

שם בית הספר: מקיף י"א ראשונים  
 שם העבודה: Forecasting electricity consumption  
 שם התלמיד: שגיא זפראני  
 ת.ז.: 328460340  
 שם המורה: דינה קראוס

תאריך ההגשה: 17.6.2023

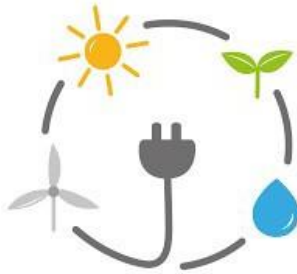


## תוכן עניינים

3	מבוא	.1
5	מבנה\ארכיטקטורה	.2
10	מדריך למשתמש	.3
25	מדריך למפתח	.4
34	בסיס הנתונים	.5
35	מסקנות הרצת המודלים	.6
37	דיון	.7
38	סיכום אישי ורפלקציה	.8
39	נספחים	.9
42	ביבליוגרפיה	.10

## 1. מבוא

תעשיית האנרגיה מסתמכת על חיזויים בשביל תכנון מערכות אנרגיה חסכוניות, שיענו על צרכי השימוש בדרך האפקטיבית ביותר.



דוגמה אחת לכך היא המעבר לתחבורה המבוססת על חשמל ומערכות חימום חשמליות. המעבר לשימוש בחשמל נובע מהצורך להפחתת גזים מזהמים, ולהפחתת הוצאות על ייצור אנרגיה. מערכות אלו מעלות את הצורך לדעת מתי וכמה חשמל משומש על ידי צרכנים. מידע זה חיוני כדי שאספקת החשמל תהיה יעילה.

בנוסף, המידע לגבי השימוש בחשמל חיוני גם לעיצוב מערכות אנרגיה מתחדשת.

למערכות אנרגיה מתחדשת יש יתרונות רבים – הן מעניקות עצמאות אנרגטית, מפחיתות את זיהום הסביבה וחוסכות את עלות האנרגיה.

השנה אנחנו, תלמידי מגמת הנדסת תוכנה בבית הספר ראשונים עשינו פרויקט בתחום שנקרא Machine learning או בעברית – למידת מכונה.

למידת מכונה היא תת תחום של ענף מחקר בין תחומי שנקרא בינה מלאכותית החותר להקנות למחשב יכולת חשיבה ויכולת ביטוי הקרובות לדרכי החשיבה והביטוי האנושיות בעזרת בפיתוח אלגוריתמים המיועדים לאפשר למחשב ללמוד מתוך ניסיונות ונתונים. במילים אחרות המחשב לקבל תחזיות או החלטות מבלי להיות מתוכנת מפורשות לעשות זאת.

למידה עמוקה היא תת-תחום של למידת מכונה העוסקת באלגוריתמים הנקראים רשתות נוירונים, וזאת בהשראת המבנה והתפקוד של המוח הבנוי מתאי עצב (נוירונים) המקושרים זה לזה.

קיימים כמה סוגים של אלגוריתמים של אלגוריתמים ליישום למידת מכונה:

למידה מונחית (supervised learning) – בלמידה זו, לכל קלט/דוגמה (sample) שעליה מאומן המודל יש תווית (הפלט הרצוי עבור אותו הקלט).

למידה בלתי מונחית (unsupervised learning) – בלמידה זו, אין לדוגמאות שמשמשות לאימון המודל תוויות.

למידת חיזוק (Reinforcement learning) – בלמידה זו, אלגוריתם הלמידה מקבל משוב חלקי על ביצועיו עליו להסיק אילו מהחלטותיו הביאו להצלחה/כישלון.

אחת מהדרכים לבצע חיזוי היא חיזוי סדרת זמן. סדרת זמן היא סדרת תצפיות של נתונים המסודרת לפי זמן, בדרך כלל במרווחי זמן שווים בין כל תצפית.

מושגים בסיסיים שכדאי לדעת על סדרות זמן:

Timestep - נקודה בזמן שיש בה תצפית.

Seasonality (מחזוריות) - מאפיין של סדרת זמן כאשר קיים בה דפוס חוזר במרווחי זמן שווים.

Trend (מגמה) - מגמה היא תבנית בסדרת הזמן המציגה את תנועת הסדרה לערכים גבוהים או נמוכים יחסית לאורך תקופה ארוכה.

Noise (רעש) - הקפיצות האקראיות שקורות לאורך סדרת הזמן. נובע מאירועים אקראיים שלא ניתן לצפות.

Stationary time series - סדרת זמן שהממוצע, השונות (ראה נספח 4-שונות) והשונות המשותפת (ראה נספח 5-שונות משותפת) שלה אינם משתנים עם הזמן.

Frequency (תדירות) - מרווח הזמן בין שני תצפיות בסדרת זמן.

קיימות כמה גישות לביצוע חיזוי סדרות זמן:

Vector autoregression - שיטה סטטיסטית המשמשת למציאת קשר לינארי בין משתנים לאורך זמן ולפי קשרים אלו, מבצע חיזויים.

Neural networks - שיטה המבוססת על רשתות נוירונים (בעיקר Reccurent neural networks).

Auto Regressive Integrated Moving Average - שיטה סטטיסטית המבוססת על רגרסיה וממוצע נע.

Exponential smoothing - שיטה המבוססת על מתן משקל כגדול יותר לאירועים קרובים מאשר לאירועים רחוקים בחישוב ממוצע, בניגוד לממוצע נע שמתייחס לכל אירוע באופן שווה, כדי לבצע חיזויים.

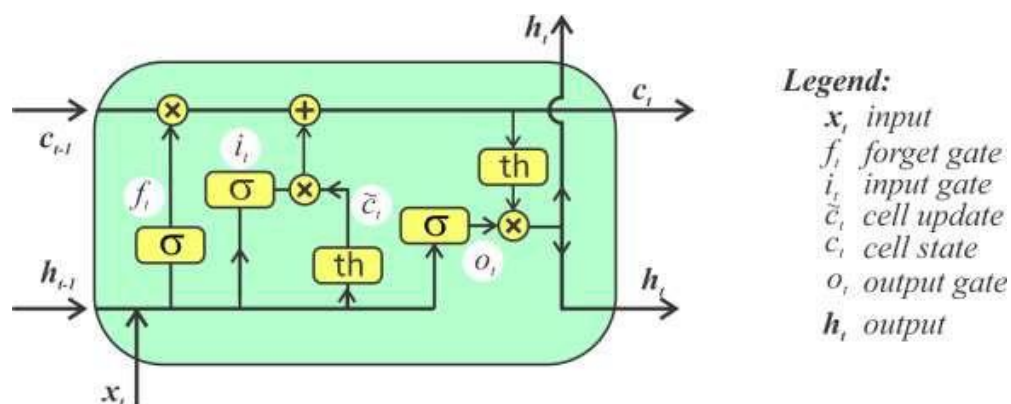
בשנים האחרונות עם התפתחות תחום הלמידה העמוקה, גדלה הפופולאריות והשתפרו הביצועים של רשתות הנוירונים לחיזוי סדרות זמן.

כדי לבצע חיזויים בעזרת מודלים סטטיסטים יש להתייחס למאפיינים שקיימים בנתונים כגון מחזוריות או טרנד אך לא חייבים להתייחס אליהם כאשר מפתחים מודל המבוסס על Neural Networks. זהו יתרון נוסף שיש לשיטה זו - לעיתים ישנם לסדרות זמן מאפיינים מורכבים.

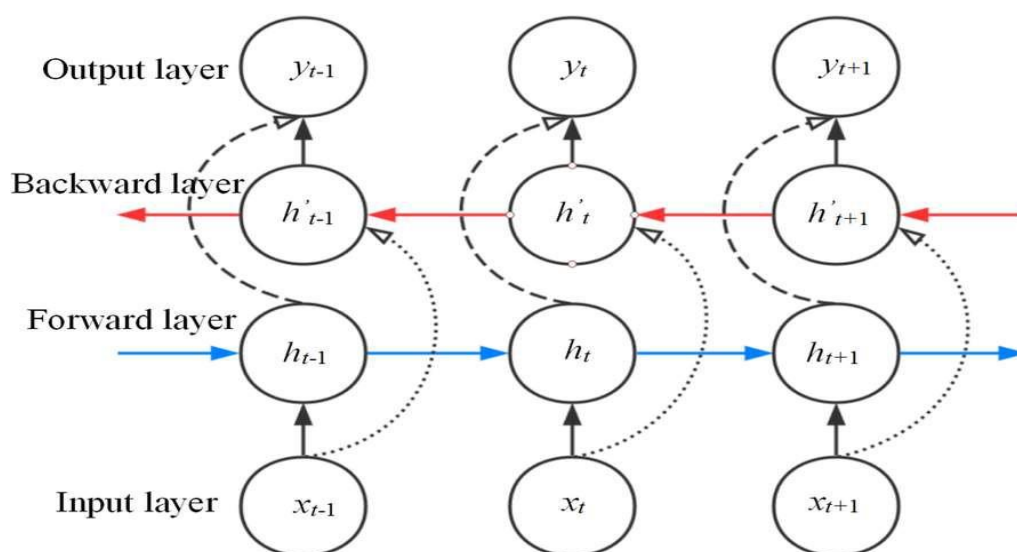
## 2. מבנה ארכיטקטורה

עבור חיזוי השימוש העתידי בחשמל, בחרתי להשתמש ברשת נוירונים מבוססת LSTM. בחרתי ליצור שני מודלים מסוג שונה ולהשוות בין הביצועים שלהם לחיזוי השימוש בחשמל. המודל הראשון הוא מודל חד-משתני. כלומר, מתבסס על סדרת זמן רבת משתנים (Multivariate) כקלט וחוצה את הערך הממוצע של כל משתנה בשעה הבאה. השני הוא מודל רב-משתני. כלומר, מתבסס על סדרת זמן עם משתנה יחיד (Univariate) כקלט וחוצה את הערך הממוצע של אותו משתנה בשעה הבאה. שני המודלים מורכבים מ-6 שכבות. שכבות ה-LSTM (Long Short Term Memory) הן השכבות שבהן מבוצעים החישובים העיקריים של תהליך החיזוי. LSTM הוא סוג של (Recurrent Neural Network) RNN. Recurrent Neural Networks הן רשתות נוירונים שנועדו "לפענח" מידע סדרתי - כלומר, להביא בחשבון את "סדר ההופעה" של התופעה בה מטפלים. רשתות אלו כוללות מרכיב של זיכרון - כך שהרשת "יודעת להתחשב" בעבר. "זיכרון הרשת" בא לידי ביטוי בכך כל פלט של שכבה ב-RNN משמש לחישוב הפלט של השכבה הבאה ובשונה מרשתות נוירונים אחרות, לכל השכבות ב-RNN יש את אותם המשקלים. בכל תא LSTM יש ארבעה שערים חישוביים שעוזרים לנו לשמור על זרימה קדימה טובה לאורך זמן ולנבא ערכים עתידיים של סדרת נתונים. לכל שער בתא ה-LSTM יש משקלים שבעזרתם מבוצע החישוב. מצב התא הוא בעצם הזיכרון לטווח ארוך של ה-LSTM. הוא בעצם מטריצה של משקלים שמייצגים סוג של זיכרון. השער הראשון - שער השיכחון: בשער זה מוחלט באילו נתונים מצב התא יש להשתמש יותר, ובאילו פחות. השער השני – שער הקלט: בשער זה מוחלט כיצד לעדכן את מצב התא לפי הקלט של התא והפלט של התא הקודם. השער השלישי – שער הפלט: בשער זה הפלט של התא מחושב בעזרת מצב התא, הפלט של התא הקודם והקלט של התא החדש.

בשכבת ה-LSTM כברירת מחדל ב-keras פונקציות ה-activation המשומשות בתא LSTM הן:  $\tanh$  ו- $\text{sigmoid}$ .

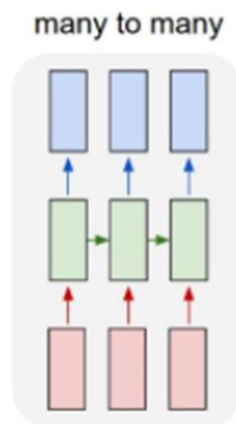


כל שכבת LSTM במודלים מוגדרת כ-Bidirectional. המשמעות של Bidirectional היא שבשכבה, יהיו 64 רשתות LSTM שבהם הנתונים יעברו בfeed forwards בסדר שבהם הם נקלטים, ו-64 רשתות LSTM שבהם הנתונים יועברו ב-feed forward בסדר הפוך.

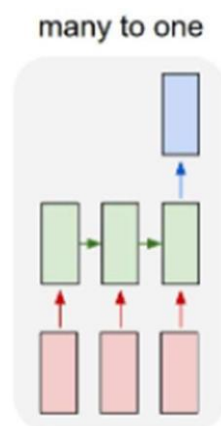


לדוגמה: עבור הקלט: [1,2,3,4,5] 64 רשתות LSTM יתחילו לעבד את הסדרה מהספרה 1 עד לספרה 5, ו-64 רשתות יתחילו לעבד את הסדרה מהספרה 5 עד לספרה 1.

שכבת ה-LSTM הראשונה עובדת בשיטה many to many – בכל LSTM עבור כל קלט ישנו פלט המועבר לשכבה הבאה.



שכבת ה-LSTM השנייה עובדת בשיטה many to one - ישנו רק הפלט אחד בכל LSTM unit מועבר לשכבה הבאה.



מספר המשקלים שבהם משתמשים המודלים: 136,275

שכבות ה-Dropout פעלו רק במהלך אימון המודל כדי למנוע overfitting, ובמהלך החיזוי אינן מופעלות, כברירת מחדל של keras.

שכבת ה-Dense הראשונה שבה 20 נירונים מהווה שכבת חישוב נוספת והיא מנצלת את Relu - פונקציית activation שימושית ויעילה לאימון מודלים של neural networks.

שכבת ה-Dense האחרונה מחשבת הפלט של המודל. במודל החד-משתני הפלט מחושב בשכבה זו מספר יחיד, ובמודל הרב-משתני הפלט המחושב הוא שלושה מספרים, בהתאם למספר המשתנים.

פונקציות ה-activation המושמשות בשכבות ה-Dense היא linear או במילים אחרות, אין פונקציית activation, כלומר הפלט הוא סכום מכפלת המשקלים בקלט לנירונים + ה-bias.

בחרתי לעשות resample לסדרת הזמן כך שמרווח הזמן בין כל תצפית הוא שעה, ובכל שעה, התצפית היא ממוצע הערכים של כל הדקות באותה שעה. זאת מכיוון שקיימים כשני מיליון דקות של מדידות עבור כל משתנה ב-dataset ואימון המודל בכמות כזו גדולה של נתונים הייתה לוקחת זמן רב מדי.

לאחר שינוי מרווח הזמן לשעות, כמות הנתונים מספיק גדולה כדי לאמן את המודל בצורה טובה, וניתן להגיד שחיזוי כמות החשמל בשעות הקרובות שימושי יותר מאשר החיזוי של הדקות הקרובות.

שינוי במרווחי הזמן של סדרת זמן עשוי לשנות את התלות בין הנתונים. במקרה זה התלות בין המשתנים אינה משתנה משמעותית, ניתן לראות זאת באמצעות ה-heatmap שמראה את גודל התלות בין המשתנים. (ראה נספח 1)

למרות שתלות בין משתנים לא בהכרח מראה על קשר ביניהם, בגלל שכל המדידות בוצעו באותו בית,



### המודל הרב-משתני:

Model: "sequential"

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	(None, 12, 128)	34816
dropout (Dropout)	(None, 12, 128)	0
bidirectional_1 (Bidirectional)	(None, 128)	98816
dropout_1 (Dropout)	(None, 128)	0
dense (Dense)	(None, 20)	2580
dropout_2 (Dropout)	(None, 20)	0
dense_1 (Dense)	(None, 3)	63
Total params: 136,275		
Trainable params: 136,275		
Non-trainable params: 0		

### המודל החד-משתני:

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
bidirectional_2 (Bidirectional)	(None, 36, 128)	34816
dropout_3 (Dropout)	(None, 36, 128)	0
bidirectional_3 (Bidirectional)	(None, 128)	98816
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 20)	2580
dropout_5 (Dropout)	(None, 20)	0
dense_3 (Dense)	(None, 3)	63
Total params: 136,275		
Trainable params: 136,275		
Non-trainable params: 0		

### 3. מדריך למשתמש

## הוראות התקנה

1. יש להוריד python 3.6.12(64 bit) מהאתר - <https://www.python.org/downloads/>

או לחילופין בעזרת anaconda  
יש לפתוח anaconda prompt ובסביבה הוירטואלית שבה רוצים להריץ את הפרויקט, יש להשתמש בפקודה:

```
conda install python=3.6.12
```

בהמשך ישנם הסברים כיצד לפתוח סביבה וירטואלית ולהפעיל אותה.

אם קיים python על המחשב, ניתן לבדוק את גרסתו בעזרת הפקודה: `python --version`  
ב- command prompt או ב-anaconda prompt.

```
(Tf) C:\Users\Niv Lifshitz>python --version  
Python 3.6.12 :: Anaconda, Inc.
```

2. יש לפתוח סביבה וירטואלית (virtual environment) ניתן לפתוח סביבה וירטואלית

ניתן לפתוח סביבה בעזרת Anaconda: <https://docs.anaconda.com/anaconda/install/>

או בעזרת הספרייה virtualenv.

כדי לפתוח סביבה וירטואלית חדשה בעזרת אנקודה, יש לפתוח את anaconda prompt ולהריץ את ה-command:

```
conda create --name envpath
```

כאשר envpath הוא path לסביבה הוירטואלית כולל השם שלה. במידה ולא מציינים path מלא, הסביבה תיפתח בתיקייה המקומית שבה מריצים את ה-command.

לחילופין ניתן לפתוח את הסביבה הוירטואלית דרך Anaconda Navigator.

כדי להפעיל את הסביבה הוירטואלית, יש להריץ את הפקודה:

```
activate newenv
```

ללא Anaconda, ניתן לפתוח סביבה וירטואלית באמצעות הפעולות הבאות:

לפתוח את ה-command prompt

לבדוק אם יש קוק בגרסת החסות python שמותקנת על המחשב. אם רואים מידע על ה commands בעזרת ה command: pip -h, זה אומר שהוא מותקן.

```
C:\Users\Niv Lifshitz>pip -h
Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  config            Manage local and global configuration.
  search            Search PyPI for packages.
  wheel             Build wheels from your requirements.
  hash              Compute hashes of package archives.
  completion        A helper command used for command completion.
  debug             Show information useful for debugging.
  help              Show help for commands.

General Options:
  -h, --help            Show help.
  --isolated            Run pip in an isolated mode, ignoring environment variables and user configuration.
  -v, --verbose         Give more output. Option is additive, and can be used up to 3 times.
  -V, --version         Show version and exit.
  -q, --quiet           Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).
  --log <path>         Path to a verbose appending log.
  --proxy <proxy>       Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries>   Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>       Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)ppend.
  --trusted-host <hostname> Mark this host as trusted, even though it does not have valid or any HTTPS.
  --cert <path>         Path to alternate CA bundle.
  --client-cert <path>  Path to SSL client certificate, a single file containing the private key and the certificate in PEM format.
  --cache-dir <dir>     Store the cache data in <dir>.
  --no-cache-dir        Disable the cache.
  --disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with --no-index.
  --no-color            Suppress colored output
```

להוריד את ה package: virtualenv בעזרת ה-command: pip install virtualenv

```
C:\Users\Niv Lifshitz>pip install virtualenv
```

לפתוח סביב וירטואלית בעזרת ה-command: virtualenv

בדוגמה זו, אני פותח סביבה וירטואלית בתיקה myenvironment שנמצאת ב-path:

C:\Users\Niv Lifshitz\

```
C:\Users\Niv Lifshitz>virtualenv myenvironment
```

כדי להפעיל את הסביבה הוירטואלית:

ב-Mac OS \ Linux:

להריץ את ה-command:

source myenvironment/bin/activate

ב-Windows:

```
C:\Users\Niv Lifshitz>myenvironment/bin/activate_
```

הערה: שים לב שתוכל להריץ את פקודות אלו רק אם התיקיה myenvironment היא local ב-path שממנו אתה מריץ את התוכנית, אחרת תצטרך לעבור אל ה-path בו התיקיה היא local או לספק path מלא ב-command.

-כדי להפסיק את השימוש בסביבה הוירטואלית:

יש להריץ את הפקודה:

Deactivate

3. יש להתקין את הספריות הבאות בסביבה הוירטואלית דרך Anaconda או בעזרת pip בסביבה הוירטואלית שיצרת:

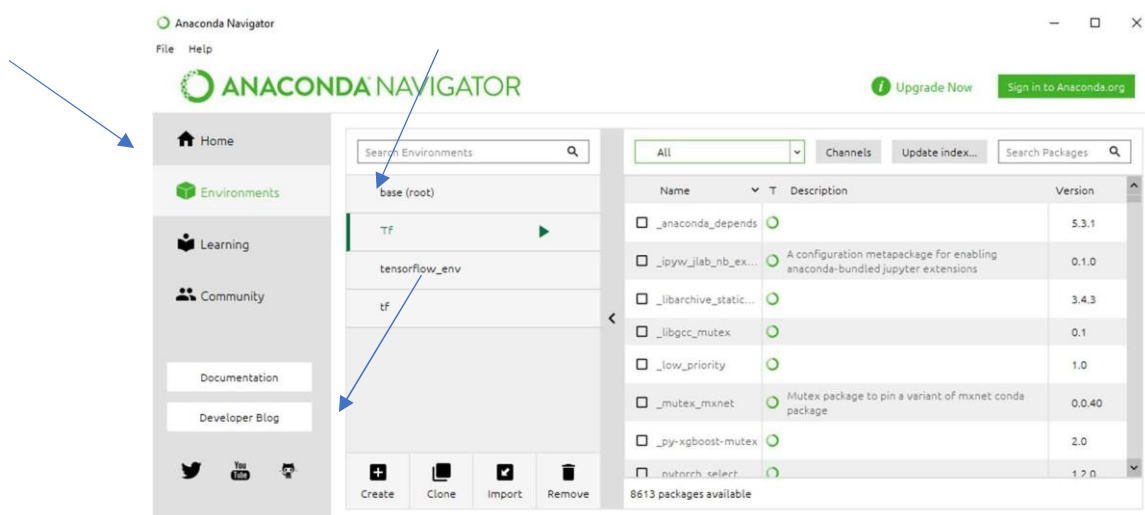
שם הספרייה	פקודת התקנה בעזרת pip	שם הספרייה ב- Anaconda	גרסה
tensorflow	pip install tensorflow	tensorflow	2.1.0
matplotlib	pip install matplotlib	matplotlib	3.3.4
numpy	pip install numpy	numpy	1.19.2
pandas	pip install pandas	pandas	1.1.5
scikit-learn	pip install -U scikit-learn	scikit-learn	0.24.1
statsmodels	pip install statsmodels	statsmodels	0.12.2
tkinter	pip install tkinter	tk	8.6
SciPy	pip install scipy	scipy	1.5.2
Pillow	pip install Pillow	pillow	8.2.0
seaborn	pip install seaborn	seaborn	0.11.1

**הערה:** אם אחת מהספריות כבר מותקנת בסביבה הוירטואלית שלך, יכול להיות שכאשר תנסה להוריד אותה לגרסה ישנה יותר, זה לא יעבוד לך. זהו באג באנקונדה שהפתרון אליו הוא למחוק את אותה ספרייה ולהוריד מחדש את הגרסה שאותה אתה רוצה להוריד.

-כדי להוריד את הספריות בסביבה הוירטואלית שיצרת בעזרת Anaconda, יש שתי דרכים:

(1) דרך Anaconda navigator:

לפתוח Anaconda Navigator, ללחוץ על Environments, ללחוץ על הסביבה שבה רוצים להוריד את הספריות, או ליצור סביבה חדשה בעזרת לחיצה על הכפתור Create.



בתמונה זו, לחצתי על הסביבה הוירטואלית ששמה הוא Tf.

לאחר מכן, יש להוריד את כל הספריות הדרושות. מדריך לניהול הספריות בסביבה הוירטואלית עם anaconda navigator:

[/https://docs.anaconda.com/anaconda/navigator/tutorials/manage-packages](https://docs.anaconda.com/anaconda/navigator/tutorials/manage-packages)

(2) דרך anaconda prompt, להפעיל את הסביבה הוירטואלית שבה אתה רוצה להוריד את הספריות, ולאחר מכן להשתמש בפקודה:

`conda install PACKAGE=VERSION`

כאשר במקום המילה PACKAGE יש לרשום את שם הספרייה, ובמקום VERSION את מספר הגרסה שרוצים להוריד.

```
(base) C:\Users\Niv Lifshitz>conda install tensorflow=2.1.0
```

בתמונה לעיל, פקודה להורדת הספרייה tensorflow בסביבה הוירטואלית base.

-כדי להוריד את הספריות בסביבה הוירטואלית שיצרת לא דרך Anaconda:

(1) יש להפעיל את הסביבה הוירטואלית כפי שכתוב בעמוד 5

(2) יש להשתמש בפקודה:

```
pip install PACKAGE==VERSION
```

כאשר במקום PACKAGE יש לרשום את שם הספרייה ובמקום VERSION יש לרשום את הגרסה של הספרייה.

4. יש להוריד את קובץ ה-zip של הפרויקט מחשבון ה-github שלי (ראה נספח 2-הורדת קובץ zip של קבצי הפרויקט מ-github), מהמאגר: Forecasting-household-electricity  
קישור למאגר: <https://github.com/5agie/Forecasting-household-electricity>

5. יש לחלץ את קבצי הפרויקט מקובץ ה-zip שהורד.

6. יש לחלץ לתיקיה שאליה חילצת את קבצי הפרויקט את קובץ הנתונים מקובצי ה-rar בשם:  
images.rar  
household\_power\_consumption.rar  
Weights.rar

## הרצת התוכנית

יש להריץ את הקובץ ProjectApplication.py דרך anaconda prompt או דרך ה-command prompt בתוך הסביבה הוירטואלית(כלומר לאחר הפעלתה) שבה הורדו הספריות הדרושות.

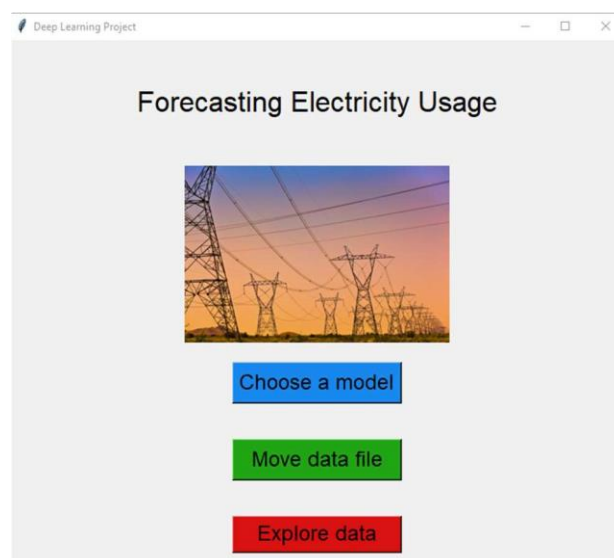
שים לב שתוכל להריץ את הקובץ בעזרת הפקודה:

```
python ProjectApplication.py
```

רק אם תריץ את פקודה זו מה-path שבו הקובץ נמצא. אחרת, תצטרך לספק את ה-path המלא לקובץ.

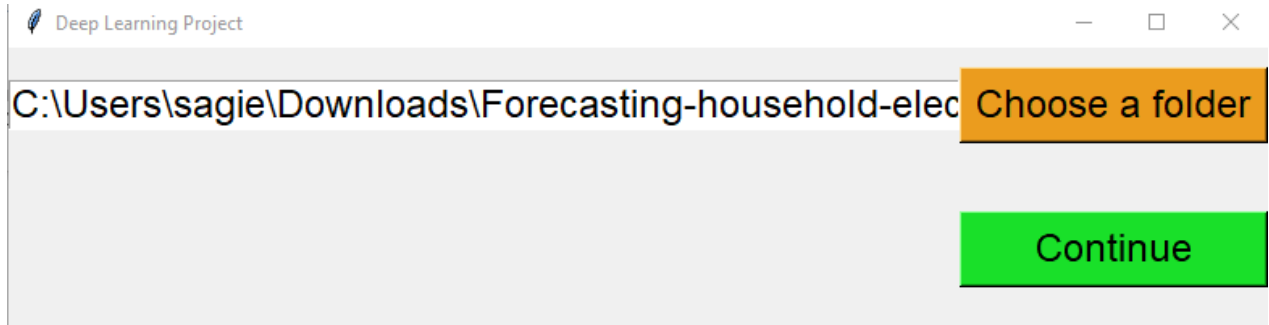
```
(Tf) C:\Users\Niv Lifshitz\Desktop\School 2020-2021\DeepLearning\Timeseries Project>python ProjectApplication.py_
```

בתמונה זו אני מריץ את הקובץ דרך anaconda prompt בסביבה הוירטואלית Tf. ניתן לראות את הסביבה הוירטואלית שבה משתמשים בסוגריים בצד שמאל.

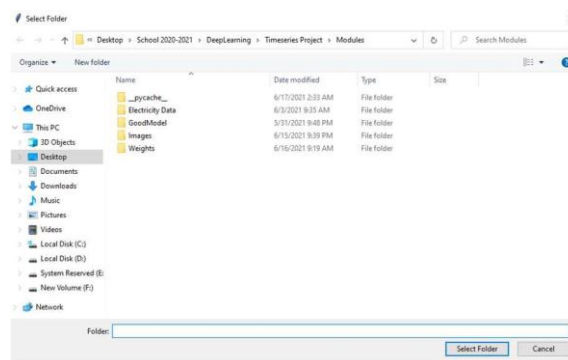


לאחר הרצת הקובץ, ייפתח התפריט הראשי, שבו יש שלושה כפתורים.

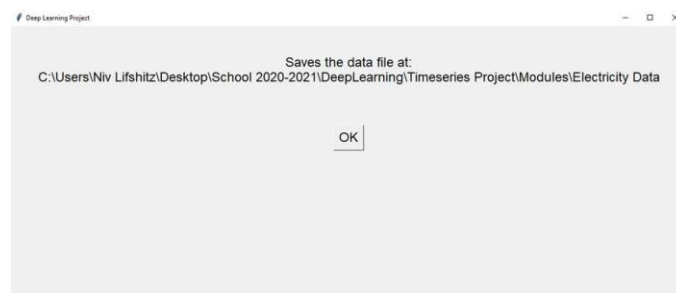
לחיצה על הכפתור שעליו רשום "move data file" תאפשר שינוי של המיקום בו נמצא קובץ הנתונים אשר שוקל 128 MB. לאחר לחיצה על כפתור זה, ייפתח חלון שבו מופיע ה-path שבו נמצא קובץ הנתונים כרגע.



ניתן לבחור את ה-path הרצוי לאחר לחיצה על הכפתור שעליו רשום "Choose a folder" ואחר כך לבחור את התיקייה שבה רוצים לשמור את הנתונים. לחילופין, ניתן לרשום את ה-path הרצוי ידנית בתיבת הטקסט.



לאחר לחיצה על כפתור "Continue", ייפתח חלון שיראה את ה-path שבו נשמר קובץ הנתונים.



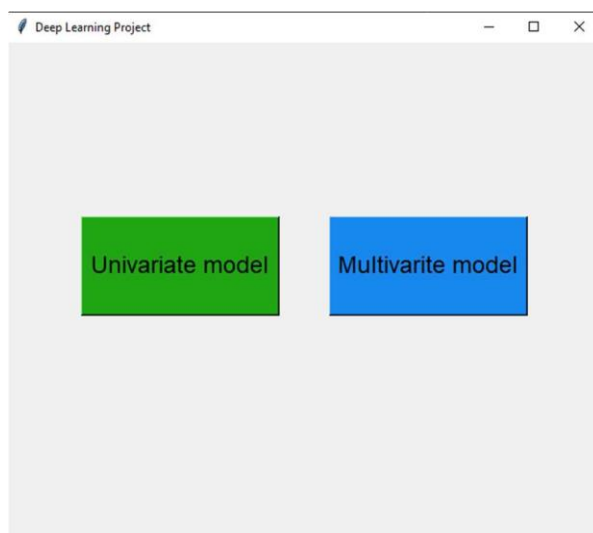
במידה וה-path השלם ארוך מכדי לראות את כולו בחלון שנפתח, אתה יכול להגדיל את החלון כדי לראות את ה-path המלא.



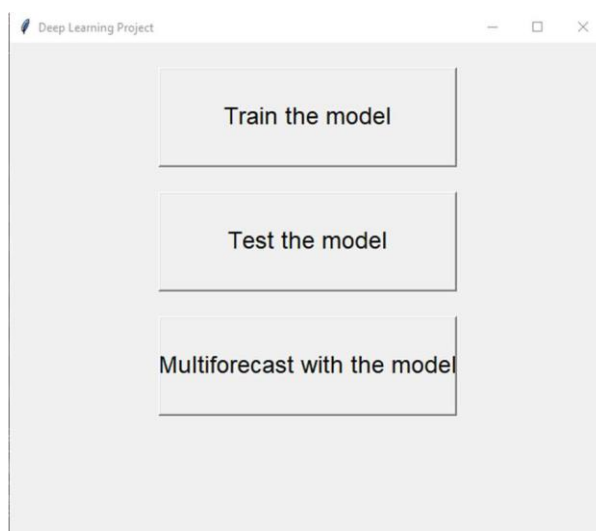
**שים לב:** במידה ושינית את מיקום קובץ הנתונים, תצטרך לעדכן את המשתנה הגלובלי DATA\_PATH שנמצא לאחר כל ה-imports בקובץ ProjectApplication (ראה נספח 3 – מיקום המשתנה הגלובלי DATA\_PATH)

לחיצה על הכפתור שעליו רשום "Choose a model" תאפשר בחירה במודל שאותו תרצו לאמן או לבחון.

יפתח חלון ובו 2 כפתורים, "univariate" – למודל החד-משתני ו-"multivariate" – למודל הרב-משתני.

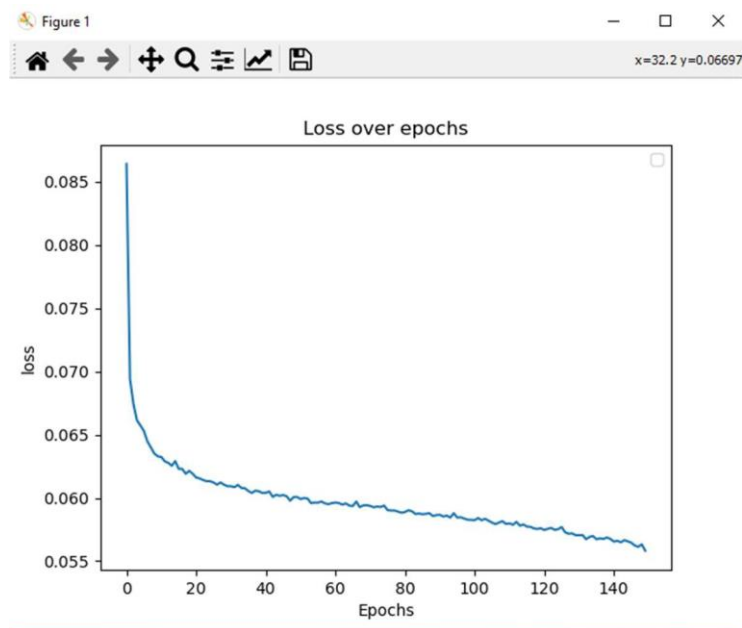
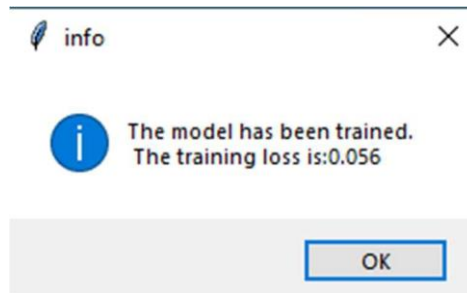


לאחר בחירת אחד המודלים, ייפתח חלון ובו תפריט של הפעולות האפשריות.

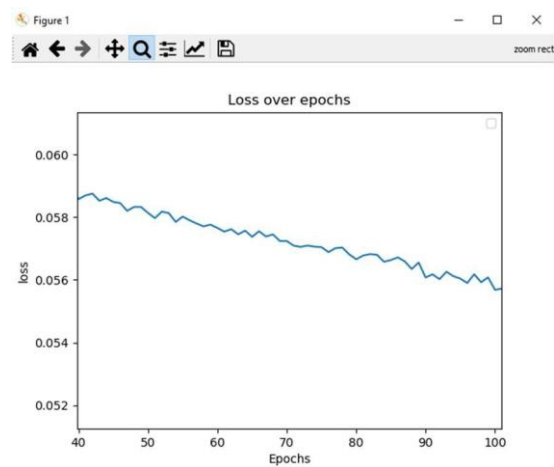
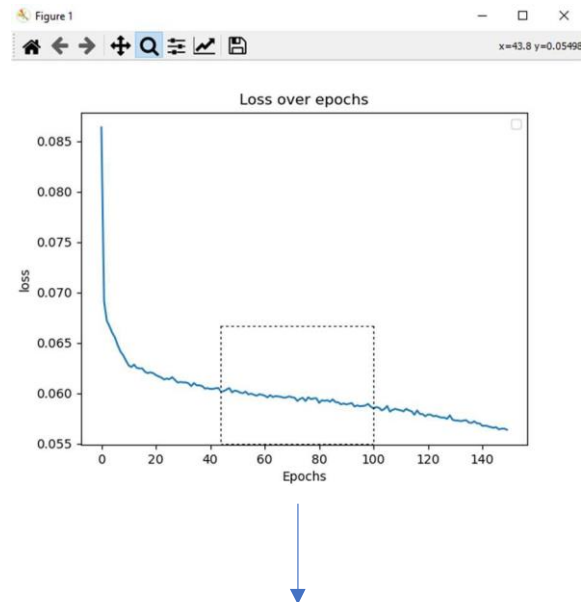


:Univariate model

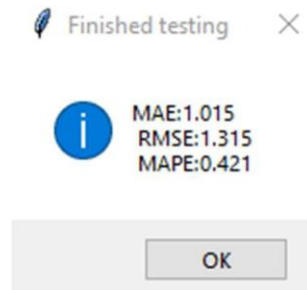
לחיצה על הכפתור שעליו רשום "Train the model" תתחיל את תהליך אימון המודל שהוגדר מראש ולאחר מכן תציג את ערך ה- loss הסופי של המודל, וגרף של ערך ה- loss כתלות ב- epochs במהלך אימון המודל.



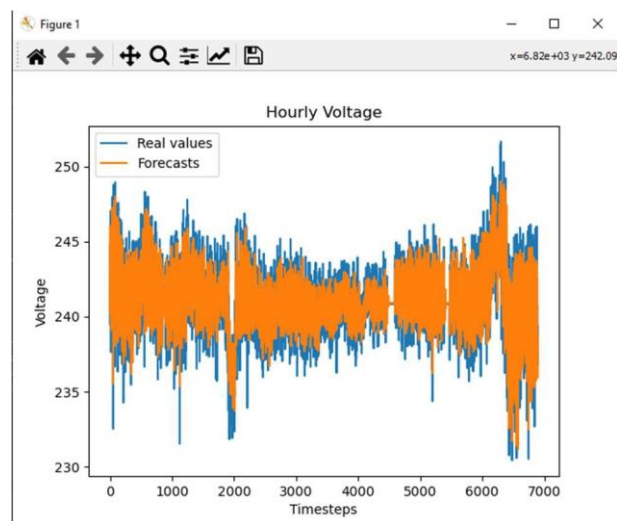
הערה: עבור כל חלון של מערכת צירים שייפתח, ניתן לראות חלקים קטנים יותר של הגרף בבירור, בעזרת לחיצה על הכפתור של הזכוכית מגדלת, ויצירת מלבן על החלק שאותו רוצים להגדיל.



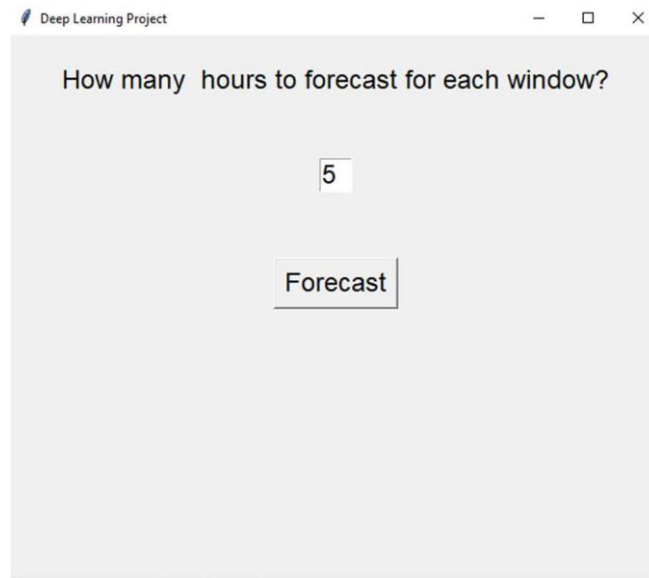
לחיצה על הכפתור שעליו רשום "Test the model" תתחיל את תהליך בחינת ביצועי המודל המוכן על נתונים שלא היו חלק מהנתונים שעליהם הוא התאמן ולאחר מכן יציג את ממוצעי שגיאות המודל, וגרפים של החיזויים והערכים האמיתיים במערכת צירים אחת.



לאחר לחיצה על הכפתור "OK" ייפתח חלון של מערכת צירים שבה יופיעה הגרף של ערך הלזס כתלות בepochs.



לחיצה על כפתור "multi-forecast", יפתח חלון שבו ניתן לבחור את אורך החיזוי הרקורסיבי שיבצע המודל, או במילים אחרות, מספר השעות בעתיד שהמודל יחזה את הנתונים בהם.



לחיצה על הכפתור שעליו רשום "Forecast" תתחיל תהליך חיזוי רקורסיבי בעזרת המודל. כלומר, המודל ישתמש בחיזוי שלו על מנת לחזות שוב, ותהליך זה חוזר על עצמו כמה פעמים, לפי מספר השעות שרוצים לחזות. המודל מבצע 800 חיזויים עבור 800 חלונות עוקבים באינדקס אקראי בנתוני ה-test.

תוכל לעקוב אחר תהליך החיזוי ב-command prompt לאחר התחלתו:

```
200 forecasts completed.  
300 forecasts completed.  
400 forecasts completed.  
500 forecasts completed.
```

ובמהלך חישוב השגיאות תוכל לראות את השגיאה המוחלטת הממוצעת:

```
100 windows errors calculated. Current MAE is:0.7776267203177598  
200 windows errors calculated. Current MAE is:0.8983413543782013  
300 windows errors calculated. Current MAE is:0.8856512669864877  
400 windows errors calculated. Current MAE is:0.8365301703933292  
500 windows errors calculated. Current MAE is:0.8043428981532954  
600 windows errors calculated. Current MAE is:0.8137924798621478  
700 windows errors calculated. Current MAE is:0.8150986165673589  
800 windows errors calculated. Current MAE is:0.8072716179297441
```

לאחר סיום החיזוי וחישוב השגיאות, ייפתח חלון קטן המראה את השגיאה המוחלטת הממוצעת של חיזוי המודל.

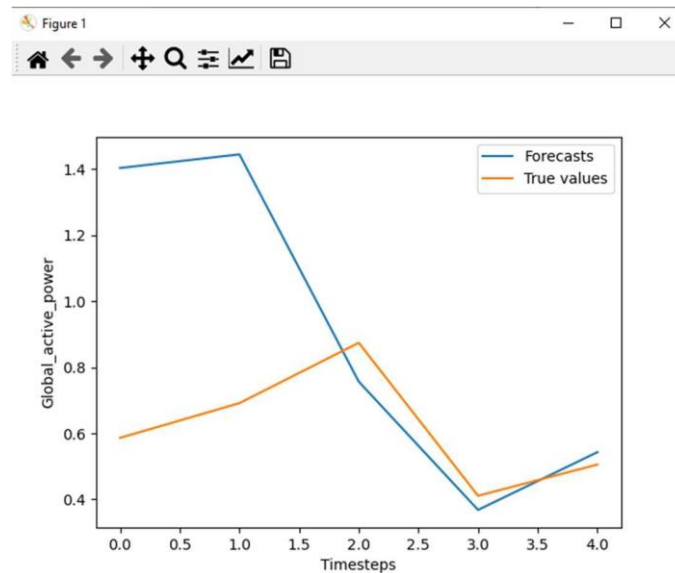
Finished forecasti... X



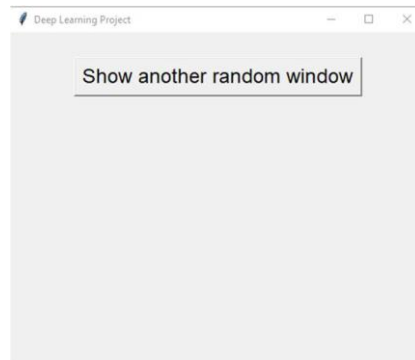
Error results:  
MAE:0.797

OK

לאחר לחיצה על הכפתור "OK" ייפתח חלון של מערכת צירים שבה יופיעו הגרפים של החיזויים של אחד מהחלונות, ואת הערכים האמיתיים.



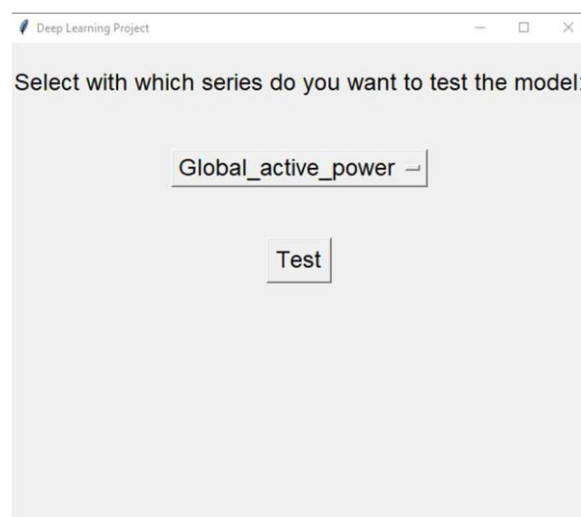
בנוסף ייפתח חלון שבו מופיע כפתור "Show another random window". לחיצה עליו לאחר סגירת החלון שבו מופיעים הגרפים, תפתח חלון נוסף של חיזויים וערכים נכונים של חלון אקראי מתוך החלונות שעליהם התבצע תהליך החיזוי

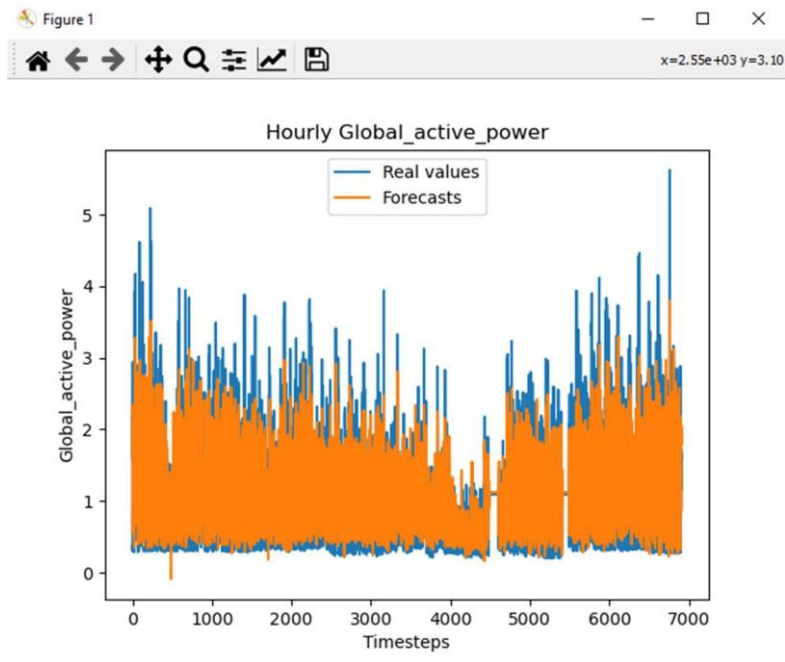


:Multivariate model

לחיצה על הכפתור שעליו רשום "Train the model" תתחיל את תהליך אימון המודל שהוגדר מראש ולאחר מכן תציג את שגיאות המודל על הנתונים שהוגדרו לבחינת הביצועים שלו, וגרף של ערך ה-loss כתלות ב-epoch, כמו במודל החד-משתני.

לחיצה על הכפתור שעליו רשום "Test the model" תפתח חלון שיאפשר בחירה של סדרת נתונים, מתוך סדרות הזמן שמשמשות לאימון המודל. השגיאה תחושב עבור סדרת הזמן שאותה בחרת. לאחר לחיצה על הכפתור שעליו רשום "Test" יתחיל את בחינת המודל ולאחר מכן יציג מערכת צירים שבה מוצגים הגרפים של חיזויי המודל והערכים האמיתיים.





לחיצה על כפתור "multi-forecast", יפתח חלון שבו ניתן לבחור את אורך החיזוי הרקורסיבי שיבצע המודל, או במילים אחרות, מספר השעות בעתיד שהמודל יחזה את הנתונים בהם ובחירה של המאפיין שאותו רוצים לחזות.

לחיצה על הכפתור שעליו רשום "Forecast" תתחיל תהליך חיזוי רקורסיבי בעזרת המודל. כפי שהוסבר בעמוד 21, ותוצאו החיזוי של סדרת הזמן שנבחרה יוצגו כמו שהן הוצגו במודל החד-משתני – ערך השגיאה הממוצעת וגרף של חלון אקראי מבין החלונות שעליהם בוצע החיזוי.



## 4. מדריך למפתח

הפרויקט שלי מחולק ל-modules אשר כל אחד מהם מכיל פעולות שימושיות לתחומים שונים. חלוקה זו מארגנת את השימוש בקוד, הופכת את הפרויקט לקל יותר להבנה, ומאפשרת לבצע שינויים לקוד בקלות.

**הערה:** ישנם פעולות ב-modules שאינן משמשות בממשק המשתמש אך השתמשתי בהן כדי לשפר את המודל, לבדוק את הביצועים שלו, או להבין את הנתונים.

קיימים הסברים נוספים באנגלית לפי תבנית reST בנוסף להסברים בעברית בעמודים הבאים, בתוך הקבצים.

שם הקובץ	תפקיד הקובץ
DatasetManager.py	מודול זה מכיל פעולות לארגון הנתונים.
TimeSeriesHelper.py	מודול זה מכיל פעולות המשמשות לחיזוי סדרות זמן.
DataVisualizer.py	מכיל פעולות שבעזרתן ניתן להציג את הנתונים בצורות שונות.
ProjectApplication.py	קובץ שאחראי על ממשק המשתמש. מכיל את הקוד לממשק המשתמש.

## DatasetManager.py הקובץ

הערה: DataFrame, DateOffset, Timedelta הכוונה לאובייקטים של מחלקות אלו בספרייה Pandas.

שם הפעולה	תפקיד הפעולה
<code>resample_data(df,rule='H')</code>	<p>הפעולה מקבלת אובייקט מסוג DataFrame מהספרייה pandas ומחרוזת או DateOffset או Timedelta המייצג את מרווח הזמן הרצוי.</p> <p>הפעולה מחזירה DataFrame חדש שבו נמצאים הנתונים שקיימים ב- DataFrame שהיא מקבלת, במרווח הזמן הרצוי.</p>
<code>get_sum_of_missing_values(df)</code>	<p>הפעולה מקבלת DataFrame.</p> <p>הפעולה מחזירה את כמות הערכים החסרים ב- DataFrame שהיא מקבלת.</p>
<code>scale_data(df,method='MinMax',in_place = False)</code>	<p>הפעולה עושה scale לערכים של ה- DataFrame שהיא מקבלת בשיטה שמצויינת ב- <code>method</code>.</p> <p>הפעולה מחזירה מילון של ערכים שבו כל שם של עמודה ב- DataFrame משמש כמפתח, וה- <code>scaler</code> שנוצר עבור אותה עמודה משמש כערך.</p> <p>הפעולה מחזירה DataFrame חדש או את אותו DataFrame שאותו קיבלה, לאחר ביצוע הפעולה, לפי הפרמטר הבוליאני.</p>
<code>rescale_data(data,scalers)</code>	<p>פעולה מקבלת data-2D numpy array ואת המילון <code>scalers</code> שהוחזר מהפעולה <code>scale_data</code>.</p> <p>הפעולה עושה <code>inverse_transform</code> לערכים ב- <code>data</code>. כלומר, מחזירה את הערכים לטווח הערכים שבו הם אמורים להיות.</p> <p>הפעולה מחזירה מילון שבו המפתחות הם המפתחות שהיו במילון ה- <code>scalers</code>, והערכים הם <code>numpy arrays</code> שבהם נמצאים הערכים בטווח המקורי שלהם.</p>

<p>הפעולה מקבלת DataFrame, מספר בין 0 ל-1, ובוליאן.</p> <p>הפעולה מחזירה tuple של שני DataFrames.</p> <p>אם הבוליאן to_numpy הוא True, הפעולה מחזירה tuple של שני numpy arrays.</p> <p>האלמנט הראשון ב-tuple הוא הנתונים לאימון, והאלמנט השני הוא הנתונים לבדיקת דיוק המודל (נתוני ה-test).</p> <p>החלוקה מבוצעת לפי הפרמטר for_train, שערכו הוא החלק מתוך כל הנתונים שרוצים שיהיה עבור אימון המודל.</p>	<p>split_to_train_test(df,for_train=0.8, to_numpy=False)</p>
<p>הפעולה מקבלת numpy array שאותו רוצים לפצל לחלונות, numpy array של הערכים שישמשו כערכים שהמודל אמור לחזות, וגודל החלונות.</p> <p>הפעולה יוצרת numpy array שמכיל חלונות בגודל window_size שמכילים את הערכים שנמצאים בdata ומערך numpy array שמכיל את הערכים שאותם המודל צריך לחזות.</p>	<p>split_to_windows(data,to_predict&gt;window_size=1)</p>

## הקובץ TimeSeriesHelper.py

חתימת הפעולה	תפקיד הפעולה
<code>single_multi_forecast(model,first_step,forecast_length=8, scaler = None)</code>	<p>הפעולה מקבלת מודל וחלון של נתונים ומבצעת חיזוי רקורסיבי מאותו חלון, <code>forecast_length</code> פעמים. במידה ונמצא הפרמטר <code>scaler</code> והחלון הוא של משתנה אחד, הפעולה עושה <code>inverse transform</code> לערכי החיזוי.</p> <p>עולה מחזירה מערך מסוג <code>numpy</code> שבו נמצאים החיזויים של המודל, לפי סדר החיזוי.</p>
<code>multi_forecast(model, data ,forecast_length=8)</code>	<p>הפעולה מקבלת מודל <code>Sequential</code> של <code>keras</code> ו-<code>numpy array</code> של חלונות של נתונים.</p> <p>הפעולה מבצעת חיזוי רקורסיבי של <code>forecast_length</code> <code>timesteps</code> מכל חלון ומחזירה רשימה של <code>numpy arrays</code> שבה כל <code>array</code> הוא החיזויים של החלון באינדקס התואם לו במערך <code>data</code>.</p>
<code>naive_forecast(series,typ='last_next',window_size = None)</code>	<p>הפעולה מקבלת <code>Series</code>, מחרוזת של סוג חיזוי ('<code>last_next</code>' או '<code>mean_average</code>'), וגודל חלון (אם סוג החיזוי הוא ממוצע נע).</p> <p>הפעולה מבצעת חיזוי של <code>timestep</code> הבא עבור כל <code>timestep</code> ב-<code>series</code> ומחזירה <code>pandas Series</code> שמכיל את החיזויים.</p>
<code>calculate_multi_forecast_errors(forecasts, trues, series_to_predict,forecast_length=8)</code>	<p>הפעולה מקבלת מילון של חיזויי הסדרות, לון של הערכים האמיתיים של הסדרות, שם הסדרה שלה רוצים לחשב את השגיאות, ואת אורך החיזוי הרקורסיבי שבוצע עבור כל חלון.</p> <p>הפעולה מחשבת את ממוצע השגיאה המוחלטת, ואת השורש של ממוצע השגיאה הריבועית, ומחזירה מילון שבו המפתחות הם סוגי השגיאות והערכים הם השגיאות (מספר).</p>

## הקובץ DataVisuallizer.py

חתימת הפעולה	תפקיד הפעולה
<code>plot (data,typ = 'line', title = "", xlabel = "", ylabel = "", legend = [])</code>	<p>הפעולה מקבלת רשימה של מספרים או numpy array, מחרוזת של הסוג של plot שאיתו רוצים להציג את ה-data, מחרוזת שתירשם ככותרת הגרף, מחרוזת שתירשם ליד ציר x, מחרוזת שתירשם ליד ציר y, ורשימה של מחרוזות של ייצוגי הגרף במידה וכמה גרפים מוצגים במערכת הצירים.</p> <p>הפעולה מציגה plot של ה-data לפי הסוג שצוין בפרמטר typ. הסוגים האפשריים הם 'line' ו-'histogram'.</p>
<code>plot_acorr_and_pacf(series,lag = 30)</code>	<p>הפעולה מקבלת רשימה של מספרים או numpy array חד מימדי, ומספר שלם.</p> <p>הפעולה מייצרת גרף של תלות של הסדרה עם סדרה מעוכבת שלה, בכל lag מ-1 עד lag.</p>
<code>plot_random_window(forecasts, trues, forecast_length=8)</code>	<p>הפעולה מקבלת numpy array חד מימדי שמכיל את החיזויים של החלונות כך שכל חלון במערך הוא (גודל החלון) אלמנטים במערך, numpy array חד מימדי שמכיל את הערכים האמיתיים בכל ה-timesteps, ומספר שלם – אורך החיזוי שבוצע עבור כל חלון.</p> <p>הפעולה מייצרת גרף של החיזויים והערכים האמיתיים של חלון רנדומלי באותה מערכת צירים.</p>
<code>multiple_line_plots(data,title="",xlabel="", ylabel = "", legend = [ ])</code>	<p>הפעולה מקבלת רשימה של numpy arrays או רשימות של מספרים, מחרוזת שתירשם ככותרת הגרף, מחרוזת שתירשם ליד ציר x, מחרוזת שתירשם ליד ציר y, ורשימה של מחרוזות בשביל הבחנה בין הגרפים.</p> <p>הפעולה מציגה את הגרפים של הערכים במערכת צירים אחת. כאשר כל גרף הוא רשימה או numpy array ברשימה data.</p>

## הקובץ Models.py

חתימת הפעולה	תפקיד הפעולה
<code>multivariate_model()</code>	הפעולה יוצרת את המודל הרב-משתני שהוגדר, ומחזירה את המודל הרב-משתני שהוגדר, מילון של ה-hyper parameters שהוגדרו בשביל אימון המודל, ואת גודל החלון שהוגדר עבור מודל זה.
<code>univariate_model()</code>	הפעולה יוצרת את המודל החד-משתני שהוגדר, ומחזירה את המודל, מילון של ה-hyper parameters שהוגדרו בשביל אימון המודל, ואת גודל החלון שהוגדר עבור מודל זה.

## הקובץ ProjectApplication.py

משתנה	תפקיד
DATA_PATH	מכיל מחרוזת של ה-path לקובץ הנתונים
root	מכיל את החלון המרכזי של התוכנית, הוא בעצם השורש של התוכנית.

שם הפעולה	תפקיד הפעולה
save_data_window()	הפעולה אחראית על ממשק המשתמש של חלון שינוי מיקום קובץ הנתונים.
path_note_window(path)	הפעולה מקבלת מחרוזת של ה-path שבו נשמר קובץ הנתונים ופותחת חלון שבו מופיע טקסט של path זה.
model_menu_window(parent)	הפעולה מקבלת את חלון(tkinter) שהממשק שלו קרא לפעולה. הפעולה אחראית על ממשק המשתמש של החלון של בחירת סוג המודל.
show_root_close_window(window)	הפעולה מקבלת חלון של tkinter. הפעולה סוגרת את חלון זה והופכת את החלון הראשי לנראה.
actions_menu_window(parent, model_type)	הפעולה מקבלת חלון של tkinter, מחזורת של סוג המודל("Uni" או "Multi"). הפעולה אחראית על ממשק המשתמש של החלון של תפריט הפעולות האפשריות עבור סוג המודל.
select_series_to_predict(parent,model_type)	הפעולה מקבלת חלון של tkinter, מחרוזת של סוג המודל("Uni" או "Multi"). הפעולה אחראית על ממשק המשתמש של החלון של בחירת סדרת הזמן שאותה רוצים לבחון (עבור המודל הרב-משתני).
select_multi_forecast_options(parent,model_type)	הפעולה מקבלת חלון של tkinter, מחרוזת של סוג המודל("Uni" או "Multi"). הפעולה אחראית על ממשק המשתמש של החלון של בחירת אורך החיזוי הרקורסיבי שיבוצע ובמידה וסוג המודל הוא רב-משתני("Multi"), בחירת סדרת הזמן שאת חיזויי המודל שלה רוצים לראות.

הפעולה אחראית על ממשק המשתמש של החלון של חקירת הנתונים.	data_window()
הפעולה מקבלת מחרוזת של path לתמונה, ופותחת חלון שמציג את התמונה.	display_image_window(img_path)
הפעולה מקבלת מחרוזת של שם העמודה בקובץ הנתונים. הפעולה פותחת חלון שבו histogram של נתוני עמודה זו, כאשר כל מרווחי הערכים ב-dataset הם שעה.	display_data_disturbution(column)
הפעולה מקבלת חלון של tkinter, מחרוזת של סוג המודל ("Uni" או "Multi"), ומחרוזת של שם אחד מהעמודות ששומשו עבור אימון המודל. הפעולה מבצעת אימון המודל ואחראית ממשק המשתמש לאחר סיום אימון המודל – מציגה גרף של ה-training loss וחלון קטן שבו רשום ה-loss האחרון של האימון.	train_model(parent,model_type, column_to_predict = 'Global_active_power')
הפעולה מקבלת חלון של tkinter, מחרוזת של סוג המודל ("Uni" או "Multi"), ומחרוזת של שם אחד מהעמודות ששומשו עבור אימון המודל. הפעולה בוחנת ביצועי המודל על נתוני ה-test. בנוסף הפעולה אחראית על ממשק המשתמש לאחר סיום בחינת המודל – הצגת ערכי שגיאות המודל והצגת גרף של החיזויים והערכים הנכונים.	test_model(parent,model_type, column_to_predict = 'Global_active_power')
הפעולה מקבלת מחרוזת של סוג המודל ("Uni" או "Multi"), מספר שלם המייצג את אורך החיזוי הרקורסיבי שיבוצע לאחר כל חלון, ומחרוזת של שם דרת הזמן שעליה רוצים לחשב את שגיאות החיזוי. הפעולה מבצעת חיזוי רקורסיבי בעזרת המודל על 500 חלונות אקראיים מנתוני ה-test ומחשבת את שגיאות המודל עבור סדרת הזמן שצויינה. לאחר מכן הפעולה אחראית על ממשק המשתמש לאחר סיום החיזוי – הצגת גרף של חיזוי של אחד מהחלונות באקראי ויצירת הכפתור לפתיחת גרפים נוספים.	multi_forecast (model_type, forecast_len , column_to_predict)
הפעולה סוגרת את התוכנית.	end_program()
הפעולה מקבלת אובייקט של Entry (של tkinter). הפעולה אחראית על ממשק המשתמש לבחירת התיקייה בה הוא רוצה לשמור את קובץ הנתונים ועדכון תיבת הטקסט.	select_dir(entry)



הפעולה מקבלת חלון של tkinter, מחרוזת של ה-path שבו המשתמש בחר לשמור את קובץ נתונים. הפעולה מעבירה את קובץ הנתונים ל-path אם הוא קיים. אחרת, פותחת חלון שמציג הודעה ש-path זה לא קיים.	save_data(parent,path)
הפעולה מקבלת חלון של tkinter, מחרוזת של סוג המודל ("Uni" או "Multi"), ומחרוזת של שם אחד מהעמודות ששומשו עבור אימון המודל. הפעולה מוודאת שלא הוכנס לתיבת הטקסט מספר שלילי או מחרוזת שאינה מספר.	validate_forecast_len(window,model_type, column_to_predict='Global_active_power' )

## 5. בסיס הנתונים

- הנתונים הם מדידות שימוש בחשמל בבית שממוקם בסו סיטי (עיר בארצות הברית). המדידות בוצעו מדצמבר 2006 עד נובמבר 2010 בכל דקה. בקובץ הטקסט שניתן לקרוא אותו כקובץ csv, ישנם 9 עמודות.
- עמודה 1 (date): התאריך שבו בוצעה המדידה בפורמט dd/mm/yyyy
- עמודה 2 (time): השעה שבה בוצעה המדידה בפורמט hh:mm:ss
- עמודה 3 (global\_active\_power): סך כל החשמל שנמצא בשימוש (נמדד בקילוואט)
- עמודה 4 (global\_reactive\_power): סך כל החשמל שלא נמצא בשימוש (נמדד בקילוואט)
- עמודה 5 (voltage): ממוצע המתח בבית (נמדד בוולט)
- עמודה 6 (global\_global\_intensity): ממוצע עוצמת הזרם (נמדד באמפר)
- עמודה 7 (sub\_metering\_1): מכשיר מדידה המחובר למטבח בבית (נמדד בוואט לשעה של פעולה)
- עמודה 8 (sub\_metering\_2): מכשיר מדידה המחובר לחדר הכביסה בבית (נמדד בוואט לשעה של פעולה)
- עמודה 9 (sub\_metering\_3): מכשיר מדידה מחובר לדוד ולמזגן בבית (נמדד בוואט לשעה של פעולה)

## 6. מסקנות הרצת המודלים

לאחר אימון המודל עם הנתונים בפעם הראשונה, ערך ה-loss היה גבוה מאוד ונתקע על ערך גבוה. לאחר חקירת הסיבות שעלולות לגרום לכך, הבנתי שזה קרה בעקבות הערכים השונים של הנתונים.

ל-Voltage יש ערכים בין 200-250 בערך. לעומת זאת ל-global active power יש ערכים בין 0.1 ל-6

בערך. הבדל זה גורם ל-Loss להיות מושפע בעיקר מה-Voltage כי ה-error שמחושב בין החיזוי של ערך ה-voltage יהיה גבוה הרבה יותר מה-error שמחושב בין החיזוי של ערך ה-global active power, ובעקבות כך, ה-loss יושפע בעיקר מה-error של ה-Voltage וגם עדכון המשקלים.

הבנתי שהפתרון לבעיה זו הוא לעשות scale לנתונים, כלומר לגרום להם להיות באותו טווח ערכים, כך שלכל feature תהיה השפעה שווה על ה-loss, ועדכון המשקלים יהיה מתון.

לעיתים שימוש בפחות נתונים ופחות features גורם למודל ללמוד טוב יותר ולביצועים טובים יותר. בעקבות זאת, הסתכלתי על התלות בין ה-features בdataset וניסיתי לאמן את המודל עם קומבינציות שונות של features.

בחרתי להוריד את נתוני ה-global reactive power מכיוון שראיתי שהקשר שלו ל-global active power קרוב ל-0. כלומר, שינוי בערך ה-global reactive power לא משפיע כמעט בכלל על השינוי ב-global active power ואכן לאחר הורדת נתונים אלו, חיזוי המודל היו טובים יותר.

מכיוון שביצועי המודל היו יותר טובים ללא השימוש במדידות ה-1 sub metering, 2 sub metering, החלטתי להוריד אותם. זאת למרות שהקשר שלהם ל-global active power יחסית גבוה.

חיזוי המודל עבור שעה בעתיד עם ערכי 3 sub metering היו טובים יותר מאשר החיזויים שלו בלעדיו, אך לא באופן משמעותי. הבחנתי בכך שהמודל חזה את ערך ה-3 sub metering כל הזמן קרוב ל-0 השגיאה בחיזוי feature זה הייתה גדולה. חיזויים אלו השפיעו לרעה על חיזוי של כמה שעות קדימה ולכן בחרתי לא לאמן את המודל עם feature זה.

ערך ה-Loss שאליו התייחסתי בעיקר לאחר ה-test של המודל הוא השגיאה המוחלטת הממוצעת, אך עדיין הוספתי ערכי השגיאות: השורש של ממוצע ריבוע השגיאה, ואחוז השגיאה המוחלטת הממוצע

ה-errors של global active power של המודל הרב משתני הסופי על נתוני ה-test:

Mean Absolute Error (שגיאה מוחלטת ממוצע): 0.326

Root Mean Squared Error (שורש של ממוצע ריבוע השגיאה): 0.505

Mean Absolute Percentage Error: 37.625

ה-errors של global active power של המודל החד משתני הסופי על נתוני ה-test:

Mean Absolute Error (שגיאה מוחלטת ממוצע): 0.315

Root Mean Squared Error (השורש של ממוצע ריבוע השגיאה): 0.493

Mean Absolute Percentage Error (אחוז השגיאה המוחלטת הממוצע): 34

**הערה לגבי אחוז השגיאה המוחלטת: אמנם אחוז הממוצע של השגיאה המוחלטת גבוה יחסית אך ניתן להבין את הסיבה לכך. Global active power מקבל ערכים נמוכים לרוב (ממוצע 1 בערך)**

ובעקבות כך, שגיאה מוחלטת של 0.1 כאשר הערך האמיתי הוא 0.2 היא שגיאה של 50%. אם שגיאה זו משמעותית או לא אני לא יודע להגיד, זה תלוי בצורך של המערכות. מישו שמבין בתחום של צריכה בחשמל או חיזוי השימוש בחשמל זה יכול להגיד עד כמה מדד שגיאה זו רלוונטי(ראה נספח 7 – ממוצע אחוז שגיאה מוחלטת).

## 7. דיון

בנושא זה יש המון מה לעשות ולחקור ואם היה לי את הידע והזמן, יש הרבה דברים שהייתי רוצה לעשות. שימוש באלגוריתם למציאת ה-hyper parameters הטובים ביותר, ולא לבחור אותם באופן ידני. השוואת ביצועי המודלים למודל שחזזה בכל פעם כמה שעות קדימה ולא שעה אחת בלבד. -אחלק את נתוני האימון ל-train and validation -יצירת מודלים לחיזוי שבועי או חודשי של השימוש בחשמל – לשנות את המרווח בין כל מדידה בנתונים לחודש או שבוע, ולאמן מודל בעזרת נתונים אלו. -בדיקת bias(נטיה) ו-variance(שונות) של המודלים השונים.

## 8. סיכום אישי ורפלקציה

העבודה על פרויקט זה בשבילי הייתה משימה מאתגרת ומהנה. היא דרשה את חקירת התחום לעומק, התמודדות עם בעיות וגרמה לי להבין את אופן החשיבה של מתכנתים העוסקים בתחום זה. מעשיית הפרויקט הפקתי כמה לקחים וקיבלתי כלים שיעזרו לי להמשיך כמו כיצד לפתח פרויקט גדול, קריאת מאמרים מדעיים, הבנת נתונים וכיצד להשתמש בהם ועוד.

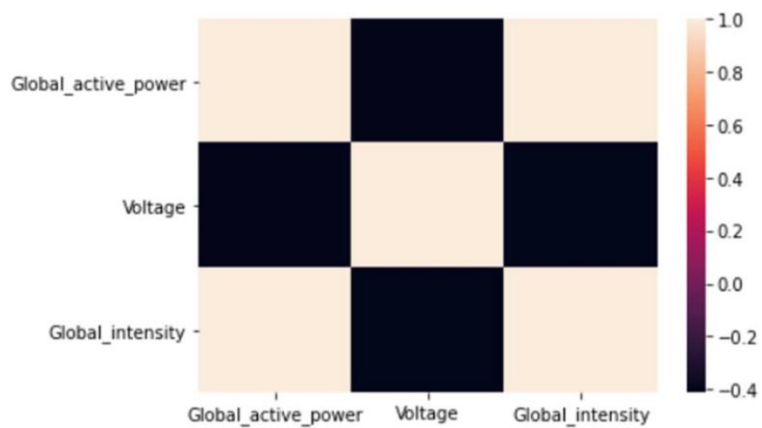
האתגרים המרכזיים שלי היו להבין מדוע המודל לא לומד, ומבצע חיזויים מוזרים. לאחר חקירת הסיבות השונות שיכולות לגרום לכך באינטרנט וניסיונות של פתרונות שונים, הצלחתי להתמודד עם אתגרים אלו. בנוסף לכך, במהלך שנת הלימודים אופן הלמידה היה מעניין ונחמד.

אחת המסקנות שלי היא שחשיבה יצירתית ויוזמה בשילוב עם חקר הנושא של הפרויקט הן המפתח לפיתוח נכון. בנוסף, הבנתי את החשיבות של תכנות גנרי, ויתרונותיו, לצד זה שהוא גורם לקוד להיות ברור ומובן.

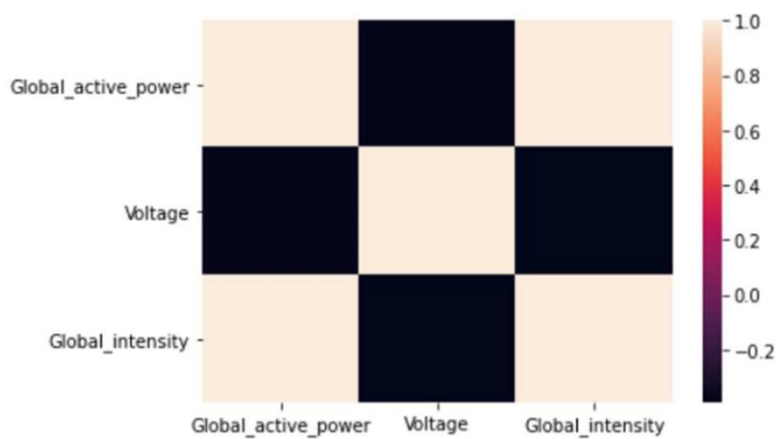
לו הייתי מתחיל היום את הפרויקט הייתי קורא מאמרים ומחקרים בנושא הפרויקט לפני התחלתו, זאת מכיוון שקיימים מאמרים רבים שמכילים מידע שעשוי לעזור מאוד לעשיית הפרויקט ולקצר תהליכים בעת עשייתו.

## 9. נספחים

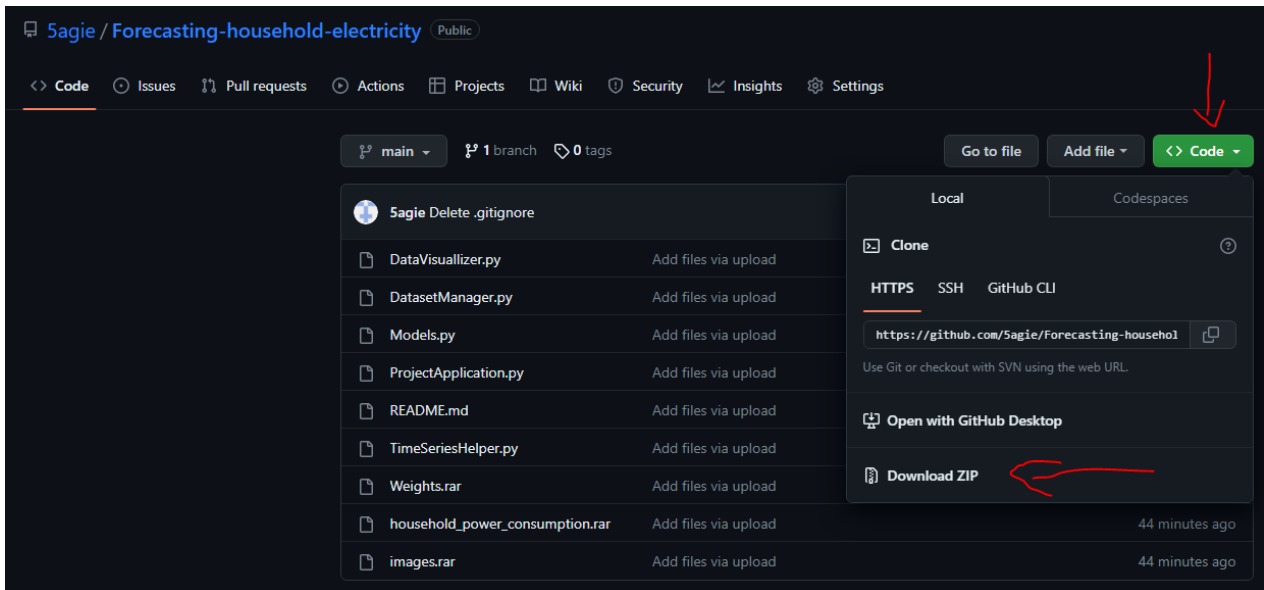
1. ה-heatmap כאשר מרווח הזמן הוא דקה:



ה-heatmap כאשר מרווח הזמן הוא שעה:



2. הורדת קובץ zip שמכיל את קבצי הפרויקט מ-github:



3. מיקום המשתנה הגלובלי DATA\_PATH:

```
7 from tkinter import
8 from tkinter import ttk
9 from PIL import ImageTk, Image
10 import tkinter.font as tkFont
11 import os
12 from tkinter import filedialog
13 import Models
14 import numpy as np
15 import DatasetManager6 as dm
16 import tensorflow as tf
17 import pandas as pd
18 import TimeSeriesHelper13 as tsh
19 import DataVisuallizer as dv
20 from shutil import move
21 import matplotlib
22
23
24
25 DATA_PATH = 'household_power_consumption.txt'
26
```

4. שונות – שונות (Variance) היא מדד לפיזור ערכים באוכלוסייה נתונה ביחס לתוחלת שלה. התוחלת (באנגלית: Expected value, ערך צפוי או Mean, מסומנת: E או  $\mu$ , בהתאמה) של משתנה מקרי היא ממוצע הערכים אותם צפוי המשתנה לקבל, משוקלל על-פי ההסתברויות לקבלת הערכים השונים.
5. שונות משותפת - השונות המשותפת (covariance) היא מדד לקשר בין שני משתנים מקריים. השונות המשותפת חיובית כאשר המשתנים נוטים לסטות מהממוצע באותו כיוון (שניהם מעליו או שניהם מתחתיו באותו מאורע), ושלילית כאשר הם משתנים בכיוונים מנוגדים זה לזה.



6. תלות (correlation) היא המדד המדורג/מותאם של שונות משותפת. ערכה בין 1- ל-1.

7. תשובות מעמיקות על חסרונות ממוצע אחוז השגיאה המוחלטת בפורום stackexchange (מבוססות על מקורות שונים):

<https://stats.stackexchange.com/questions/299712/what-are-the-shortcomings-of-the-mean-absolute-percentage-error-mape>

<https://stats.stackexchange.com/questions/280464/is-mape-a-good-error-measurement-statistic-and-what-alternatives-are-there>

## 10. ביבליוגרפיה

היו מקורות רבים באינטרנט שבהם השתמשתי, ושכחתי לשמור אותם. אלו אינם כל המקורות שבהם השתמשתי.

[1] Hong, T., Pinson, P., Wang Y., Weron W., Yang D., and Zareipour H. (2020). Energy Forecasting: A Review and Outlook. ResearchGate.

[2] Calado J.M.F., Cardiera C., and Rodrigues F. (2014). The Daily and Hourly Energy Consumption and Load Forecasting Using Artificial Neural Network Method: A Case Study Using a Set of 93 Households in Portugal. ResearchGate.

[3] Dolphin R. (2020). LSTM Networks | A Detailed Explanation. Towards data science  
<https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9>

[4] Somers K., Speelman E., Roelofsen O., and Witteveen M. (2020). Plugging in: What electrification can do for industry. McKinsey & Company.  
<https://www.mckinsey.com/industries/electric-power-and-natural-gas/our-insights/plugging-in-what-electrification-can-do-for-industry>

[5] Greenham D., Jacob M., and Neves C. (2019). Short Term Load Forecasting. Springer Link.  
[https://link.springer.com/chapter/10.1007/978-3-030-28669-9\\_2](https://link.springer.com/chapter/10.1007/978-3-030-28669-9_2)

[6] <https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/variance-standard-deviation-population/v/range-variance-and-standard-deviation-as-measures-of-dispersion>

[7] <https://www.kaggle.com/uciml/electric-power-consumption-data-set>

[8] Moroney L., Sequences, Time Series and Prediction. Coursera  
<https://www.coursera.org/learn/tensorflow-sequences-time-series-and-prediction#about>

מערכות אנרגיה מתחדשת. שחר אנרגיה בע"מ. [9]

<https://shaharenergy.co.il/system/renewable-energy-systems/>