Regressziós modellek illesztése egy kombináltciklusú erőmű adataira

Szilágyi Gergő

May 21, 2024

Contents

1 Bevezetés

Ebben a dolgozatban a kombinált ciklusú erőművek (CCPP) adatainak különböző regressziós modellekkel történő elemzését mutatom be. Célom, hogy a később felsorolt környezeti változók alapján meg tudjam becsülni az erőmű által leadott elektromos teljesítményt. Ehhez a következő adatokat fogom felhasználni:

1. Leadott elektromos teljesítmény (PE)
2. Környezeti hőmérséklet (AT)
3. Atmoszférikus nyomás (AP)
4. Relatív páratartalom (RH)
5. Vákuum (V)

A dolgozat az alábbi cikk alapján készült:

Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods

A hivatkozott cikkel ellentétben jelen dolgozat nem foglalkozik a változók kiválasztásával, bár az elkészült kódbázis erre lehetőséget ad.
Ezt a témát egy későbbi dolgozat keretében fogom részletezni. Ebben a dolgozatban a következő lépéseket és eredményeket fogom bemutatni:

1. **Feltáró Adatelemzés:**
   Az adatok betöltése, vizsgálata, tisztítása és ábrázolása után azonosítom és eltávolítom az outliereket, majd egy heatmapet és egy pairplotot mutatok be.
2. **Adatok skálázása és normalizálása:**
   Látni fogjuk, hogy az adatok különböző nagyságrendűek, ami problémát jelenthet a regressziós modellek illesztésénél. Ennek kezelésére előállítom a skálázott és normalizált adatokat.
3. **Regressziós modellek előkészítése és illesztése:**
   Több különböző regressziós modellt fogok bemutatni és alkalmazni, melyeket először röviden ismertetek, majd csoportokba rendezve példányosítom őket. Az előkészített modelleket a tanulóhalmazra illesztem, és az illesztett modellek segítségével előállítom a predikciókat a teszthalmazon.
4. **Kiértékelés és értékelés:**
   A predikciókat összevetem a teszthalmazban található értékekkel, és az alábbi módszerek segítségével értékelem a modelleket:
   (a) MSE (Mean Squared Error)
   (b) R^2 a továbbiakban $R2$
   (c) CV keresztvalidáció
5. **Összegzés és diszkusszió**
   Végül kiválasztom a legjobbnak ítélt modellt rövid indoklással, összegzem a megoldásom, és kitekintek a további fejlesztési lehetőségekre.

Munkám során számos Python könyvtárat használtam az adatkezelés, -elemzés és modellezés megvalósításához. Az adatok betöltéséhez és manipulálásához a **pandas** és **numpy** könyvtárakat, a vizualizációk készítéséhez a **matplotlib** és **seaborn** könyvtárakat, míg a regressziós modellek illesztéséhez és értékeléséhez az **scikit-learn** (sklearn) könyvtárat használtam.
Az általam írt teljes kódbázis elérhető a GitHub profilomon, amely a következő linken található:

https://github.com/szilagyi93/regression.

2 Feltáró Adatelemzés

2.1 Beolvasás, hiányos adatok, duplikációk

Table 1: Head of Train dataset

	AT	V	AP	RH	PE
0	10.54	34.03	1018.71	74.00	478.77
1	7.08	39.99	1010.55	91.44	482.83
2	14.49	41.16	1000.50	82.17	465.24
3	10.73	25.36	1009.35	100.15	469.43
4	22.88	63.91	1009.63	87.82	442.50

Table 2: Head of Test dataset

	AT	V	AP	RH	PE
0	9.59	38.56	1017.01	60.10	481.30
1	12.04	42.34	1019.72	94.67	465.36
2	13.87	45.08	1024.42	81.69	465.48
3	13.72	54.30	1017.89	79.08	467.05
4	15.14	49.64	1023.78	75.00	463.58

Elős lépésben betöltöttem pandas frame-be a tanuló (pd_train) és teszt (pd_test) halmazokat. Miután meggyőződtem róla, hogy a beolvasott táblázatok alakja konzisztensek, tehát az oszlopok száma egyenlő, ellenőriztem az esetleges hiányosságokat.

Listing 1: Info about Train

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8568 entries, 0 to 8567
Data columns (total 5 columns):
 #   Column Non-Null Count Dtype
---  ------ -------------- -----
 0   AT     8568 non-null  float64
 1   V      8568 non-null  float64
 2   AP     8568 non-null  float64
 3   RH     8568 non-null  float64
 4   PE     8568 non-null  float64
dtypes: float64(5)
memory usage: 334.8 KB
```

Listing 2: Info about Test

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column Non-Null Count Dtype
---  ------ -------------- -----
 0   AT     1000 non-null  float64
 1   V      1000 non-null  float64
 2   AP     1000 non-null  float64
 3   RH     1000 non-null  float64
 4   PE     1000 non-null  float64
dtypes: float64(5)
memory usage: 39.2 KB
```

Mivel az adathalmazok nem volt hiányosak így a duplikációk keresésével és eliminálásával haladtam tovább. A tanuló halmaz 36 a teszt halmaz pedig 0 duplikációt tartalmazott. Mivel 8568 sorból áll a tanulóhalmaz, így úgy döntöttem, hogy törlöm a duplikált adatokat.

Listing 3: Duplication of Train data set

```
Duplications of TRAIN data:

Duplicates: (36, 5)
Original Shape of Data: (8568, 5)
No Duplicates Data: (8532, 5)
```

Listing 4: Duplication of Test data set

```
Duplications of TEST data:

Duplicates: (0, 5)
Original Shape of Data: (1000, 5)
No Duplicates Data: (1000, 5)
```

2.2 Vizualizáció

2.2.1 Hisztogramok, Boxplotok és outlierek

Annak érdekében, hogy teljesebb képet kapjak a különbőzű változók eloszlásáról ábrázoltam azokat, hisztogram formájában.

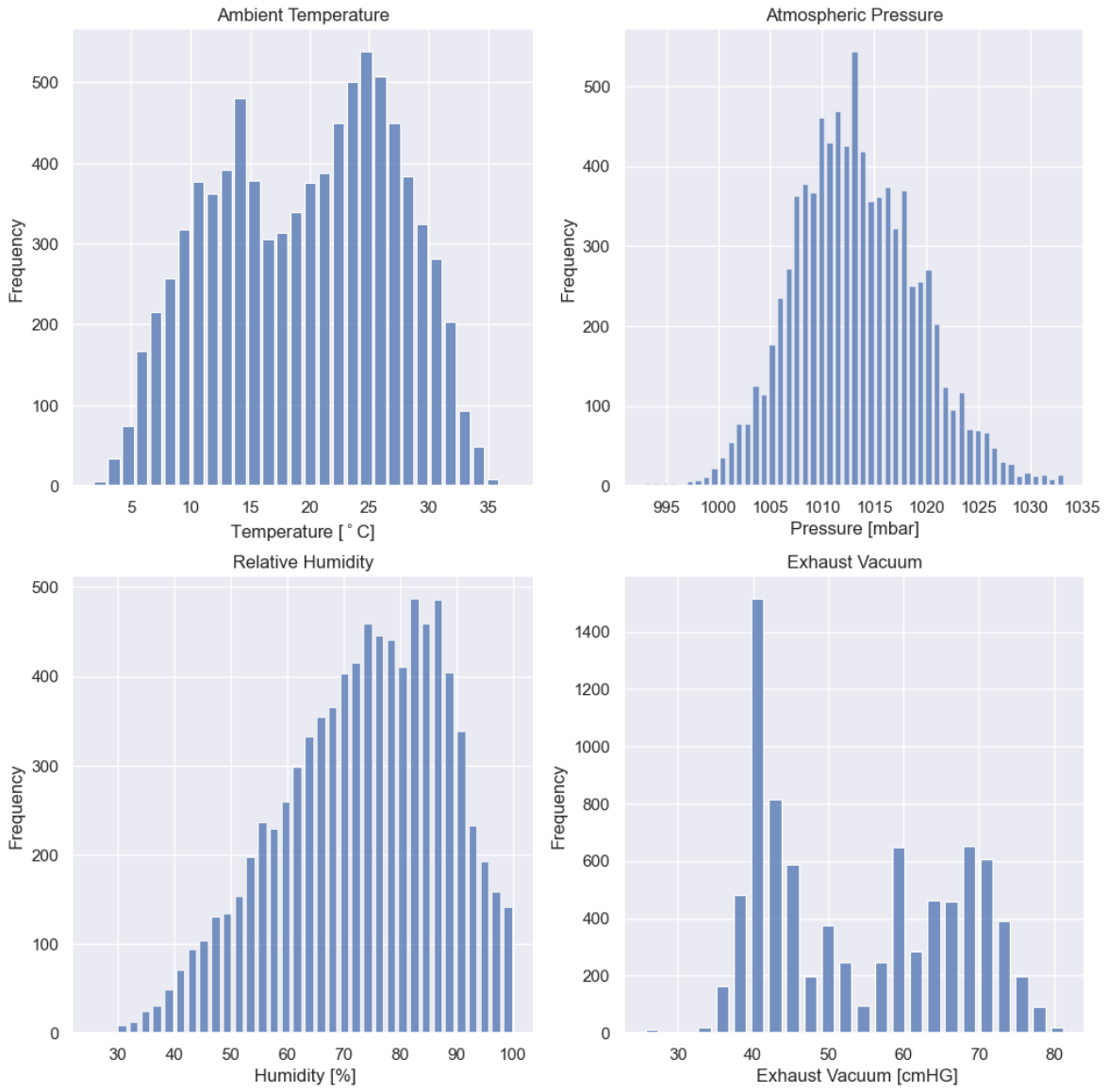

Figure 1: Histograms of Temperature, Pressure, Humidity, and Vacuum

Az AT, AP, RH, és V változók hisztogram ábárái azt mutatják, hogy az eloszlások többékevésbé normálisak, de az AT (környezeti hőmérséklet) és a V (vákuum) esetében két csúcs látható, ami bimodális eloszlásra utal.
Valamint az R (relatív páratartalom) esetén látható negatív skew, ami azt jelzei, hogy az adatok átlaga kisebb, mint a mediánja, ami kisebb mint a módusza.
Mindez jól látható az adatokat jellemző statisztikai momentumokat ábrázoló táblázatban.

Table 3: Statisztikai összefoglaló a különböző változókról

Stat	AT	V	AP	RH	PE
Count	1000.000	1000.000	1000.000	1000.000	1000.000
Mean	20.188	54.841	1013.106	72.501	453.176
Std	7.339	12.559	5.930	15.178	16.591
Min	3.210	34.030	996.350	26.300	425.300
25%	14.148	42.763	1008.935	61.620	438.835
50%	21.035	52.780	1013.005	74.395	450.415
75%	26.145	66.560	1017.043	84.610	466.388
Max	35.010	80.180	1033.300	100.140	494.870
Median	21.035	52.780	1013.005	74.395	450.415
Skew	-0.162	0.185	0.357	-0.417	0.352

Ezek után outliereket azonosításával haladtam tovább. Elsőként elkészítettem a boxplotokat.

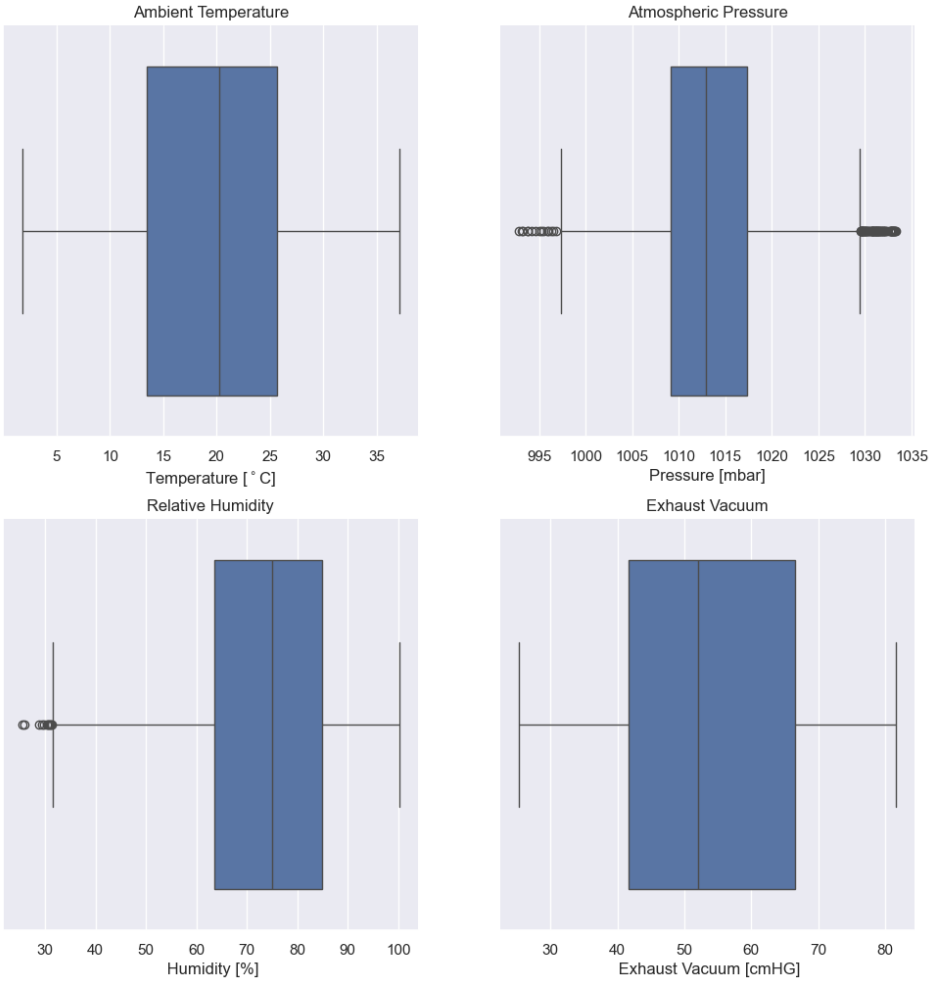

Figure 2: Visualizations of environmental variables

A következő kódrészlet jól mutatja, ami a boxplotokon is valamelyest látható, hogy összesen $92db$ outlier került azonosításra és eldobásra.

Listing 5: Duplication of Train data set

```
def remove_outliers(data):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    return data[(data >= lower_bound) &
        (data <= upper_bound)].dropna()
```

Listing 6: Duplication of Test data set

```
shape of original_train_shape
    (duplications removed):
(8532, 5)
shape of filtered_train_shape:
(8440, 5)
Number of rows has been removed: 92

Filtered data has been saved:
 02_data/filtered_train.xlsx
```

2.2.2 Pair plot és heat map

További elemzés céljából elkészítettem az adatokra vonatkozó pair plotot.

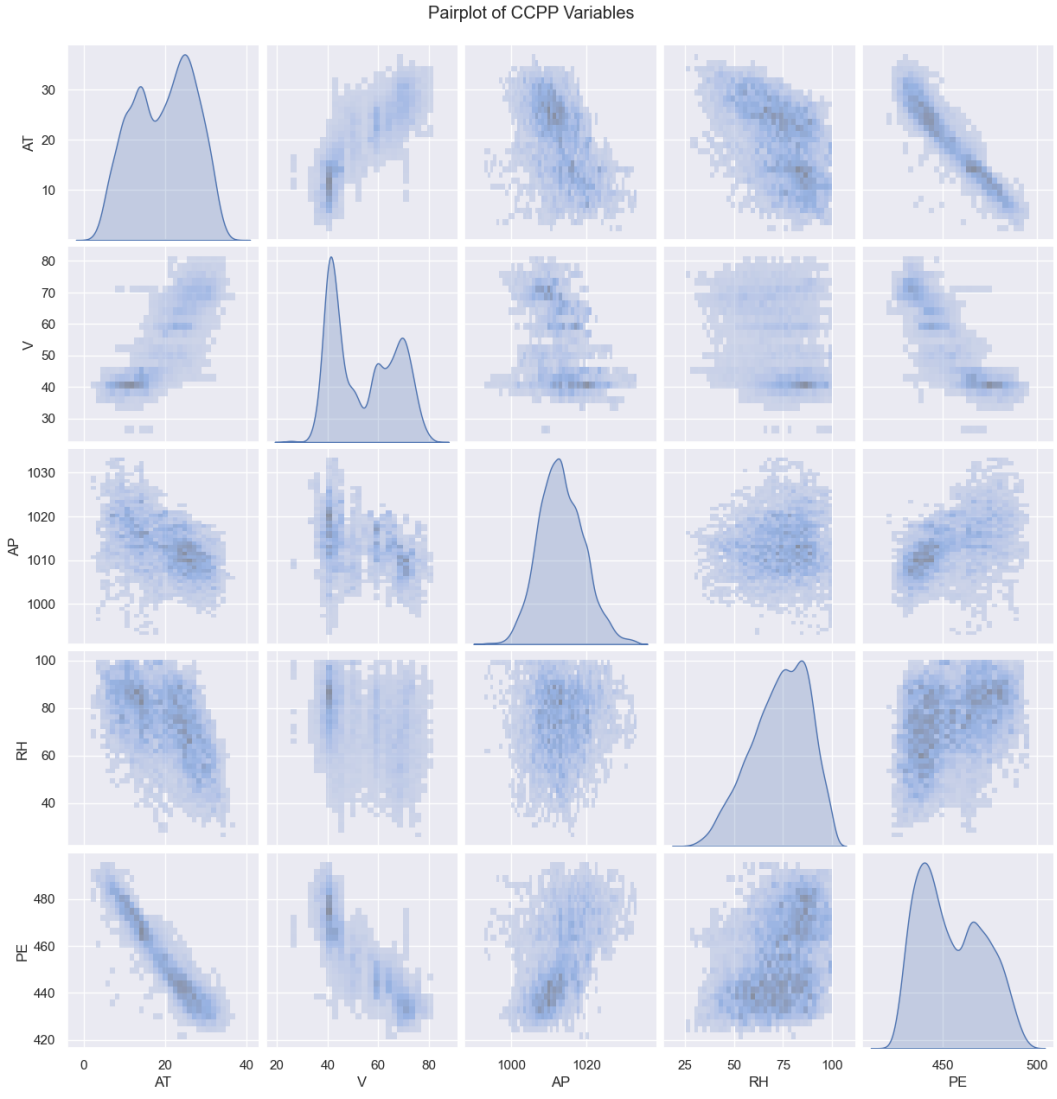

Figure 3: Pair plot of the variables

A pair plot [3.] ábrán részletesebben látható, hogy az egyes változók hogyan viszonyulnak egymáshoz.

Az AT és V erőteljes negatív kapcsolatot mutat a PE-vel, míg az AP viszonylag kevésbé befolyásolja a teljesítményt. Kicsit letisztultabb képet kapunk az adatokról a heatmap segítségével.

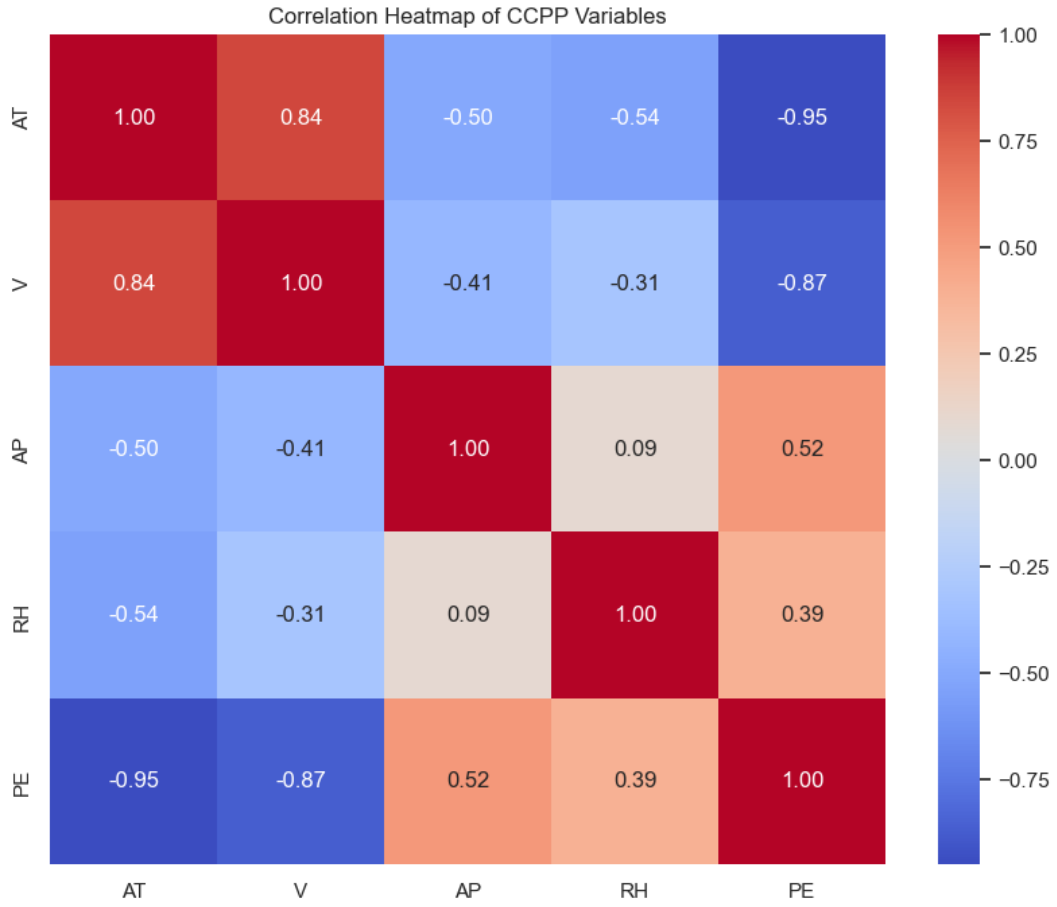

Figure 4: Heat map of variables

A korrelációs heat map [4.] azt mutatja, hogy erős negatív kapcsolat van az AT és a PE között, valamint a V és a PE között is, ami azt sugallja, hogy magasabb AT hőmérséklet és magasabb vákuum V esetén csökken a PE elektromos teljesítmény. Az AP és a PE között mérsékelt pozitív korreláció látható.

3 Adatok skálázása és normalizálása

Mivel az adatok más nagyságrendbe esnek és ez egyes regressziós modellek illesztését elviheti, így az adatokat transzformálva előállítom a standarizált és a normalizált adatsort.

Listing 7: Info about Train

```
scaler = StandardScaler()

X_train_scaled =
    scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Listing 8: Info about Test

```
min_max_scaler = MinMaxScaler()
X_train_normalized =
    min_max_scaler.fit_transform(X_train)
X_test_normalized =
    min_max_scaler.transform(X_test)
```

4 Regressziós modellek és illesztések

4.1 Illesztési eljárás

Végül a tisztított és skálázott adatokra egy sor regressziós modellt illesztettem, a következő függvény segítségével[9.].

Listing 9: Fiting of the regression models

```
def regerssio_modeling(model, X_train_scaled, X_test_scaled, Y_train, Y_test):
    # Model fiting
    model.fit(X_train_scaled, Y_train)
    # Makeing prediction
    Y_pred = model.predict(X_test_scaled)
    # Evaluating/Scoring the model
    mse = mean_squared_error(Y_test, Y_pred)
    r2 = model.score(X_test_scaled, Y_test)
    cv_results = cross_val_score(model, X_test_scaled,Y_test, cv=kf,
        scoring='neg_mean_squared_error')
    return model, Y_pred, mse, r2, cv_results
```

A $regerssio\_modeling(model, X\_train\_scaled, X\_test\_scaled, Y\_train, Y\_test)$ függény főbb lépései:

1. Modell illesztése a skálázott tanító halmazon.
2. Az illesztett modell segítségével az Y_pred predikciók előállítása.
3. MSE és $R2$ értékek kiszámítása
4. Keresztvalidációs eljárás az illesztet modellen, negatív átlogaos négyzeteshibát használva.

Végül a függvény visszatér az illesztett modellel, az MSE, $R2$ valamint a CV kersztvalidációs értékekkel.

4.2 Illesztett modellek

4.2.1 Lineáris regressziós modellek:

Linear Regression: Alap lineáris regressziós modell
Ridge Regression: L2 regularizációval ellátott Ridge regresszió.

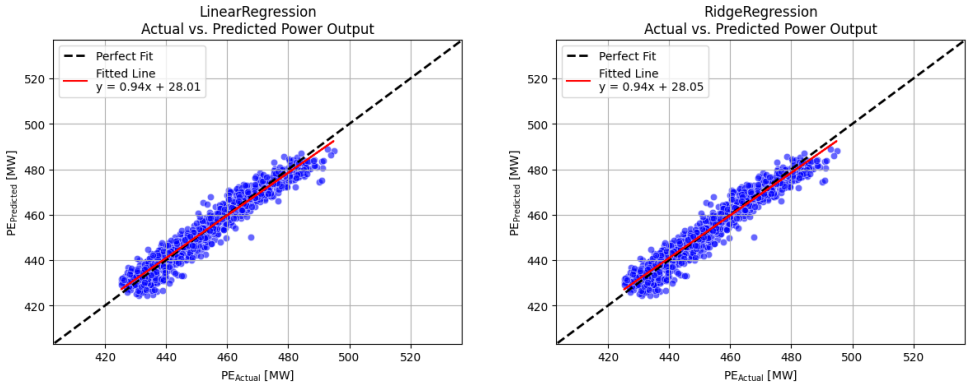

Figure 5: Egyszerű regressziós modellek

4.2.2 KNNq alapú regressziós modellek

KNeighborsRegressor ($N = 5$): Az 5 legközelebbi szomszéd átlagát használja.
KNeighborsRegressor ($N = 10$): Az 10 legközelebbi szomszéd átlagát használja.

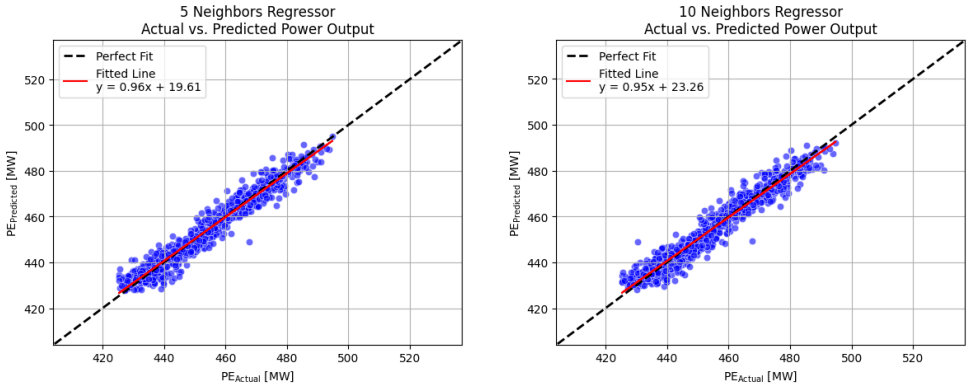

Figure 6: KNN regressziós modellek

4.2.3 Support Vector alapú regressziós modellek (SVR)

SVR (poly kernel): Polinomiális kernellel.
SVR (rbf kernel): Radiális bázisfüggvény (Gauss) kernellel.

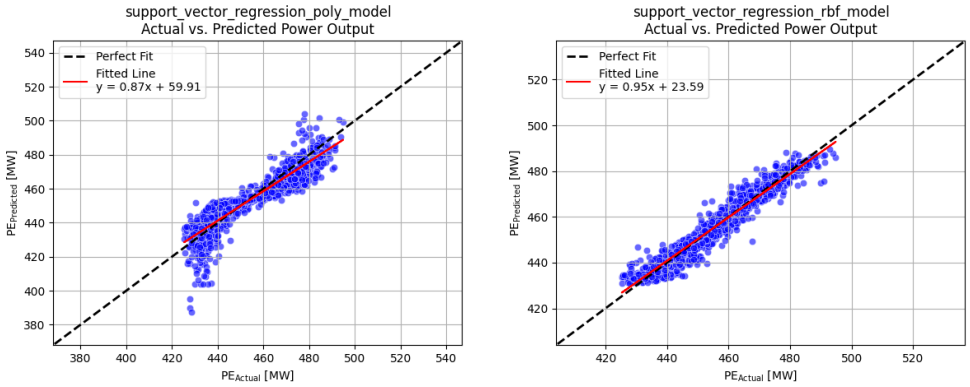

Figure 7: Támogató vektor alapú modellek

4.2.4 Fa-alapú regressziós modellek:

RandomForestRegressor: Több döntési fa kombinációját használja.
DecisionTreeRegressor: Egyetlen döntési fa modell.
BaggingRegressor: SVR alapmodellként, több példány átlagolásával.

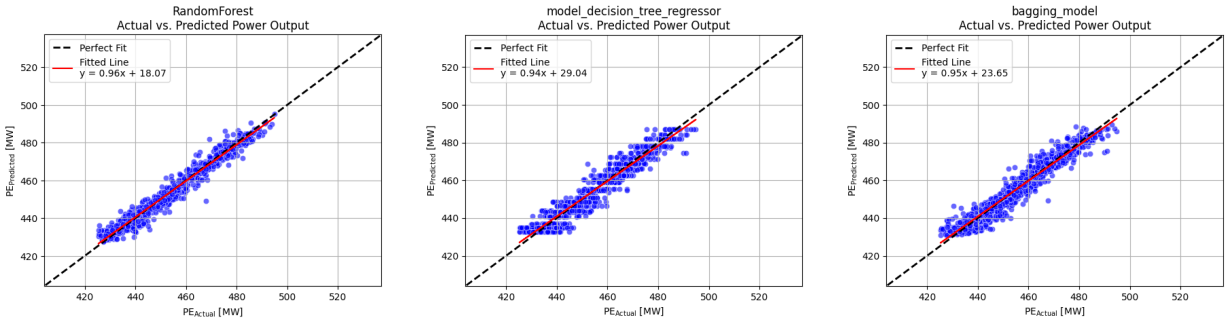

Figure 8: Fa-alapú regressziós modellek

4.2.5 Neurális hálózat alapú regressziós modellek (MLPRegressor)

Multilayer Perceptron 100 RELU ADAM: 100 rejtett neuronnal, RELU aktivációval, ADAM solverrel.
Multilayer Perceptron 100 RELU LBFGS: 100 rejtett neuronnal, RELU aktivációval, LBFGS solverrel.
Multilayer Perceptron 100 IDENTITY ADAM: 100 rejtett neuronnal, IDENTITY aktivációval, ADAM solverrel.
Multilayer Perceptron 100 LOGISTIC ADAM: 100 rejtett neuronnal, LOGISTIC aktivációval, ADAM solverrel.

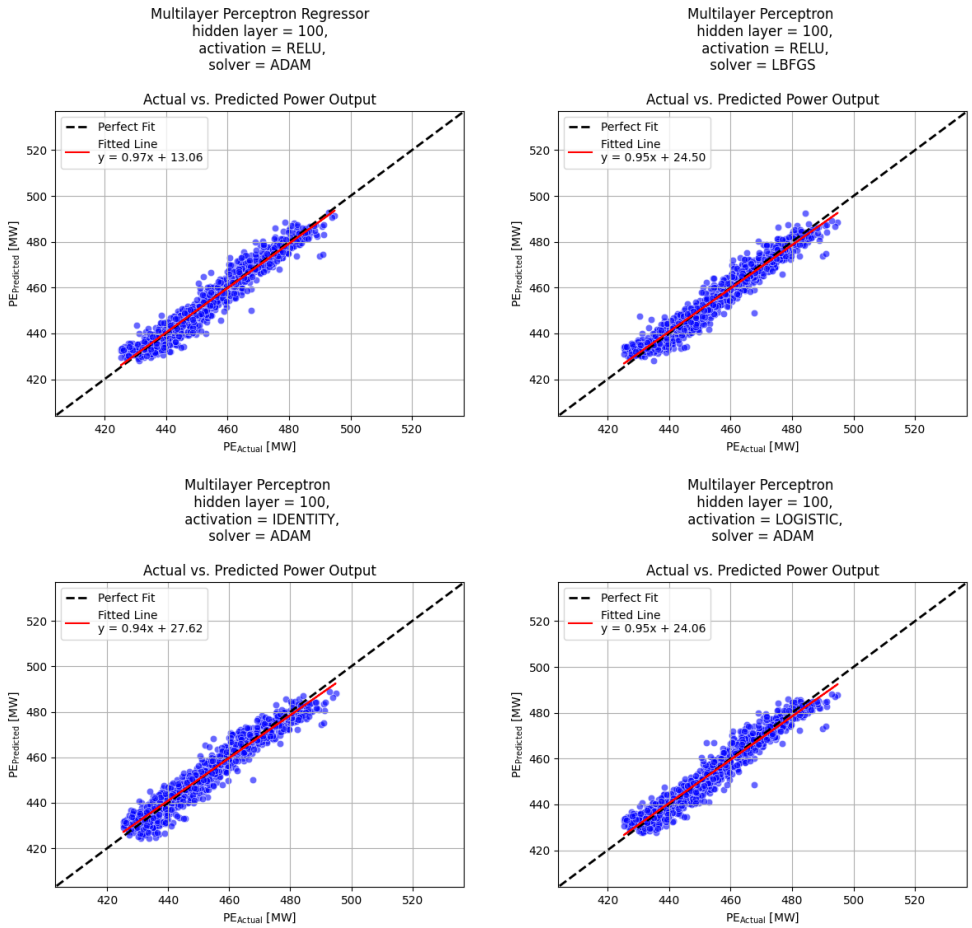

Figure 9: Neurálishálozat alapú regressziós modellek 100 neuronnal, különböző solver és aktivációs függvénnyel

4.2.6 További MLPRegressor variánsok

50, 40, és 30 rejtett neuronnal, minden esetben RELU aktivációval és ADAM solverrel.

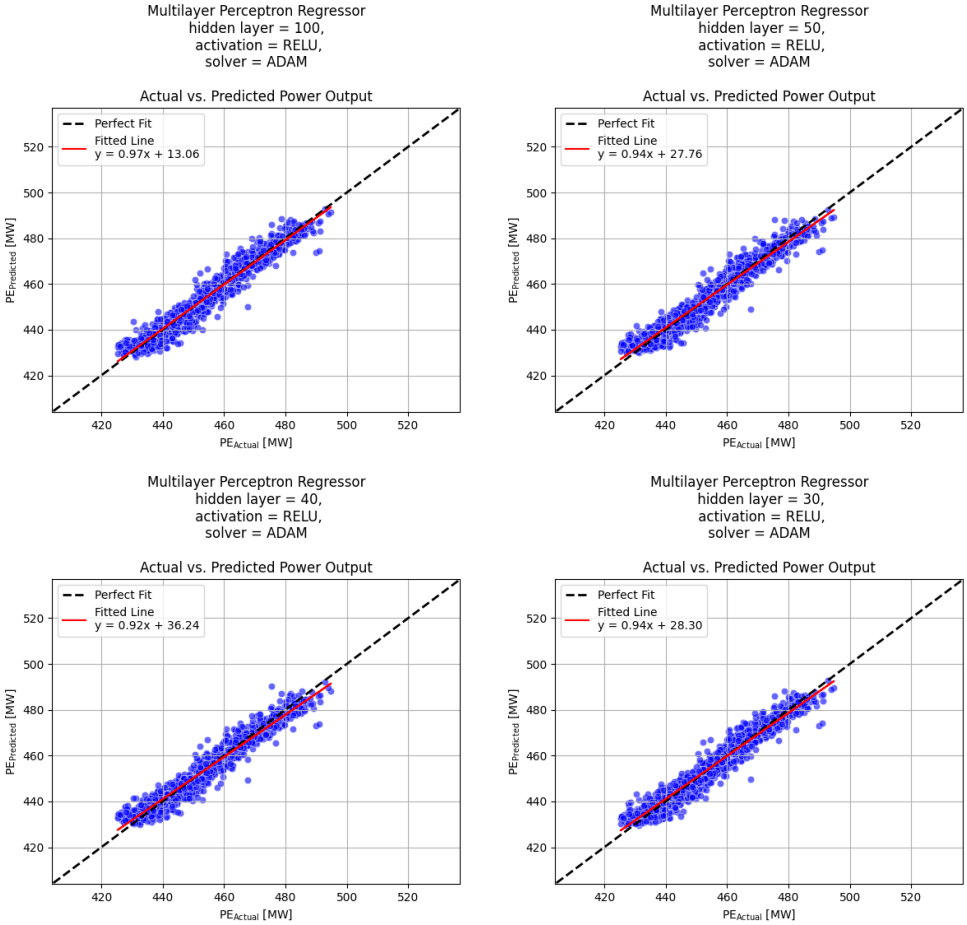

Figure 10: Neurálishálozat alapú regressziós modellek: 30, 40, 50, 100 neuronnal

5 Az MSE, $R2$ és CV értékek

5.1 Az MSE értékek

Az illesztett modellek MSE értékeit ábrázoló eredményeket [11.] növekvő sorrendben rendeztem mivel a kisebb MSE érték mellett teljesít jobban az illesztett modell, így ebből a szempontból a táblázat első elemi lesznek a legjobbak. A következő tábláztat [5.1.] foglalja össze a sorbarendezett adatokat.

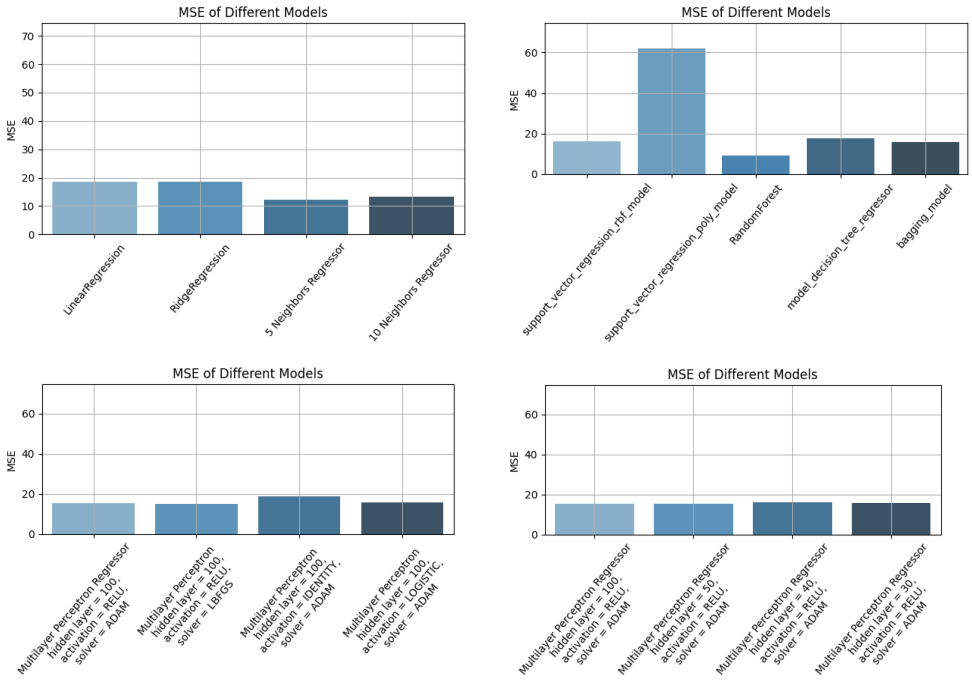

Figure 11: Az illesztett modellek MSE értékei

	Model	MSE
1	RandomForest	9.139454
2	5 Neighbors Regressor	12.332690
3	10 Neighbors Regressor	13.351379
4	MLPRegressor h.l. = 100, act. = RELU, sol. = LBFGS	14.816404
5	MLPRegressor h.l. = 100, act. = RELU, sol. = ADAM	15.217100
6	MLPRegressor h.l. = 100, act. = RELU, sol. = ADAM	15.437234
7	MLPRegressor h.l. = 50, act. = RELU, sol. = ADAM	15.551677
8	MLPRegressor h.l. = 30, act. = RELU, sol. = ADAM	15.904665
9	MLPRegressor h.l. = 100, act. = LOGISTIC, sol. = ADAM	15.933903
10	MLPRegressor h.l. = 40, act. = RELU, sol. = ADAM	15.962066
11	bagging model	16.020971
12	support vector regression rbf model	16.070784
13	model decision tree regressor	17.750337
14	LinearRegression	18.603112
15	RidgeRegression	18.603730
16	MLPRegressor h.l. = 100, act. = IDENTITY, sol. = ADAM	18.608023
17	support vector regression poly model	62.051293

Table 4: Model Mean Squared Error (MSE)

Az növekvekvő sorrendbe rendezett MSE értékeket ábrázoló táblázat [5.1.] alapján a legjobb illesztett modell a Random Forest $MSE = 9.139454$ értékkel.

5.2 Az $R2$ értékek

Továbbá megvizsgáltam az illesztett modellek $R2$ értékeit is [12.], amelyeket szintén táblázatba rendeztem[5.2.]. Minél inkább 1-hez tart az $R2$ értéke annál jobb az illesztett modell.

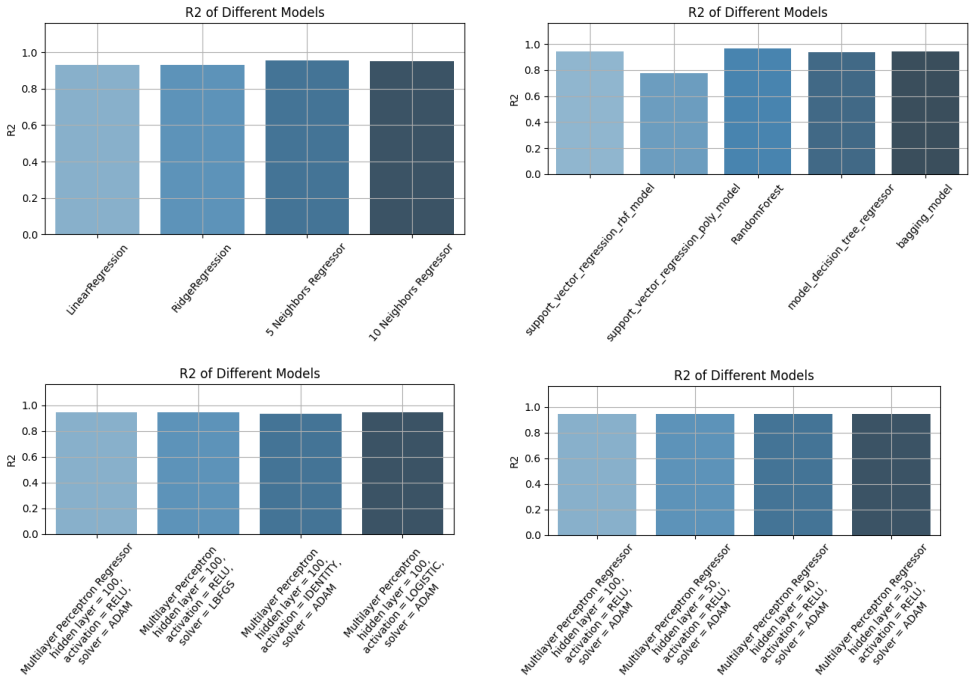

Figure 12: Az illesztett modellek $R2$ értékei

	Model	$R2$
1	RandomForest	0.966764
2	5 Neighbors Regressor	0.955152
3	10 Neighbors Regressor	0.951448
4	MLPRegressor h.l. = 100, act. = RELU, sol. = LBFGS	0.946120
5	MLPRegressor h.l. = 100, act. = RELU, sol. = ADAM	0.944663
6	MLPRegressor h.l. = 100, act. = RELU, sol. = ADAM	0.943862
7	MLPRegressor h.l. = 50, act. = RELU, sol. = ADAM	0.943446
8	MLPRegressor h.l. = 30, act. = RELU, sol. = ADAM	0.942162
9	MLPRegressor h.l. = 100, act. = LOGISTIC, sol. = ADAM	0.942056
10	MLPRegressor h.l. = 40, act. = RELU, sol. = ADAM	0.941954
11	bagging model	0.941739
12	support vector regression rbf model	0.941558
13	model decision tree regressor	0.935451
14	LinearRegression	0.932349
15	RidgeRegression	0.932347
16	MLPRegressor h.l. = 100, act. = IDENTITY, sol. = ADAM	0.932332
17	support vector regression poly model	0.774350

Table 5: Model $R2$

A sorbarendezett $R2$ értékeket bemutató táblázat [5.2.] alapján a legjobb modell ismét a Random Forest.

5.3 A CV értékek

Továbbá az illesztett modellek jóságáról keresztvalidációval győződtem meg. Ehhez az sklearn beépített függvényét használtam és 4 szekcióra osztásos, negatív MSE validálást használtam. Végeredményben azt várom el, hogy a negatív MSE értékek minél közelebb legyenek a nullához.

A különböző szekciókon elért *CV* értékek egyben nehezen olvashatóak. Az áttekinthetőség érdekében a modelleket kisebb csoportokra bontva ábrázoltam minden egyes szekción elért *CV* értéket: [13.], [14.], [15.], [13.], [16.].

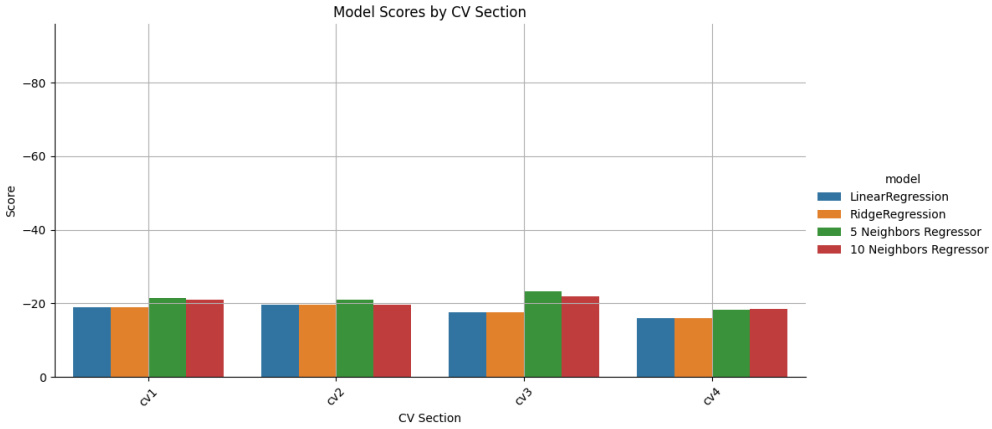

Figure 13: *CV* értétkek az első csoportra

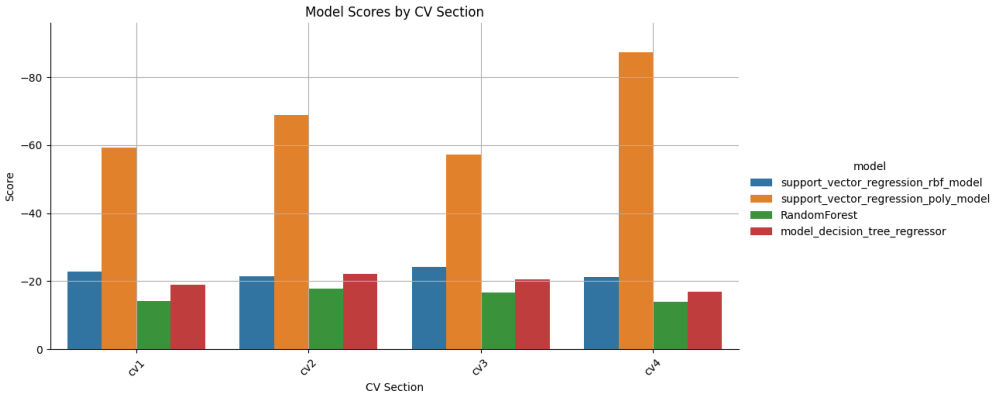

Figure 14: *CV* értétkek a második csoportra

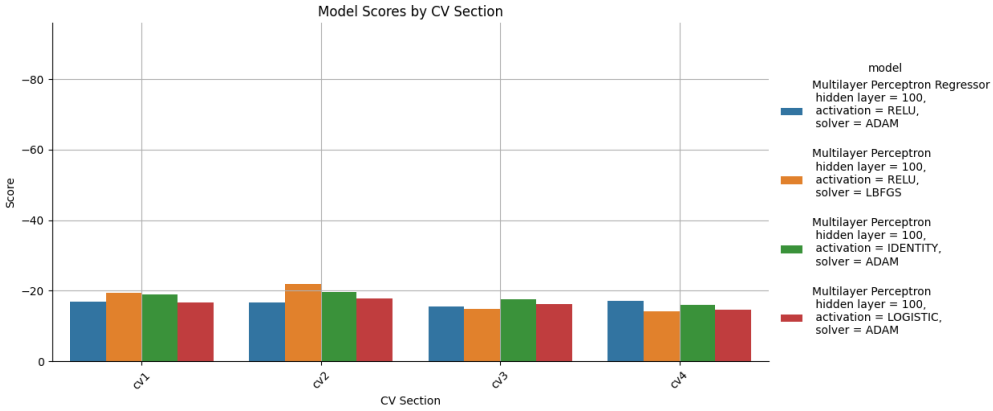

Figure 15: *CV* értétkek a harmadik csoportra

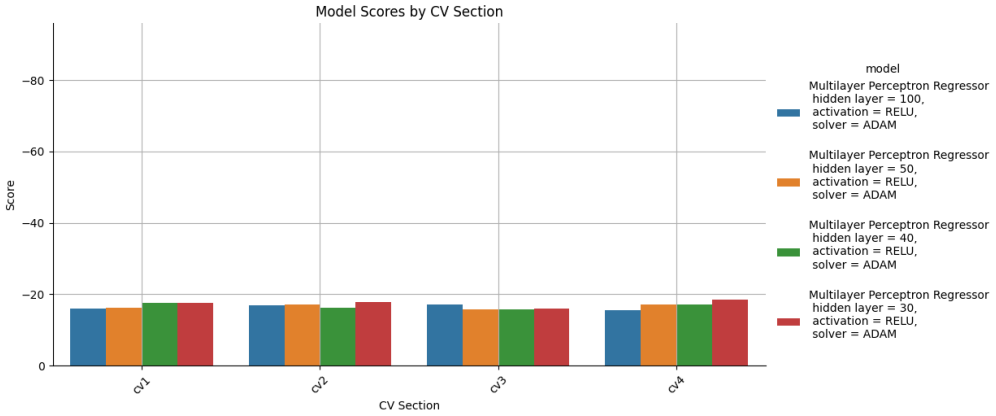

Figure 16: *CV* értétkek a negyedik csoportra

Az ábrázolt *CV* értékeket átlagoltam és szórásukat kiszámoltam. Majd az átlagos *CV* alapján sorba rendeztem és táblázatba foglaltam. [5.3.]

	model	mean	std
1	RandomForest	-15.689774	1.917985
2	MLP Regressor h.l = 100, act. = LOGISTIC, sol. = ADAM	-16.336734	1.346644
3	MLP Regressor h.l. = 100, act. = RELU, sol. = ADAM	-16.397656	0.736577
4	MLP Regressor h.l. = 50, act. = RELU, sol. = ADAM	-16.521706	0.671067
5	MLP Regressor h.l. = 100, act. = RELU, sol. = ADAM	-16.526903	0.638601
6	MLP Regressor h.l. = 40, act. = RELU, sol. = ADAM	-16.677764	0.804133
7	MLP Regressor h.l. = 30, act. = RELU, sol. = ADAM	-17.483360	1.054920
8	MLP Regressor h.l. = 100, act. = RELU, sol. = LBFGS	-17.596056	3.671532
9	LinearRegression	-18.053879	1.678370
10	MLP Regressor h.l. = 100, act. = IDENTITY, sol. = ADAM	-18.055432	1.680409
11	RidgeRegression	-18.058969	1.664198
12	model_decision_tree_regressor	-19.637381	2.204280
13	10 Neighbors Regressor	-20.270605	1.595526
14	5 Neighbors Regressor	-20.986786	2.132396
15	support_vector_regression_rbf_model	-22.452122	1.359736
16	bagging_model	-22.504685	1.261549

Table 6: Mean and std of *CV* scores of the models

Az átlagos keresztvalidációs értékeket és azok szórását bemutató táblázat [5.3.] alapján legjobb modell ismét a Random Forest.
Azonban több modell is hasnló átlagos *CV* értékkel szerepel de eltérő szórással. Így érdekes lehet egy scatter ploton ábrázolni az adatokat.

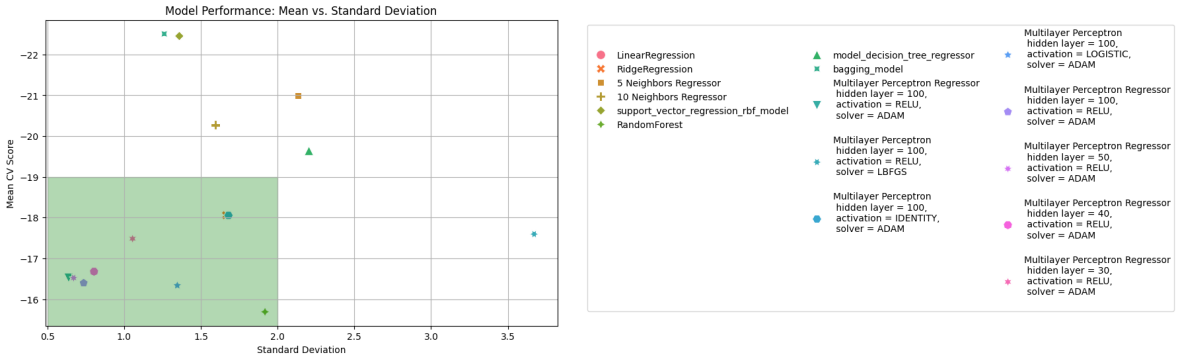

Figure 17: Illesztett modellek átlagos CV értéke és szórása.

A [17.] képen jól látható az illesztett modellek elhelyezkedése az átlagos CV és CV szórás térben. Itt az átlagértétkek negatívak, mivel negatív MSE validálást használtam. Lényegében azt a modelt érdemes kiválasztani aminek átlagos CV értéke és szórása közel 0, tehát a lehető legközelebb esik az origóhoz.
Elsőként a zöld négyzetben található modellekre szűkítettem a lehetséges legjobb modellt választását.

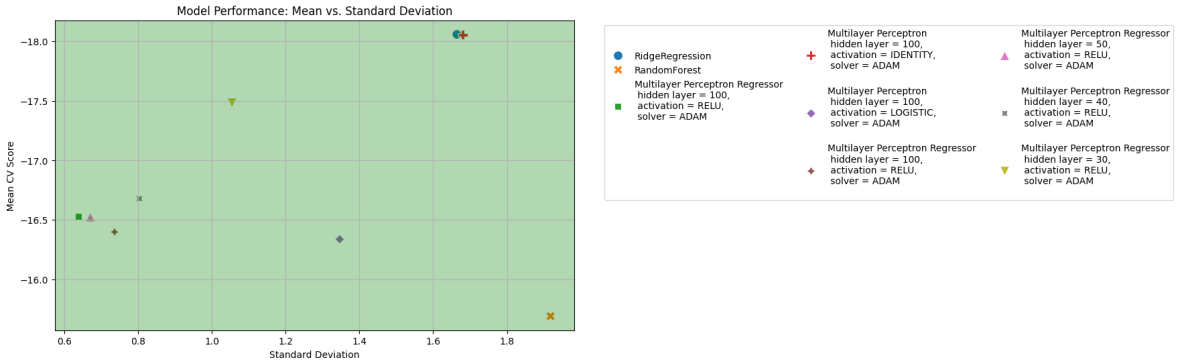

Figure 18: A legjobbnak ítélt modellek csoportja.

Ezt követően tovább szűkítettem a kört és eldobtam azokat a modelleket amelyek CV átlaga ugyan benne van a zöld négyzetben de közel a határhoz. Továbbá azt a modellt is eldobtam amely CV átlaga nem sokkal jobb a többinél, viszont szórása sokkal nagyobb. Így kaptam meg a végső ábrát ahonnan már leolvasható a legjobb modell. [19.]

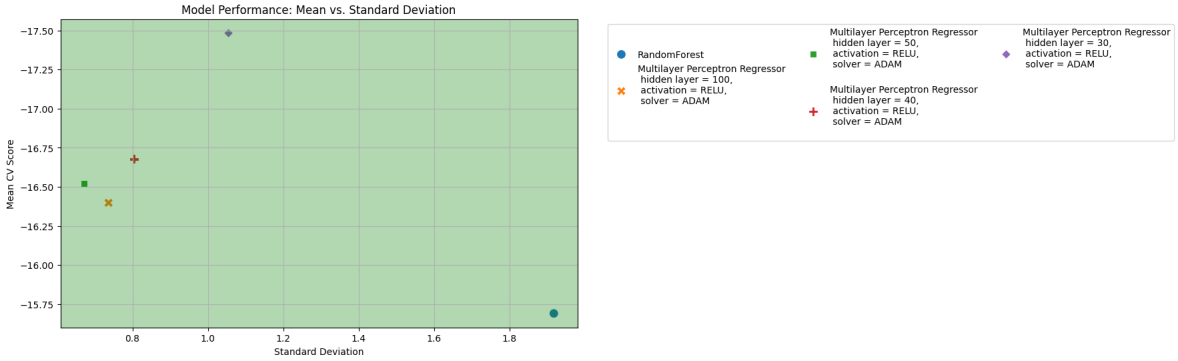

Figure 19: A legszűkebb csoport ahonnan érdemes modellt választani.

A [19.] ábrán nem csak ki tudtam választani a legjobb modellt, de néhány megfigyelést is tettem az MLP Regresszorokra nézve. Ezt az 6. Összegzés és diszkusszió fejezetben részletesen bemutatom.

6 Összegzés és diszkusszió

6.1 Legjobb modell kiválasztása

Az [20.] ábrán bejelöltem piros színnel az általam legjobbnak ítélt modellt. Ez pedig az **MLP Regresszor 100 neuronnal, ADAM solverrel és RELU aktivációs függvénnyel**. Továbbá megfigyeltem, ahogyan növekszik a modell teljesítménye a neuronok számának 30-ról 100-ig történő növelésével amelyet kék nyíllal jelöltem.

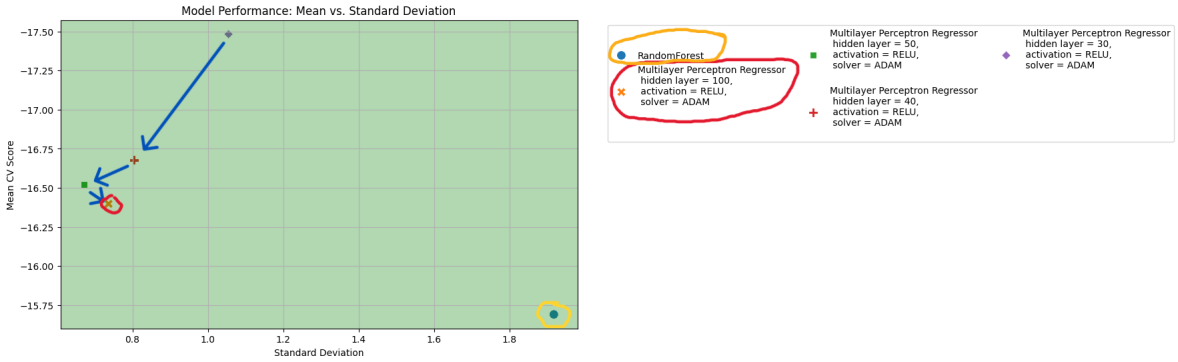

Figure 20: A legjobb modell piros színnel jelölve. A kék nyilak mutatják a MLP Regresszor modell teljesítményének változását a neuronok növelésével. A narancssárga keret pedig egy alternatív legjobb választást jelöl.

6.2 Alternatív legjobb modell

Egy másik alternatív modell is kiválasztásra került, ez pedig nem más mint a korábbi MSE és $R2$ táblázatokban legjobbnak ítélt **Random Forest**.
Azért tartom fontosnak megemlíteni ezt modellt is mint lehetséges jó megoldást, mivel ez a modell átlagosan egy kicsivel jobban teljesít a keresztvalidáció 4 tartományán. Jól lehet, a szórása (1.917985) több mint kétszerese a legjobbnak ítélt MLP Regresszornál (0.736577).
Mivel jelenleg a szomszédban dúló háború miatt a velünk egy hálózatba kapcsolt Ukrajna elveszítette az erőművi kapacitásainak több mint 10%-, továbbá a rendszerbe kötött megújuló energiaforrások leadott teljesítménye is ingadozó így elképzelhető, hogy jobb egy átlagosan körülbelül 5%-al jobban teljesítő modell, a teljes hálózat szempontjából, még akkor is ha a szórása több mint kétszerese az MLP Regresszorénak. Ilyen eset lehet például az amikor hosszú évekig nem lehet jelentősen javítani a hálózatot, és így az ingadozások elfogadhatóak.

6.3 Kódbázis

Ezen dolgozat minden eleme, a kiértékeléssel kapcsolatos adatok és lejárások megtalálhatóak a gitHub oldalamon. https://github.com/szilagyi93/regression/tree/main/01_main.

A főbb lépések a következő 3 jupyter notebookban találhatóak:

1. eda_ccpp.ipynb: Adtok beolvasása, tisztítása és EDA .
2. modelfitting_ccpp.ipynb: Regressziós modellek példányosítása, illesztés és predikált vs. aktuális adatok és ábrák (ennek futtatása körülbelül 7-9 percet vesz igénybe).
3. score_evaluation_ccpp.ipynb: az illesztett modellek MSE, $R2$ és CV értékeit itt értékelem ki, foglalom táblázatba és ábárázolom.

Az egyszerűség kedvéért .html formátumban is feltöltöttem az említette notebook-okat így, egy böngészőből is megnyitható.

6.4 Kimaradt részek és potenciális fejlesztési pontok

Ahogy a dolgozat elején is írtam, egy lehetséges fejlesztési pont lenne, hogy ha nem az összes változót, hanem azoknak egy csoportját használnám fel modell illesztéshez. Ehhez valószínűleg jobb lenne az sklearn helyett más könyvtárat használni.
Továbbá jupyter notbook helyett ekkor már más fejlesztőkörnyzetben rendes OO programot kell írni, tesztekkel megtámogatva.
Végül az adatokat és a függvényillesztést is elegáns lenne kiszervezni egy felhőszolgátásba.
Végezetül pedig egy példát lehetne mutatni a túltanult és alul tanult modellre.