השוק של מטבעות אלו (כלל המטבעות הקיימים) השתווה להונם של כמה מהבנקים הגדולים בעולם.
השיטה שאפשרה את קיומם של המטבעות המבוזרים נקרא – בלוקצ'יין. זהו למעשה ספר ניהול עסקאות, כאשר עותקים של אותו ספר מופצים בין מחשבים ברחבי העולם. ניתן להגיד שהבלוקצ'יין מהווה מערכת כספים בשליטת ההמון, ללא חוקים וללא תקנות. רבים רואים בשיטה ובמטבע המבוזר המצאות ללא דופי, משמעותיות כמו האינטרנט ואף יותר.

מאז 2009 ערכו של הביטקוין השתנה בתדירות רבה, לעיתים צנח ולעיתים נסק כך שהוא גרם לאנשים להתייחס אליו כמו למניות בשוק המניות. עובדה זו ואלו שצוינו לעיל עוררו את סקרנותי ושל עוד רבים אחרים בניסיון לחקור את הסדרה העתית המורכבת מערכו של המטבע בכל פרק זמן קבוע, ולנסות לחזות את ערכו בעתיד. קיימים פתרונות שונים כמו סקר שוק ואחרים שאמורים לתת אינדיקציה טובה לירידה או עלייה כללית ואף מודלים שונים המתבססים על למידת מכונה שמנסים לחזות את הערך עצמו. עם זאת משום שאין מדובר בתופעת טבע ותיקה אלא בתופעה חסרת תקדים, נאלצתי לחקור ולהציע מודל שיתגבר על מספר קשיים ואתגרים וגם יספק תוצאה טובה שתתבסס על מסקנותיי.

ראשית, היה עליי להבין מושגים חשובים בתחום שוק ההון והמניות ולהגדיר מפורשות איזה ערך המאפיין את המטבע ברצוני לחקור. מבין מספר הקטגוריות השונות המופיעות במאגר המידע בו השתמשתי, החלטתי לחקור את הערך המופיע בשם Weighted_Price שמייצג את ה - Volume Weighted Average Price (WVAP) שלא כמו הערך הפותח או הסוגר של אותו היום ה – WVAP הוא אינדיקטור יותר אמין לסוחרי מניות משום שהוא מתבסס על מספר ערכים: המחיר הסוגר את אותו חלון זמן והגבוהה, הנמוך ונפח (מספר העסקאות שנסגרו בזמן מסוים). יש שיתייחסו אליו כמחיר ממוצע של מניה המהווה מעין סף להגדרת השקעה טובה: אם ההשקעה עלתה פחות משמע שהיא מוצלחת ולהפך.

שנית, במצבו המקורי מאגר המידע בו השתמשתי מכיל מאות מיליוני שורות של נתונים המתעדים כל דקה במשך פרק זמן של יותר מ – 9 שנים. לאור הגודל העצום של מאגר המידע החלטתי לסלק ערכים שלא נקלטו (Nan) ולעבוד בחלונות זמן של שעה אשר הקטינו משמעותית את גודל מאגר המידע.

נוסף על כך, האתגר המרכזי היה למידת התחום DL בדגש על חקירת סדרות עתיות, שם נחשפתי למושגים חדשים ולסוג חדש של שכבות בתוך רשת נוירונים. באמצעות הידע שרכשתי הצלחתי לבצע שינויים ברשת שהגדילו משמעותית את אחוזי ההצלחה שלה.

אני מאמין שבאמצעות הפרויקט שלי משקיעים יוכלו לבצע החלטות שקולות ונבונות יותר בנוגע להשקעתם ולעסקאות שהם מבצעים בכל הקשור למטבע המבוזר – ביטקוין.

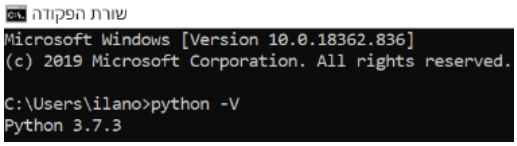
מדריך למשתמש

לפני שאציג את האופן בו מומש הפרויקט, אציג מדריך אשר ינחה את המשתמש כיצד להשתמש בתוכנית.

הוראות התקנה:

1. יש להוריד Python 3.7 או גרסה עדכנית יותר - https://www.python.org/downloads/

אם יש לך Python על המחשב, בדוק מהי גרסתו באמצעות הפקודה: python -V



2. יש להתקין במחשב את סביבת העבודה Anaconda או כל IDE אחר.
לינק להורדת Anaconda - https://www.anaconda.com/products/individual

3. יש להוריד מספר ספריות קוד אשר בהן הפרויקט משתמש:

library name	Installation command	Link
keras	pip install keras	https://pypi.org/project/Keras/
tensorflow	pip install tensorflow	https://pypi.org/project/tensorflow/
matplotlib	pip install matplotlib	https://pypi.org/project/matplotlib/
numpy	pip install numpy	https://pypi.org/project/numpy/
Pandas	pip install pandas	https://pypi.org/project/pandas/
sklearn	pip install -U scikit-learn	https://pypi.org/project/scikit-learn/

4. יש להוריד מן חשבון ה GitHub שלי את הקבצים הבאים:

א. את קבצי ה - python אשר עליהם מתבסס המודל: Main1 ו - gui
ב. את מאגר המידע (קובץ CSV).
ג. את pycache (לא חובה – קובץ מקומפל שמאפשר הרצה מהירה יותר)
ד. את קובץ המודל השמור
ה. את תקיית התמונות Assets.

***אין לשנות את תוכן הקבצים**

5. עדכון הקוד בהתאם ל directories החדשים:

עדכון בקובץ Main1.py :

- path_df פרמטר של פונקציה preprocess- המקום בו שמור קובץ מאגר הנתונים.
- model_path – המקום בו שמור המודל השמור.

```
model_path = r'C:\Users\Admin\pracExc\Project_Bitcoin\trained_model.h5'
data_path =  r"C:\Users\Admin\pracExc\Project_Bitcoin\Bitcoin_History.csv"
```

עדכון בקובץ gui.py :

- path_to_Main1 - המקום של הספרייה המכילה את קובץ Main1.py
- path_assets - המקום של תיקיית Assets

```
path_to_Main1 = r'C:\Users\Admin\pracExc\Project_Bitcoin'
sys.path.append(path_to_Main1)

import Main1

path_assets = 'C:\\Users\\Admin\\pracExc\\Project_Bitcoin\\Assets'
```

הרצת התוכנית + שימוש בממשק משתמש גרפי:

יש להריץ ב Command Line של סביבת העבודה Anaconda את קובץ ה Python:
gui.py.
הרצת קובץ זה תריץ את כל התוכנית.
לאחר הרצת התוכנית ייפתח בפני המשתמש ממשק המשתמש הגרפי:

הממשק מכיל תפריט ראשון עם 2 אפשרויות:

Train Model– לחיצה על כפתור זה תתחיל לאמן את המודל.

בטרם ייפתח התפריט השני התוכנה תאמן מודל בו יהיה ניתן לחזות בזמן אמת באמצעות הטרמינל/ חלון שנפתח להרצת הפקודה:

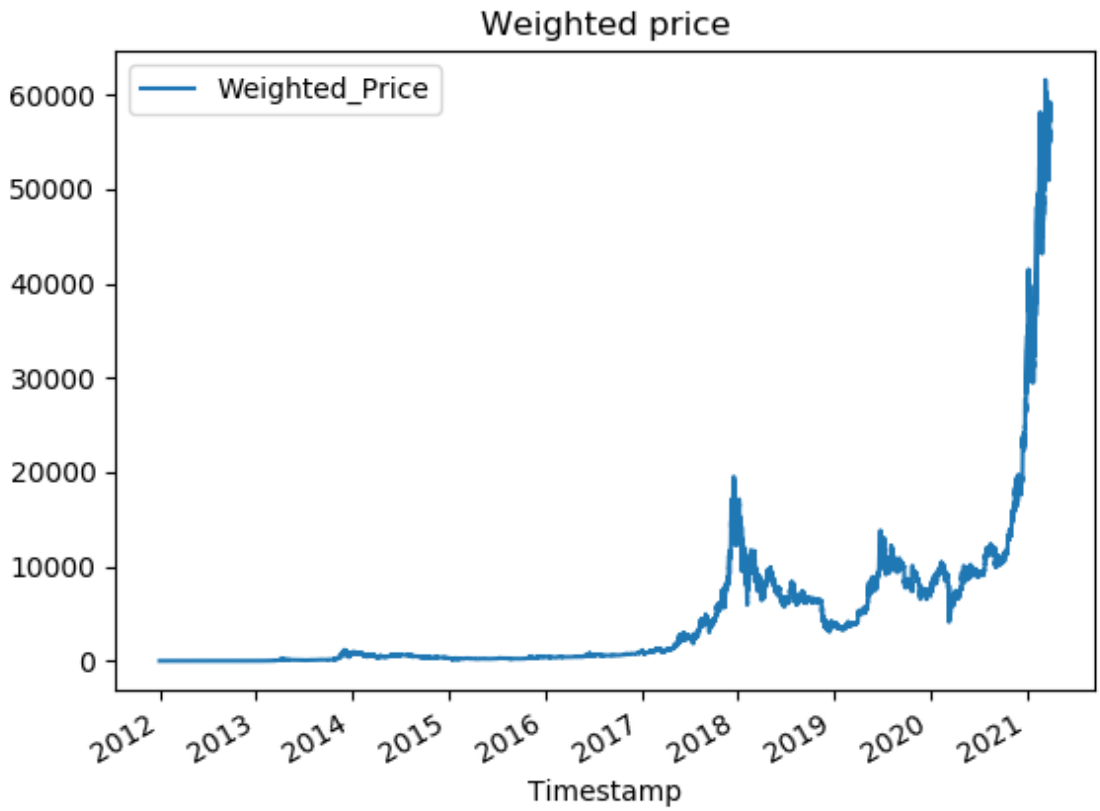
```
54265/54265 [==============================] - 17s 317us/step - loss: 4.3903e-05 - mae: 0.0026
Epoch 2/100
54265/54265 [==============================] - 15s 284us/step - loss: 4.2679e-05 - mae: 0.0023
Epoch 3/100
54265/54265 [==============================] - 16s 301us/step - loss: 4.2342e-05 - mae: 0.0022
Epoch 4/100
54265/54265 [==============================] - 17s 318us/step - loss: 4.2420e-05 - mae: 0.0022
```

Pretrained Model - לחיצה על כפתור זה תדלג על שלב האימון של המודל ותטען את המודל המאומן (התוכנה תשתמש בקובץ trained_model.h5 לטעינת המודל).

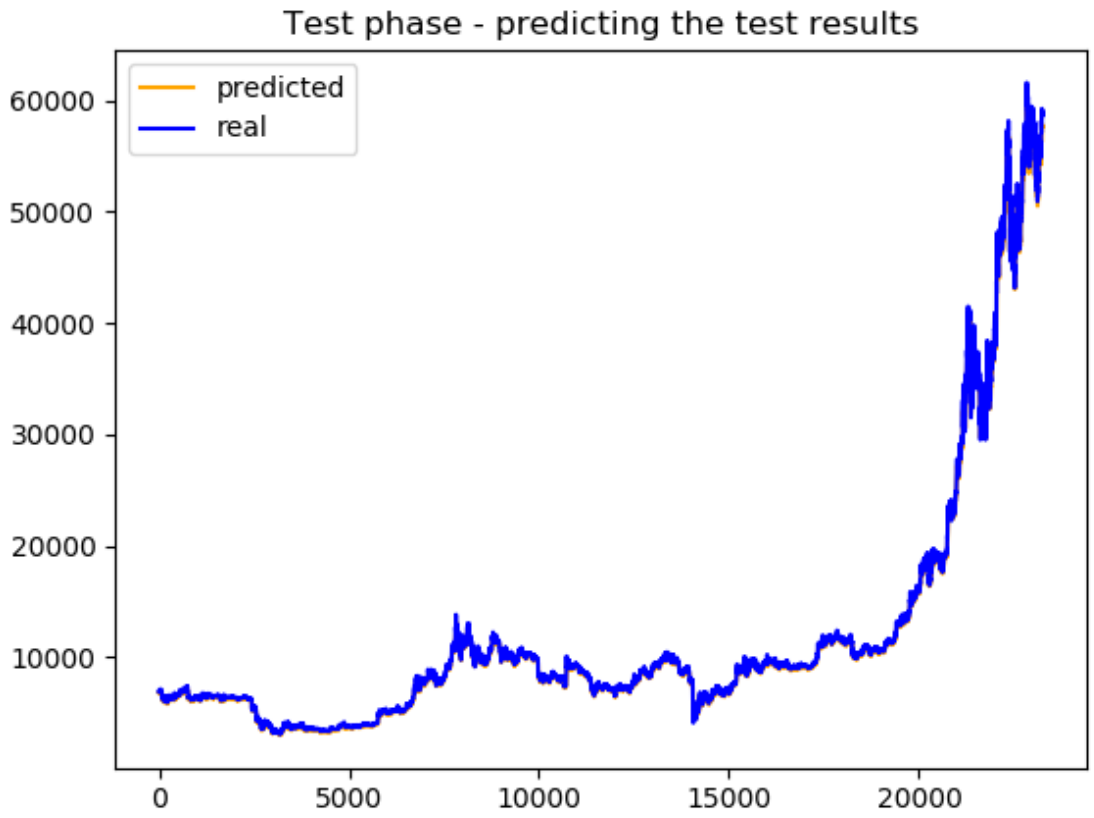
אחרי בחירת אחת משתי האפשרויות המוצעות ייטען תפריט חדש עם 3 אפשרויות:

להלן הסבר על כל אחת מהאפשרויות:

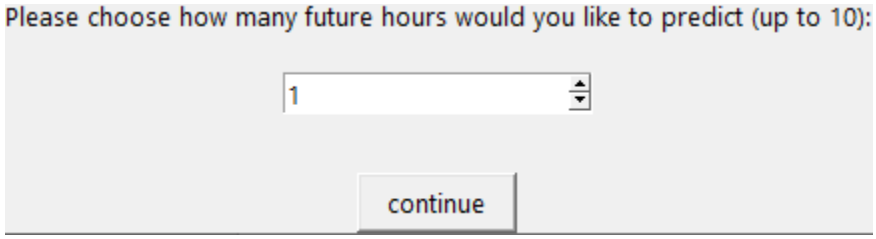
Evaluate Model – אפשרות זו תפתח חלון חדש בו יהיה ניתן לצפות בגרף המקורי של הנתונים כדי להתרשם מהמראה שלו:



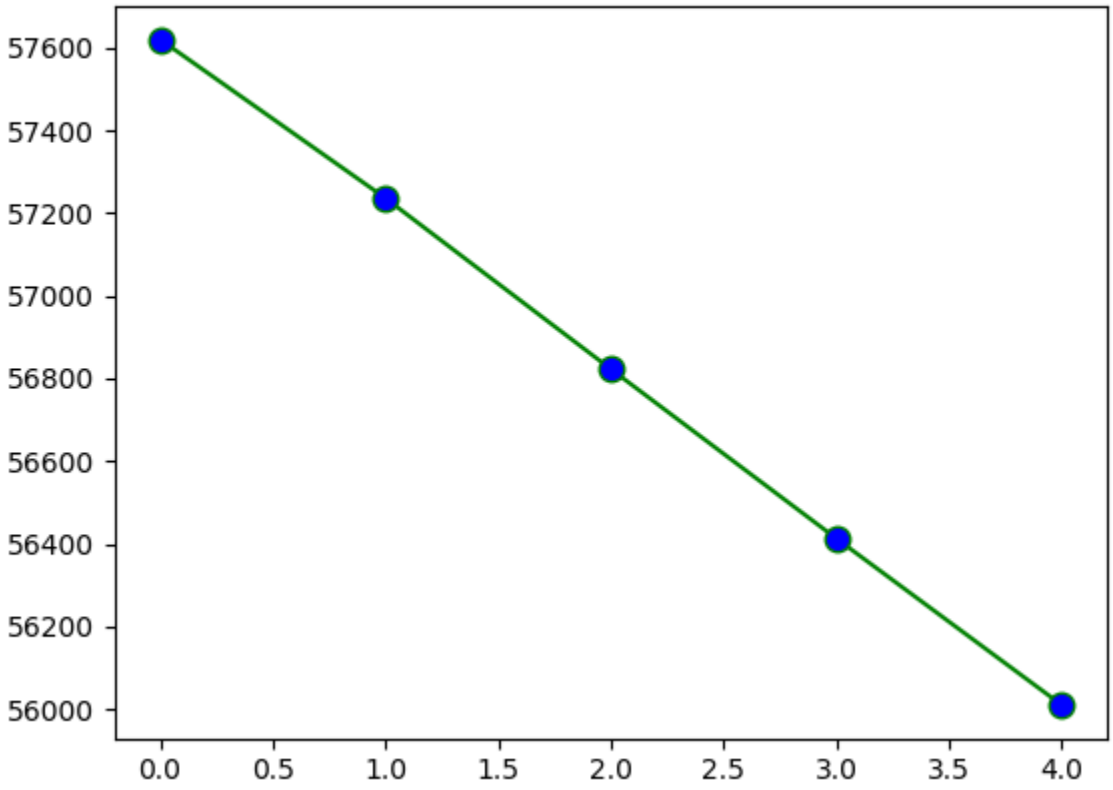
בסגירת החלון הנוכחי ייפתח חלון אינטראקטיבי נוסף, שיציג את החיזוי של חלק ה – test על ידי המודל יחד עם תוצאות האמת כדי שיהיה אפשר לקבל משוב ויזואלי לגבי תפקוד המודל:



Forecast – לחיצה על כפתור זה תפתח חלון חדש עם הוראה לבחור מספר בין 1 ל – 10. מספר זה הוא מספר השעות העתידיות בנתונים שהמודל יחזה. לאחר בחירת המספר יש ללחוץ על continue.



לאחר מכן ייפתח חלון חדש עם גרף המכיל את השעות הבאות אותן חזה המודל (השעה הראשונה מתחילה ב – 0):



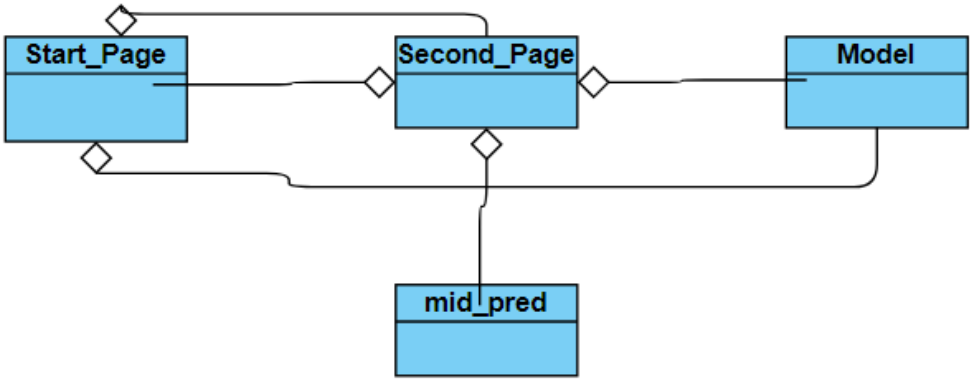
Back – בלחיצה על אפשרות זו יחזור החלון להציג את התפריט הראשוני והמשתמש יוכל לבחור מחדש אם לאמן את המודל או לטעון מודל מאומן.

מדריך למפתח:

ראשית כל, הפרויקט שלי מחולק למספר קבצי קוד אשר לכל אחד ישנו תחום אחריות שונה. חלוקה זו בין חלקי הקוד השונים העוסקים בחלקים שונים בפרויקט מאפשרת ארגון קוד, ממזערת באגים למיניהם ואף אפשרה לי לבצע את הפרויקט ביתר קלות.
כעת אציין את שמות הקבצים השונים ותחום האחריות שהוטל על כל אחד מהם:

gui.py	קובץ זה מנהל את כל התוכנית לפי בחירות המשתמש – זהו הקובץ הראשי הכולל בתוכו את הגדרת ממשק המשתמש
Main1.py	קובץ זה אחראי על בניית מודל והשימושים השונים בו

Class diagram זו מציגה את המחלקות השונות בפרויקט.



כעת אעבור על המחלקות השונות והפונקציונאליות הכלולה בהם:

שם המחלקה	מטרתה	מזמנת את	מזומנת על ידי	נמצאת בקובץ
Model	מכילה את המודל ואת הפונקציונליות הקשורה אליו: טיפול מקדים במאגר המידע, אימון מודל, טעינת מודל, חיזוי ועוד...	-----	Start_Page, Secon_Page	Main1.py
Start_Page	מכילה את הגדרות והאלמנטים המרכיבים את התפריט הראשון, כולל פונקציונאליות הקשורה למודל: אימון מודל וטעינת מודל מוכן ובנוסף גם פונקציה ליצירת מופע חדש של Second_Page	Model Second_Page	Second_Page	gui.py
Second_Page	מכילה את הגדרות והאלמנטים המרכיבים את התפריט השני, כולל פונקציונאליות הקשורה למודל: חיזוי והצגת גרפים ובנוסף גם פונקציה ליצירת מופע חדש של Start_Page	Model Start_Page mid_pred	Start_Page	gui.py
Mid_pred	מכילה הגדרות ואלמנטים המרכיבים חלון ביניים לאחר לחיצה על	-----	Second_Page	gui.py

	אפשרות החיזוי, שמאפשרים לקבוע הגדרה ספציפית של החיזוי			

ממשק המחלקה Model:

הפונקציה	תפקידה
__init__(self)	הבנאי של המחלקה. תפקידה לבצע השמה של תכונות המחלקה בהן יחול שימוש בפונקציות השונות
predict_plot_future(self,times)	מקבלת מספר שלם המייצג שעות בעתיד (אחרי השעה האחרונה במאגר המידע), חוזה את הערכים של שעות אלו ומציבה אותן על קנבס גרף אינטראקטיבי ביחד עם קו מגמה. **אופן ביצוע**: כפי שמצוין בתיאור תפקידה של פונקציית האימון, המודל משתמש בחלונות כדי לנבא, כלומר באמצעות מספר קבוע של תצפיות מהעבר שמתקבל כקלט ברשת הנוירונים, חוזה הרשת את הערך הבא. אם כן כאשר מתקדם החלון בזמן עולה הצורך בתצפיות חדשות, אך משום שהשעות העתידיות לא מתועדות במאגר הנתונים חלק או כל התצפיות הדרושות לחיזוין יהיו חסרות. על בעיה זו התגברתי כאשר השלמתי את חלונות הזמן עם תצפיות שחזה המודל. נוסף על כך, בחיזוי ובאימון מקבל המודל כקלט, קלט מותאם לסקאלה של נתונים בין 0 ל – 1 ומחזיר פלט באותה סקאלה. אם כן על מנת לשחזר ערכים אמיתיים של המטבע נאלצתי להשיג מדד סטטיסטי מחלק ה – test של מאגר המידע שאפשר להחזיר את הערכים לסקאלה מציאותית.
predict_plot_test(self)	חוזה ערכים בתחום ה – test ומציג אותם על גרף אינטראקטיבי לצד ערכים אמיתיים ממאגר המידע. גם פונקציה זו משתמשת בנתונים סטטיסטים של מאגר הנתונים בתחום ה – test כדי להמיר את התחזיות לערכים בסקאלה מציאותית.
retrieve_model (self,saved_model_path)	מקבלת את הנתיב במחשב בו נשמר קובץ המודל אשר נגמר בסיומת h5 וטוענת אותו לתוך התכונה model של המחלקה.
train (self)	מאמנת את המודל. עושה שימוש בחלק ה – train שפוצל לחלונות בגודל מתאים, בגודל batch אידיאלי שנמצא לאחר בדיקה של המודל ומספר איפוקים לאימון המודל.
print_shapes(self)	נותנת מידע כללי על גודל תכונות שונות של המחלקה, ספציפית תכונות שנוצרו על ידי חלוקה של מאגר המידע בפונקציה - preprocess.

@staticmethod preprocess (path_df, partition)	פעולה סטטית שמקבלת את הנתיב בו שמור מאגר המידע במחשב ומחזירה מספר משתנים המהווים חלקים מותאמים של המאגר החיוניים לתהליכים שונים
@staticmethod split_sequence(sequence, to_predict, n_steps)	פעולה סטטית שנועדה להשלים את פעילותה של פונקציית הטיפול המקדים במאגר המידע. היא מסדרת קטעי מידע בחלונות והופכת אותם לקלט בגודל הרצוי לאימון ולחיזוי של המודל.
@staticmethod make_model ()	פעולה סטטית שנועדה לבנות את המודל האידיאלי ולהחזיר את האובייקט שנוצר.

ממשק המחלקה Start_Page:

הפונקציה	תפקידה
__init__(self, master,path_to_asset)	הבנאי של המחלקה. תפקידה, תוך שימוש ב- module tkinter הוא לבנות את התפריט הראשון בממשק המשתמש כאשר היא מקבלת את הפרמטר master – השלד הויזואלי עליו היא מארגנת את האלמנטים השונים ו – path_to_asset הנתיב בו שמורה התיקייה Assets במחשב שמשמשת שילוב של תמונות בממשק.
On_Train()	עושה שימוש במשתנים שהוגדרו באותו scope ובמשתנים גלובליים כדי לאמן את המודל וליצור מופע חדש של Second_Page
On_Pretrain()	עושה שימוש במשתנים שהוגדרו באותו scope ובמשתנים גלובליים כדי לטעון מודל מאומן וליצור מופע חדש של Second_Page

ממשק המחלקה Second_Page:

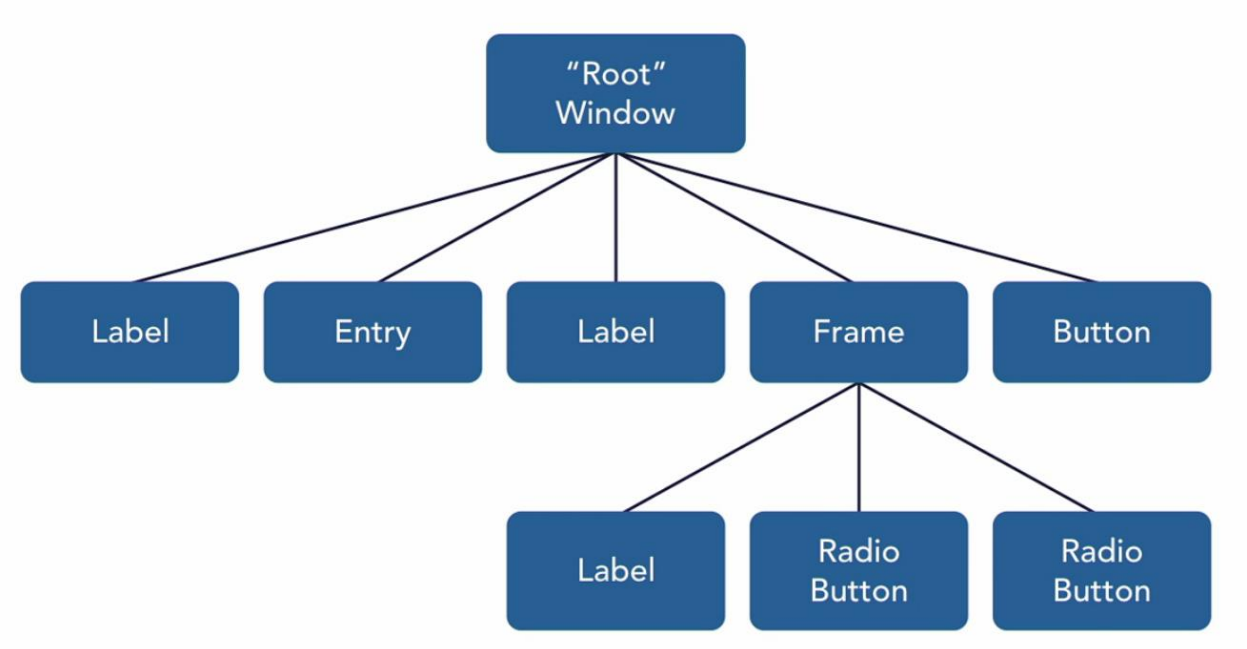
הפונקציה	תפקידה
__init__(self, master,path_to_asset)	הבנאי של המחלקה. תפקידה, תוך שימוש ב- module tkinter הוא לבנות את התפריט השני בממשק המשתמש כאשר היא מקבלת את הפרמטר master – השלד הויזואלי עליו היא מארגנת את האלמנטים השונים ו – path_to_asset הנתיב בו שמורה התיקייה Assets במחשב שמשמשת שילוב של תמונות בממשק.
eval_window()	עושה שימוש במשתנים גלובליים כדי להציג גרף אינטראקטיבי של המשתנה הנחקר במאגר המידע לאורך זמן, ואחר כך מציגה גרף זה ביחד עם גרף התצפיות שנחזו בתחום test כדי לקבל הערכה ויזואלית - כללית על ביצועיו של המודל

forecast_window()	מזמנת מופע חדש של mid_pred

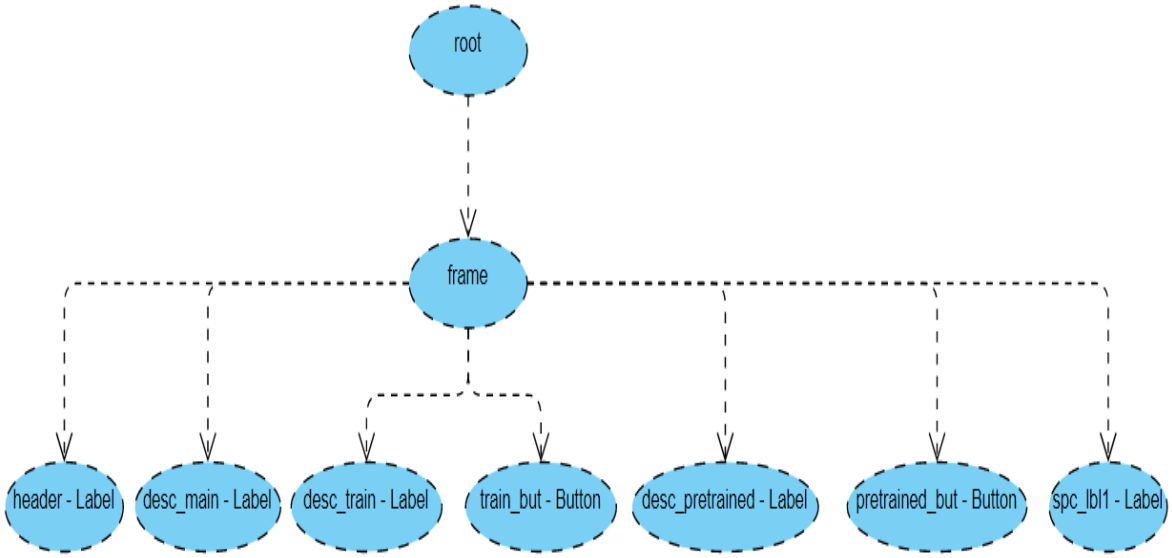
ממשק המחלקה mid_pred:

הפונקציה	תפקידה
__init__(self,master)	הבנאי של המחלקה. תפקידה, תוך שימוש ב- module tkinter הוא לבנות את חלון הביניים טרם שימוש בחיזוי המודל בממשק המשתמש כאשר היא מקבלת את הפרמטר master – השלד הויזואלי עליו היא מארגנת את האלמנטים השונים.
to_preds()	עושה שימוש במשתנים שהוגדרו באותו scope ובמשתנים גלובליים כדי לחזות תצפיות עתידיות במודל ולהציג אותן על גרף אינטראקטיבי

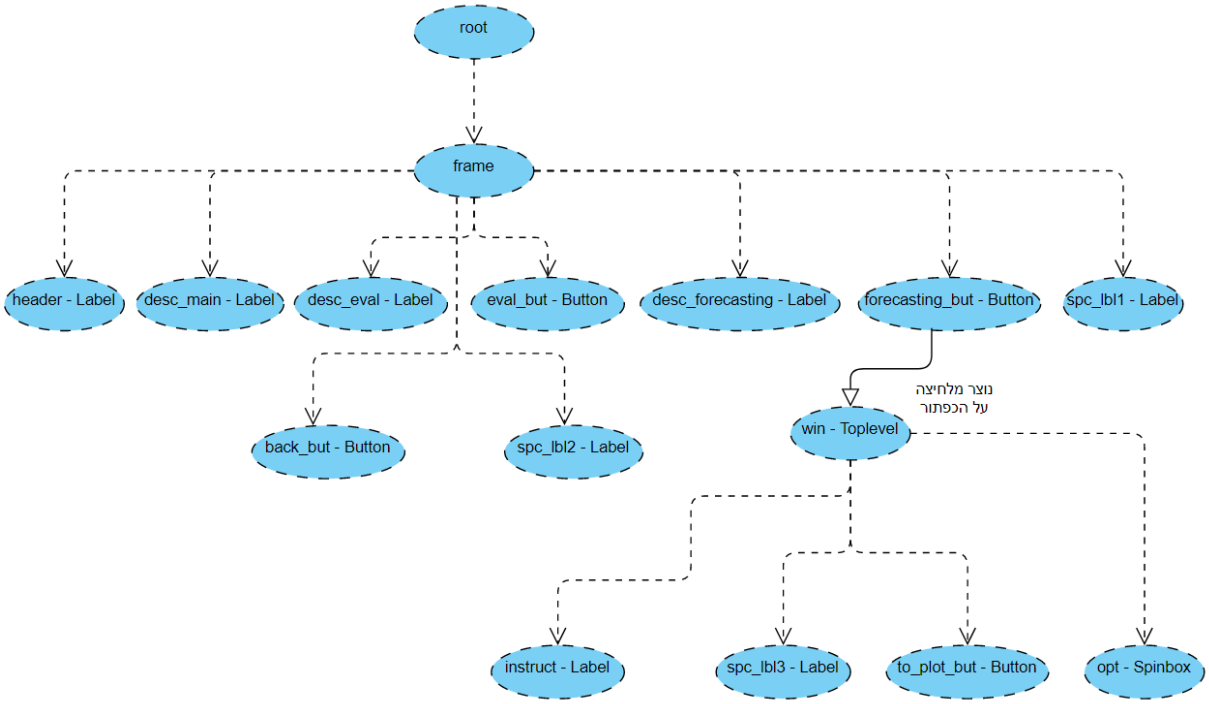
להלן תמונה ויזואלית לפירוט פריסה שרירותית של אלמנטים ויזואליים השייכים ל – tkinter (אלמנטים נפרסים על גבי אלמנטים אחרים – האלמנט המכיל נמצא למעלה בישר המחבר בין שניהם והמוכל למטה):



להלן תרשים הכלה של אלמנטים ויזואליים במחלקה Start_Page :



להלן תרשים הכלה של אלמנטים ויזואליים במחלקה Second_Page + mid_pred :



כעת אציג כמה מן המשתנים המרכזיים ותפקידם:

תפקיד	שם משתנה
מוגדר בפונקציית main בקובץ gui ומטרתו להוות השלד הוויזואלי בממשק המשתמש	root
משתנה גלובלי המוגדר בפונקציית main בקובץ gui ומטרתו להוות הפנייה למופע חדש של המחלקה Model בקובץ Main1	mod
תכונה של המחלקה Model האחראית על קביעת גודל חלון הזמן (כמה תצפיות יכלול כל חלון)	self.window_size
משתנה מסוג panda.df המכיל את מאגר המידע לאחר שבוצע עליו טיפול מקדים	self.df
אותו משתנה למעלה המוכל בתוך מערך דו מימדי של המודול numpy	self.dataset
משתנה שנועד לקבוע את הסקאלה בה נתונים יוצגו – במקרה זה מינימום – מקסימום סקאלה	self.scaler
החלק המיועד לאימון במודל	self.train_set
אותו דבר מוצג בסקאלה שנקבעה	self.sc_train_set
החלק המיועד לבחינת המודל	self.test_set
אותו דבר מוצג בסקאלה שנקבעה	self.sc_test_set

מסקנות הרצת המודל

לפני שאנתח את תוצאות המודל ואתאר את הניסיונות שהביאו לתוצאות הסופיות אסביר מספר מושגים נדרשים:

<u>משתנה בלתי תלוי – סדרה עתית</u>:

מערך הנתונים מתחלק למשתנה תלוי (שאינו קבוע) ומשתנה בלתי תלוי קבוע שהוא הזמן (T) בו נמדד המשתנה הבלתי תלוי (שנים, חודשים,ימים, שעות, דקות...).

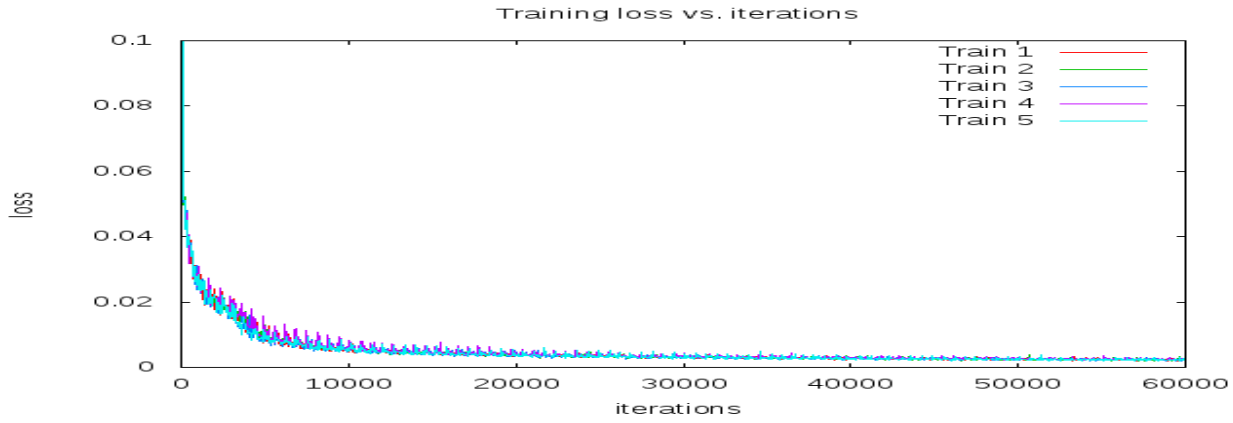
<u>מערך נתונים של חלונות:</u>

זהו עיבוד מקדים של הנתונים לפיו מחלקים את הנתונים לקבוצות קטנות בעלות גודל קבוע של נתונים עוקבים הנקראים חלונות.

```
x =  [[5 6 7 8]
 [4 5 6 7]]
y =  [[9]
 [8]]
x =  [[1 2 3 4]
 [2 3 4 5]]
y =  [[5]
 [6]]
x =  [[3 4 5 6]
 [0 1 2 3]]
y =  [[7]
 [4]]
```

<u>שגיאה (loss/error):</u>

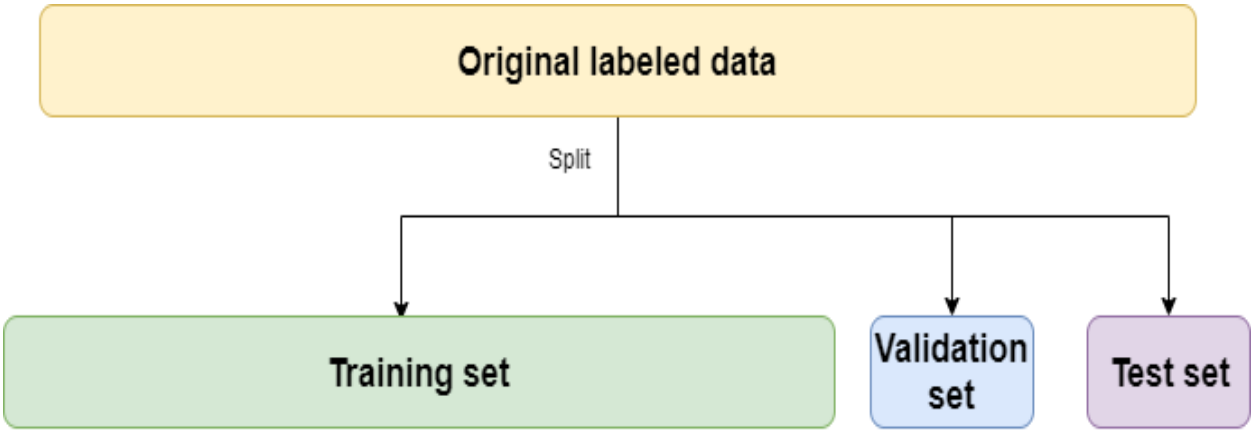
השגיאה היא מושג שבאמצעותו ניתן לכמת את מידת האי - דיוק של המודל, כלומר עד כמה שונה הפלט מהנתונים שנאספו בפועל. ניתן למדוד את השגיאה באמצעות פונקציות שונות. פונקציות אלו מאפשרות את תהליך ה - back propagation וכך למעשה את משתפר המודל בכך שהשגיאה קטנה.



אימון,בדיקה, אימות (train,test validation):

אימון רשת נוירונים מתבסס על דוגמאות. כדי לקבל רשת יעילה, שתעבוד על מקרים שלא ראתה בעבר, יש לאמן אותה על הרבה דוגמאות שונות המייצגות את הקלטים המציאותיים. מציאת מאגר הנתונים מהווה לעיתים קרובות אתגר בפני עצמו.
כדי לבחור במודל המיטבי, קיימת שיטת חלוקה של הנתונים שמאפשרת לאמן את המודל, להעריך אותו ולהשוות את ביצועיו מול מודלים אחרים. להלן חלוקת הנתונים:
תת- מאגר לאימון: מהווה כ- 50% מכלל הנתונים. אלו הם הנתונים בהם משתמשים לאמן את המודל.
תת- מאגר לבדיקה: מהווה כ - 25% מהנתונים. משתמשים בנתונים אלה כדי להשוות בין המודלים השונים ולבחור במודל עם הביצועים הטובים ביותר.
תת - מאגר לאימות: מהווה כ - 25% מהנתונים. משתמשים בנתונים אלו כדי לבדוק את רמת הדיוק של המודל הנבחר.

יש לציין כי הערכה של המודלים מתבצעת באמצעות השוואת ערכים הנוצרים על ידי המודל לעומת ערכים מציאותיים ונמדדת באמצעות פונקציית שגיאה (loss function).



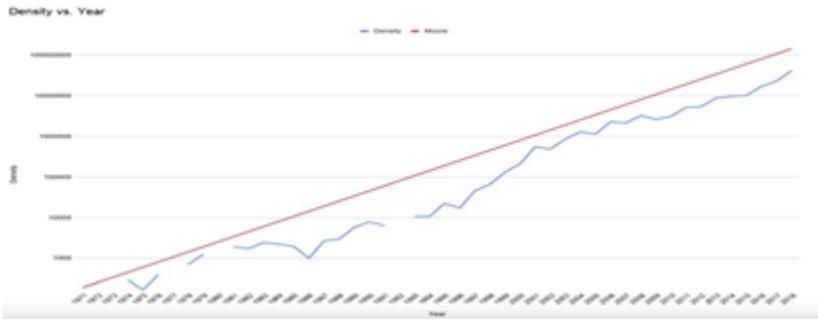
Overfitting- "התאמת יתר" היא מצב בו המודל מותאם יתר על המידה למאגר אותו הוא לומד ופחות מצליח בביצוע תחזיות של המאגר אותו הוא לומד. אנו נזהה מצב זה כאשר ה test / validation loss גדול משמעותית מן ה training loss.

Under fitting – מצב ההפוך מ overfitting, אשר בו מאגר הלמידה הינו פשוט מידי ואינו כולל מספיק נתונים ללמידה. במצב זה המודל אינו מצליח ללמוד את התנהגות הגרף שכן הוא בלתי אפשרי ללמידה. נזהה מצב זה כאשר ה test / validation loss משמעותית מן ה training loss.

Num epoch- מספר הפעמים בהם יתבצע תהליך הלמידה מחדש.

מאפיינים של סדרות עתיות:

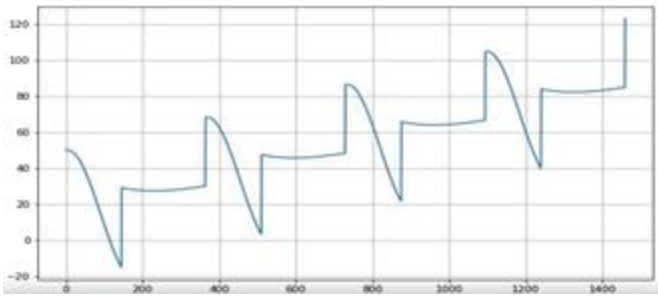
מגמה(טרנד)- מאפיין של גרף כאשר יש לו כיוון ספציפי אליו הוא שואף לאורך הזמן. לדוגמא, לגרף הבא יש מגמה חיובית:



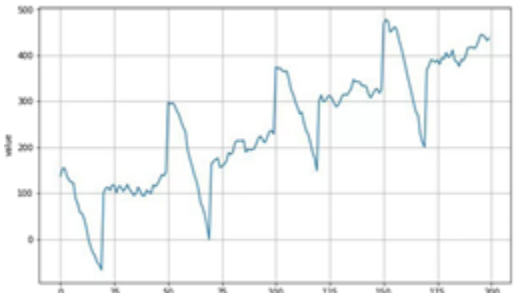
.

מחזוריות/ עונתיות- מאפיין של גרף כאשר יש דפוס חוזר במרווחים צפויים. לדוגמא, לגרף הבא יש דפוס מחזורי כל שבוע (חנות ששיא העסקים לקראת סוף השבוע ואז דועכת).

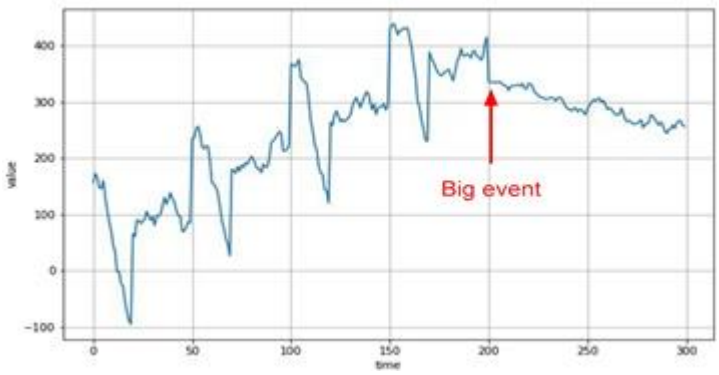
הנה גרף שיש בו גם מחזוריות וגם מגמה חיובית:



רעש- מכיוון שבמציאות קורים דברים רנדומליים שאותם לא ניתן לצפות , יש לגרף מאפיין נוסף שנקרא רעש והוא הקפיצות האקראיות שקורות לאורך הגרף. הנה דוגמא לגרף שבו יש מחזוריות, מגמה ורעש שמסמל גרף יותר מציאותי ממקודם.



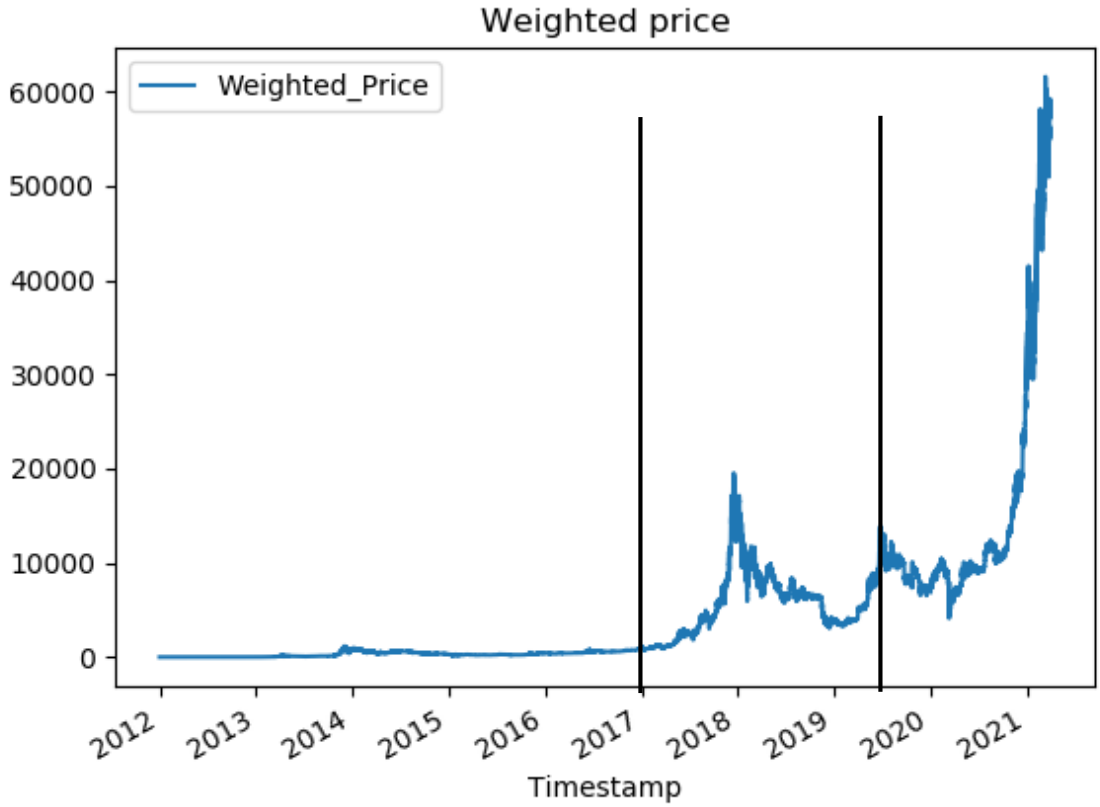
<u>גרף סטיישונרי-</u> גרף סטיישונרי הוא גרף שהתנהגותו לא משתנה לאורך זמן (כל הגרפים שהצגתי עד כה היו סטיישונרים). בגלל אירועים מסוימים שקורים כמו נפילה כלכלית של חברה , הכנסת רפורמה שמשנה התנהגות וכו'. בגרף הבא יש מחזוריות , מגמה ורעש עד לנקודה 200 בזמן בה קרה אירוע מסוים ששינה את התנהגות הגרף. חשוב לזהות מקרים כאלה כדי שנוכל לאמן את המודל על הגרף בצורה מהימנה.



תהליך הקשת המסקנות

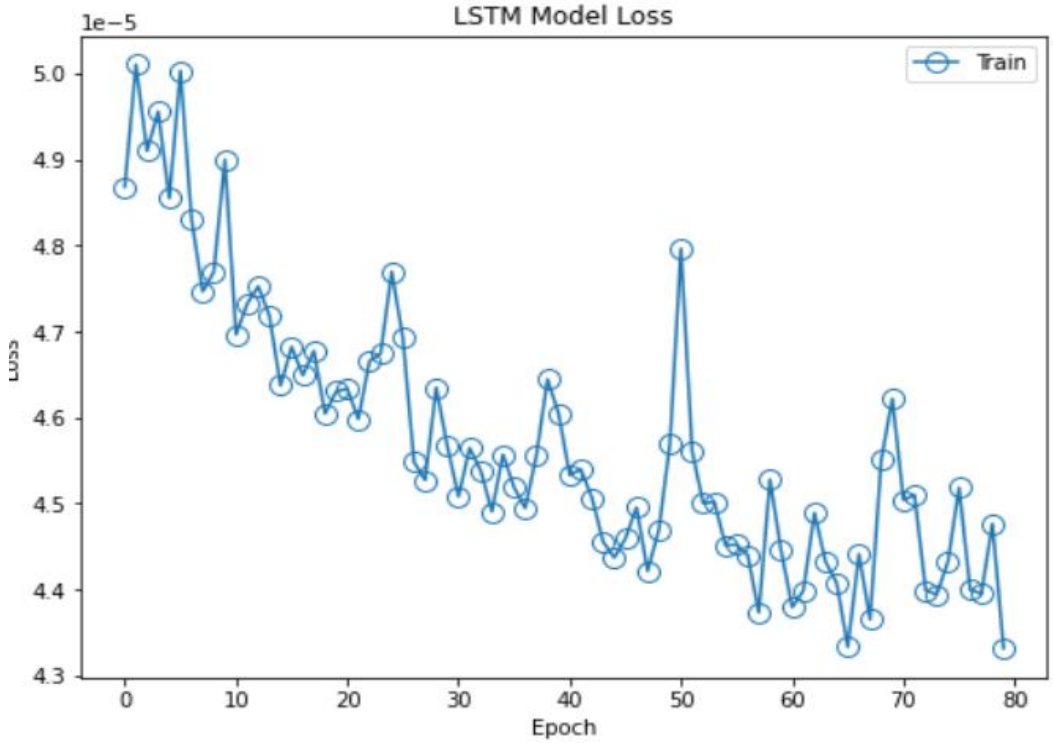
כעת אפרט על התהליך שהוביל אותי לשיפור המודל ולמציאת הגרסה האידיאלית שלו תוך שאציג את המסקנות המבקשות:

תחילה פיצלתי את מאגר המידע לפי החלוקה המתוארת תחת בכותרת : אימון, בדיקה, אימות (50,25,25). עם זאת נוכחתי לדעת שבכל מודל שאימנתי נוצר מצב של overfitting, כלומר ערכי השגיאה היו נמוכים בשלב האימון אך בשלב הבדיקה והאימות ניכר שתוצאות החיזוי שונות מהותית מהתוצאות האימות ובבדיקה ויזואלית היה ניתן לראות שבאופן חד משמעי לא קיימת הלימה בין הגרפים. בשביל לפתור מצב זה החלטתי להתבונן לעומק ולנסות לנתח את המידע שאיתו אימנתי את המודל (אציג פעם נוספת את גרף הנתונים ממאגר הנתונים) :

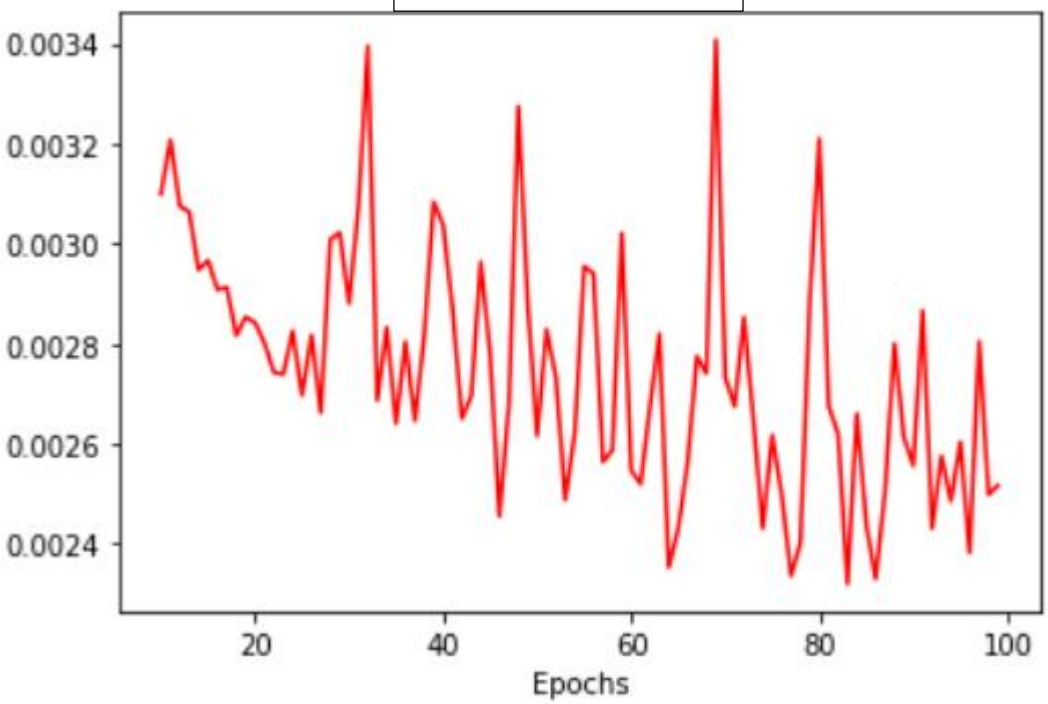


במבט על ניתן לזהות לפי המחיצות שהצבתי באופן איכותי כי המחצית הראשונה של המידע שונה כמעט לחלוטין מהמחצית המאוחרת יותר. ידיעה זו גרמה לי להחליט להגדיל את כמות הנתונים המיועדים לאימון המודל ואף לבטל את ה - validation. אציין שגם ניסיתי להשתמש רק במחצית השנייה של הנתונים – דבר שהוביל ל - Under fitting. יתר על כן, הקטנת גודל החלונות (כפי שפירטתי במבוא) גרמה למודל להתמקד בחלונות זמן קטנים יותר ולנסות להבין מגמות ומחזוריות בזמנים קצרים בהרבה. אמנם במבט על נראה כי הגרף לא סטיישונרי אך במרווחי זמן קצרים הוא אכן כך וזה מתבטא באמצעות ערכי השגיאה הקטנים של המודל בשלב הבדיקה וגם בבדיקת גרף סטיישונרי (נמצא בנספחים).

להלן חלק מגרפי התוצאות של הלמידה מן המאגר המחודש והחלונות הקטנים:



שגיאה ממוצעת אבסולוטית



רפלקציה/סיכום אישי:

ביצוע פרויקט זה אינו היה מטלה פשוטה כפי שציינתי בפרקי המבוא ומסקנות הרצת המודל. הואיל ובחרתי להתעמק בנושא המטבע המבוזר היה עליי לשפר את ידיעתי בתחום ולחקור על נושא חדש לחלוטין. כמו כן, גם הנושא של חקירת סדרה עתית היה חדש בעבורי ולא פעם קרה שהצטרכתי לעבוד ביחד עם חבריי ולהיעזר במורה כדי להבין מושג או שיטה מסוימת. במבט לאחור אני שמח שבחרתי עם נושא זה משום שיצא לי לעבוד בעבודת צוות יחד עם חברי לכיתה בהבנת החומר וגם כי הצלחתי להתגבר על אתגרים וקשיים שגרמו להסתכל מחוץ לקופסה ובנוסף העשרתי את הידע שלי לא רק בנושא "למידה עמוקה" אלא גם באחרים.

ביבליוגרפיה:

– Medium

https://towardsdatascience.com/simple-multivariate-time-series-forecasting-7fa0e05579b2

https://medium.com/analytics-steps/introduction-to-time-series-analysis-time-series-forecasting-machine-learning-methods-models-ecaa76a7b0e3

https://machinelearningmastery.com/how-to-get-started-with-deep-learning-for-time-series-forecasting-7-day-mini-course/

– machine learning mastery

https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

– kaggle

https://www.kaggle.com/vigneshsubramanians/time-series-analysis

– tensorflow

https://www.tensorflow.org/tutorials/structured_data/time_series

– towardsai

https://towardsai.net/p/deep-learning/beginners-guide-to-timeseries-forecasting-with-lstms-using-tensorflow-and-keras-364ea291909b

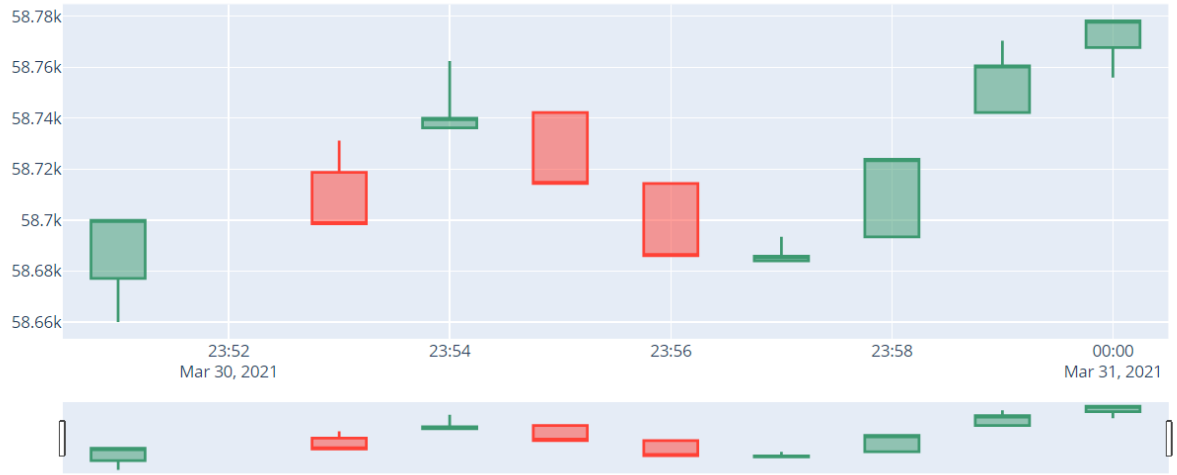
קורסים:

Coursera

https://www.coursera.org/learn/tensorflow-sequences-time-series-and-prediction

נספחים:

גרף נרות יפניים של 10 הדקות
האחרונות במאגר המידע



בדיקת קורלציה של ספירמן לבדיקת
הקשר בין המשתנים במאגר המידע

	Open	High	Low	Close	Volume_(BTC)
Open	1.000000	0.999999	0.999999	0.999998	-0.016767
High	0.999999	1.000000	0.999998	0.999999	-0.016362
Low	0.999999	0.999998	1.000000	0.999999	-0.017214
Close	0.999998	0.999999	0.999999	1.000000	-0.016776
Volume_(BTC)	-0.016767	-0.016362	-0.017214	-0.016776	1.000000
Volume_(Currency)	0.607892	0.608217	0.607533	0.607884	0.752208
Weighted_Price	0.999999	0.999999	0.999999	0.999999	-0.016817

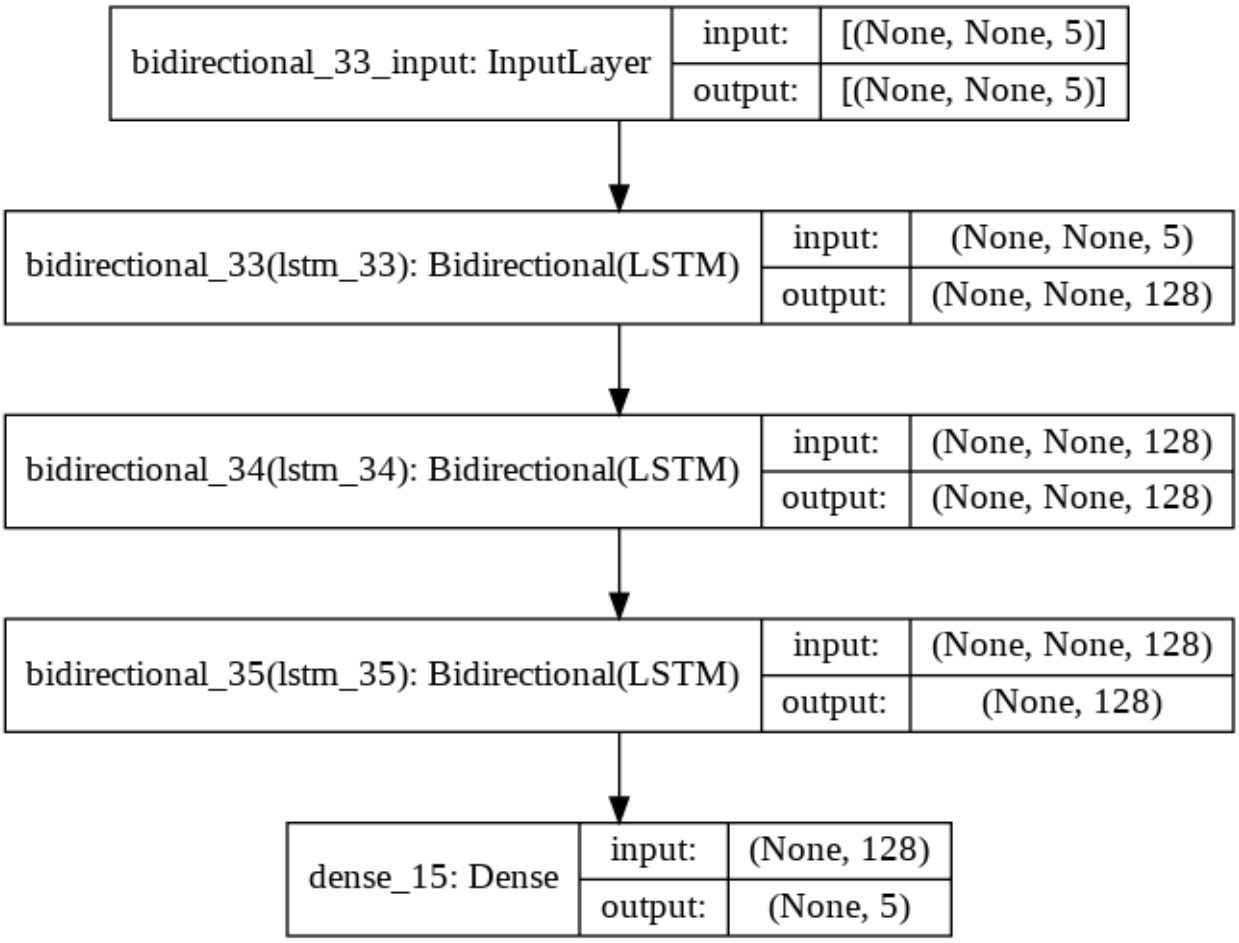
	Volume_(Currency)	Weighted_Price
Open	0.607892	0.999999
High	0.608217	0.999999
Low	0.607533	0.999999
Close	0.607884	0.999999
Volume_(BTC)	0.752208	-0.016817
Volume_(Currency)	1.000000	0.607853
Weighted_Price	0.607853	1.000000

בדיקת דיקי פולר
לבדיקת גרף סטיישונרי

```
Column: Weighted_Price
Test Statistic                  -36.343205
p-value                           0.000000
# Lags                           38.000000
# Observations                49783.000000
Critical Value (1%)              -3.430481
Critical Value (5%)              -2.861598
Critical Value (10%)             -2.566801
dtype: float64

Series is Stationary
```

שכבות המודל:


bidirectional_33_input: InputLayer
input:
[(None, None, 5)]
output:
[(None, None, 5)]
bidirectional_33(lstm_33): Bidirectional(LSTM)
input:
(None, None, 5)
output:
(None, None, 128)
bidirectional_34(lstm_34): Bidirectional(LSTM)
input:
(None, None, 128)
output:
(None, None, 128)
bidirectional_35(lstm_35): Bidirectional(LSTM)
input:
(None, None, 128)
output:
(None, 128)
dense_15: Dense
input:
(None, 128)
output:
(None, 5)