הוראות שימוש בקוד לשחזור ניסויים:

1. משיכת מידע מהשירות המטאורולוגי יכול להיעשות ע״י מספר פונקציות, כאשר לכל אחת מהן ייעוד שונה מבחינת המידע שמתקבל ובפרט טווח התאריכים בו מעוניינים.

```
############### Retrieve data from the meteorological institute. ###############
# File named stations_information.json located in data/ will be created.
getStations(file_name='stations_information')
# File named _daily.json located in data/43/ will be created.
getStationDailyData(station=43)
# File named xmpl_monthly.json located in data/43/ will be created
getStationMonthlyData(station=43, file_name='xmpl')
# File named _2019_10_1.json located in data/43/ will be created
getStationDailyDataForDate(station=43, day=1, month=10, year=2019)
# File named _2019_10.json located in data/43/ will be created
getStationMonthlyDataForMonth(station=43, month=10, year=2019)
# File named _2019-9-20-2019-10-1.json located in data/43/ will be created
getStationRangeData(43, 2019, 9, 20, 2019, 10, 1)
```

בנוסף, אם רוצים מסיבה כלשהי למשוך מידע רק עבור פיצ׳ר ספציפי, ניתן לעשות זאת בצורה הבאה:

```
############### Retrieve data for specific channel from the meteorological
institute. ###############
ids = getChannelIds()
# File named _daily.json located in data/43/ will be created contains only  data
for TD channel.
getStationDailyData(station=43, channel=ids['TD'])
```

2. יצירת Dataframe שנבנה באמצעות הנתונים שנמשכו ב(1).
בנוסף, ייצוא הDataframe לקובץ לשם נוחות עבודה וגישה למידע שנבנה בו במקומות שונים במהלך הפרוייקט.

```
############### Data Frame Creation ###############
file_name = '2019-9-20-2019-10-1'
# Data frame will be created from data/43/_2019-9-20-2019-10-1.json
technion_final_test_dataframe = createDataFrame(file_name='{}/_{}.json'.format(43,
file_name))
# Export data frame to csv.
technion_final_test_dataframe.to_csv('./data/{}/dataset_{}.csv'.format(43,
file_name))
```

3. יצירת Dataframe שבנוי מאיחוד כל הפיצ׳רים של כלל התחנות.

```
############## All Stations Merged Data Frame Creation ###############
# File named merged_all_2019-7-1-2019-9-1.csv located in data/ will be created.
create_data_for_all_stations('2019-7-1-2019-9-1', 2019, 7, 1, 2019, 9, 1)
```

- נשים לב שעל מנת לבצע ניסויים הנוגעים רק לתחנה ספציפית נשתמש במידע שנבנה ב(2). לעומת זאת, לביצוע ניסויים המשלבים את כלל התחנות נשתמש בתוצאה של המידע הנבנה ב(3).

4. מציאת הפרמטרים הטובים ביותר עבור המסווגים השונים.
לשם ביצוע הדבר קיימות שתי אפשרויות:
   - הרצה של findFeaturesScript.py ב- Terminal.
   - קריאה לפונקציה find_features_runner (שאותה הסקריפט הנ״ל מריץ ברקע) באופן ישיר.

   ללא תלות בבחירת שיטת ההרצה, יש לבצע את השינויים הבאים בקובץ findFeaturesScript.py:

   4.1. תחת הפונקציה find_features_runner יש להחליף את המשתנה checked_station לתחנה הרלוונטית בבדיקה (בדוגמא הנ״ל 43 – תחנת הטכניון)

4.2. תחת המain יש לשנות את הקריאה לפונקציה find_features_runner בהתאם לפרמטרים הרלוונטיים כאשר:
raw_data_file_name – שם הקובץ (כולל הנתיב אליו) כפי שנוצר בשלב (2) או בשלב (3).
output_path_name – תיווצר תיקיית פלט תחת הנתיב - ./data/Experimanets/station_43_7-8_2019_all_stations_example/.
output_file_name – שם קובץ הפלט.
using_merged – במידה והקובץ שנשלח ל raw_data_file_name נוצר לפי (3), כלומר קובץ merged של כלל התחנות – ערך השדה הנ"ל צריך להיות True, אחרת False.

```
 ############## Finding Best parameters for each regressor  ##############
# Can also be run using terminal - just run fundFeaturesScript.py.
find_features_runner(raw_data_file_name="./data/merged_all_2019-7-1-2019-9-1.csv",
                     output_path_name='7-8_2019_all_stations_example',
                     output_file_name='regression_all_stations_dataset_7-
                     8_2019.csv',
                     using_merged=True)
)
```

5. לאחר שלב (4) מתקבל קובץ ובו כל התוצאות עבור הרצה של כל רגרסור, עבור האופציות השונות הניתנות בכל איטרציה לפרמטרים המשתנים – k עבור select_k_best, days עבור כמות הוספת ימים לסט הנתונים, וכדומה.
כעת נרצה לערוך קבצים חדשים הנקראים בשם best_features_regression, כאשר במקום regression נכתוב את שם הרגרסיה אותה נרצה לבדוק. את הקבצים נערוך בעזרת Excel כאשר בכל קובץ נשאיר את שורת המידע בה התקבלה התוצאה הטובה ביותר עבור הרגרסיה המתאימה כך שFeatures אינו ריק.
לבסוף נשמור כקובץ csv.
לדוגמא, הקובץ best_features_Ridge.csv יראה כך:

Days	Corr	K_best	Reg	Features	Mean absolute error
0	0	5	Ridge(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=None, normalize=False, random_state=None, solver='auto', tol=0.001)	[['TD_269', 'TDmin_269', 'TDmin_90', 'TDmax_90', 'TD_90']]	1.644945809

את הקבצים נשים בתיקייה בשם שנבחר, ונשתמש בהם בהמשך להרצת הסקריפט הסופי.

6. נריץ את הסקריפט mainScript.py. כדי להריץ אותו בהתאם לניסוי הנוכחי נערוך אותו בצורה הבאה:
6.1. בקובץ mainScript.py קיים מילון בשם feature_regressions_map. עבור כל מפתח (שם הרגרסיה אותה אנו בודקים), נחליף תחילה את הנתיבים ושמות הקבצים בהתאם למיקום בו נשמרו בשלב (5), ולאחר מכן ניקח את השדה Features מהקובץ המתאים לו משלב (5).
6.2. נגדיר את התחנה אותה אנו בודקים:

```
checked_station = 43
```

6.3. נגדיר את המיקום שבו נרצה שישמרו תוצאות ההרצה. נעשה זאת ע"י שינוי הפרמטרים path, path_result באופן הבא:

```
# Create path folders if needed
# Main path where data will be saved.
path = './data/Experiments/station_43_1_years_all_stations_submit_test'
```

```
# Results folder within the path location.
path_results = path + '/grid_search_resultes/'
```

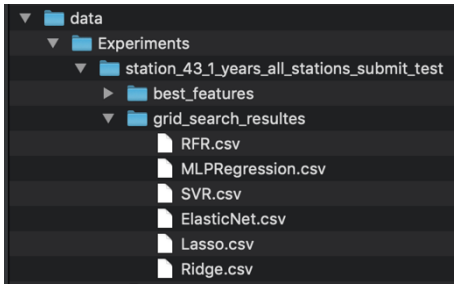
6.4. את קובץ סט המידע המתאים לניסוי זה אותו הכנו בשלב (2) או (3) (תלוי איזה סט מידע אנחנו בודקים בניסוי זה) נשמור במקום אותו נבחר, כאשר חשוב להשתמש באותו סט מידע כפי שהועבר בשלב (4.2).
לאחר מכן, נטען אותו על מנת להרחיבו בפיצ׳רים כמספר המקסימלי של הימים אשר נוספו במהלך הבדיקה בfindFeaturesScript.py. פעולה זו תעשה בשורה הבאה:

```
main_script_runner(raw_data_file_name='./data/merged_all_2019-7-1-2019-9-1.csv',
using_merged=True)
```

כאשר בדוגמא זו טענו קובץ התואם לשלב (3) עם הוספת ימים מקסימלית של 30 ימים, כפי שזו נבדקה בfindFeaturesScript.py.

6.5. לבסוף, מהטרמינל נריץ את – mainScript.py.

6.6. התוצאות ישמרו ב-path שנבחר והן ישמרו עבור כל רגרסיה, כלומר נקבל קובץ בשם הרגרסיה אותה בדקנו, המכיל את כל התוצאות האפשריות עבור כל פרמוטציה של הפרמטרים אותם חקרנו.
בסופו של התהליך היררכיית הפלט עבור הדוגמא שנתנו בשלב (6.2) תהיה כדלהלן:



6.7. התוצאה הטובה ביותר עבור כל רגרסיה תהיה התוצאה המוצגת בשלב הניסויים.