

# Примена на регресиони модели за предвидување на потрошувачка на електрична енергија

---

АНДРЕЈ СТАНОЈКОВИЌ 186039

# Подготовка на податоците

Множеството за тренирање содржи податоци за временските фактори (температура, притисок, брзина на ветер) и нивното влијание врз потрошувачката на електрична енергија. Податоците се однесуваат на првите 23 дена од месеците во периодот помеѓу 2013 до 2017 година.

train

	ID	datetime	temperature	var1	pressure	windspeed	var2	electricity_consumption
0	0	2013-07-01 00:00:00	-11.4	-17.1	1003.0	571.910	A	216.0
1	1	2013-07-01 01:00:00	-12.1	-19.3	996.0	575.040	A	210.0
2	2	2013-07-01 02:00:00	-12.9	-20.0	1000.0	578.435	A	225.0
3	3	2013-07-01 03:00:00	-11.4	-17.1	995.0	582.580	A	216.0
4	4	2013-07-01 04:00:00	-11.4	-19.3	1005.0	586.600	A	222.0
...	...	...	...	...	...	...	...	...
26491	34891	2017-06-23 19:00:00	-0.7	-15.0	1009.0	51.685	A	225.0
26492	34892	2017-06-23 20:00:00	-2.9	-11.4	1005.0	56.105	A	213.0
26493	34893	2017-06-23 21:00:00	-1.4	-12.9	995.0	61.275	A	213.0
26494	34894	2017-06-23 22:00:00	-2.9	-11.4	996.0	67.210	A	210.0
26495	34895	2017-06-23 23:00:00	-2.1	-11.4	1009.0	71.880	A	210.0

26496 rows × 8 columns

# Подготовка на податоците

Множеството за тестирање содржи податоци за истите временски фактори, но од 24тиот до последниот ден од месецот во периодот помеѓу јули 2013 до јули 2017. Целта е да се предвиди потрошувачката на електрична енергија за секој час од овие денови врз основа на податоците од претходните 23 дена.

test

	ID	datetime	temperature	var1	pressure	windspeed	var2
0	552	2013-07-24 00:00:00	-10.0	-16.4	1011.0	263.280	A
1	553	2013-07-24 01:00:00	-10.0	-20.7	1011.0	267.175	A
2	554	2013-07-24 02:00:00	-10.7	-17.1	1003.0	269.555	A
3	555	2013-07-24 03:00:00	-13.6	-20.7	1008.0	273.060	A
4	556	2013-07-24 04:00:00	-10.7	-17.1	1006.0	1.765	A
...	...	...	...	...	...	...	...
8563	35059	2017-06-30 19:00:00	-5.7	-18.6	998.0	233.595	A
8564	35060	2017-06-30 20:00:00	-5.7	-17.1	995.0	238.780	A
8565	35061	2017-06-30 21:00:00	-7.1	-19.3	1004.0	244.325	A
8566	35062	2017-06-30 22:00:00	-6.4	-19.3	1008.0	247.470	A
8567	35063	2017-06-30 23:00:00	-5.0	-16.4	1001.0	250.600	A

8568 rows × 7 columns

# Опис на податоците

---

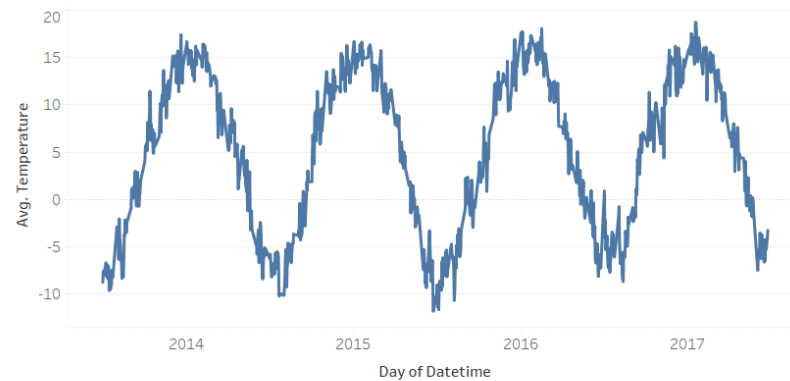
```
train.describe()
```

	ID	temperature	var1	pressure	windspeed	var2	electricity_consumption
<b>count</b>	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000	26496.000000
<b>mean</b>	17455.500000	5.098989	-1.916233	986.450615	23.959956	0.086692	298.359601
<b>std</b>	10122.873673	8.682860	10.424860	12.002647	48.280321	0.397096	108.020555
<b>min</b>	0.000000	-17.100000	-32.900000	953.000000	1.075000	0.000000	174.000000
<b>25%</b>	8717.750000	-2.900000	-10.700000	978.000000	3.155000	0.000000	219.000000
<b>50%</b>	17435.500000	6.400000	-1.400000	986.000000	6.545000	0.000000	267.000000
<b>75%</b>	26177.250000	12.100000	7.900000	995.000000	22.260000	0.000000	342.000000
<b>max</b>	34895.000000	23.600000	18.600000	1024.000000	586.600000	2.000000	1386.000000

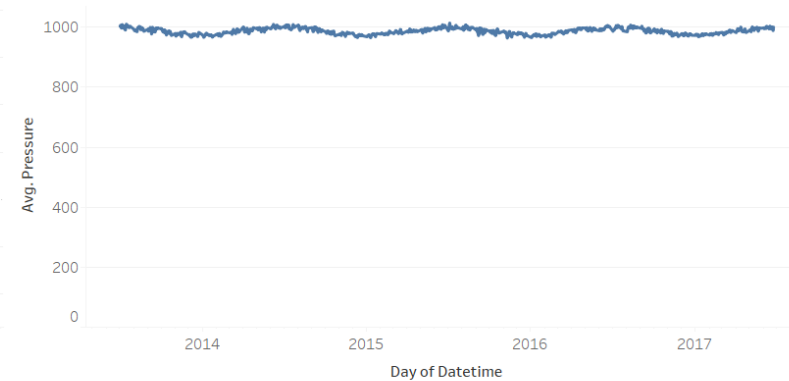
# Визуелизација на податоците

---

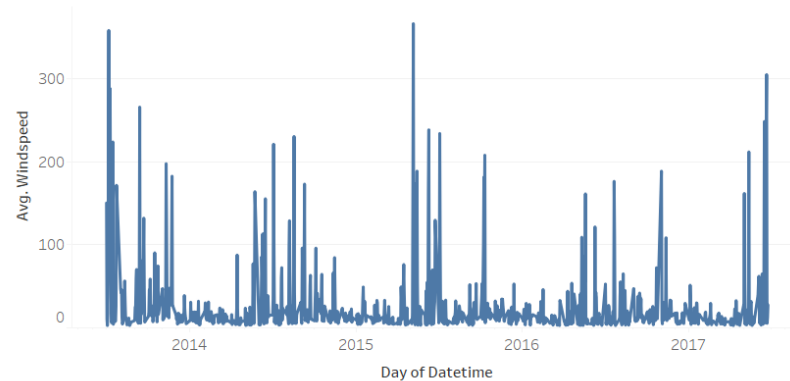
Temperature



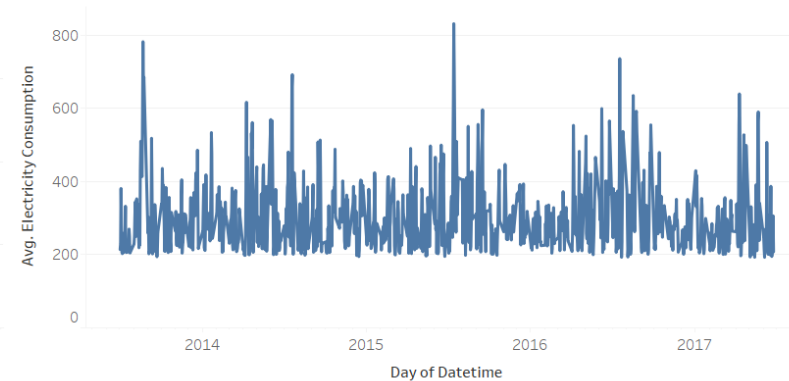
Pressure



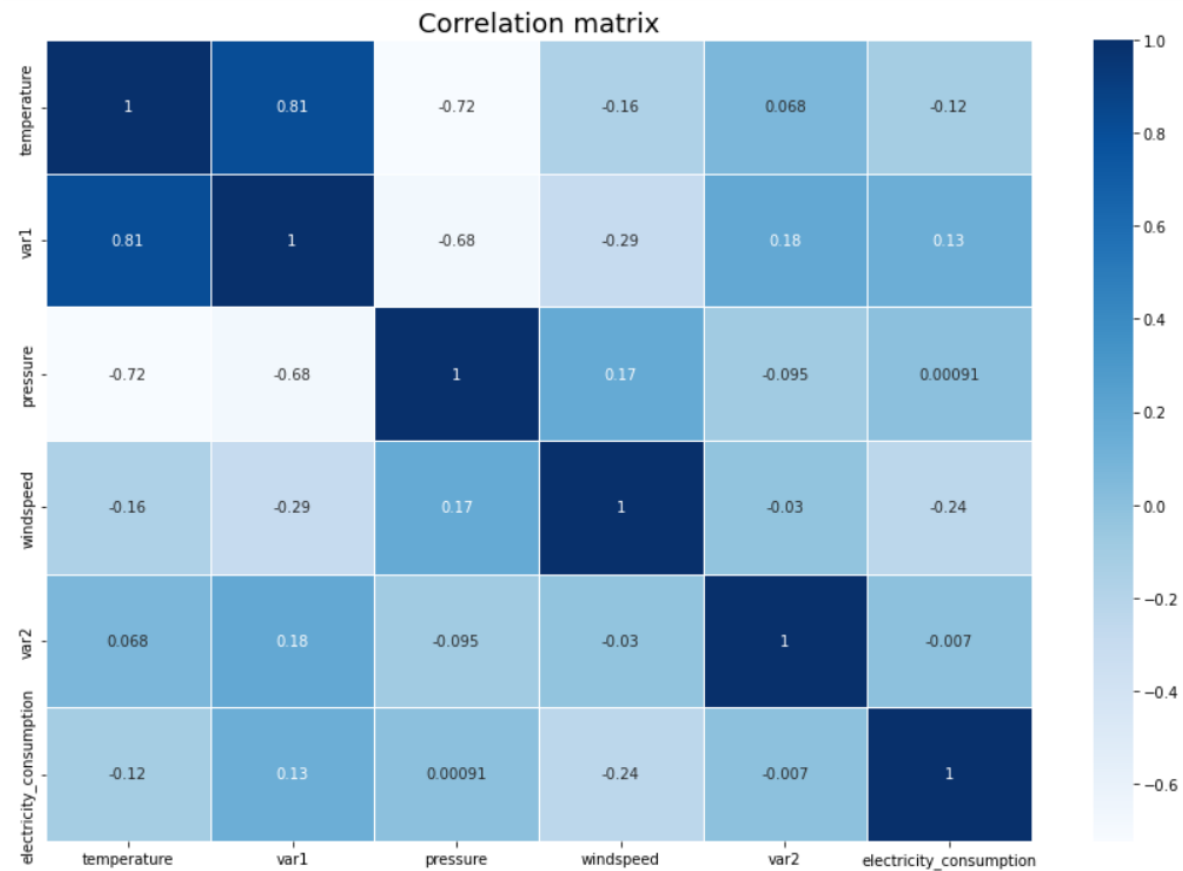
Windspeed



Electricity consumption



# Визуелизација на податоците



# Регресиони модели

---

## Линеарна регресија

```
linear = LinearRegression().fit(X_train, Y_train)
Y_pred_linear = linear.predict(X_test)
Y_pred_linear_df = pd.DataFrame(Y_pred_linear, columns = ['Linear'])
```

## Ridge регресија

```
ridge = Ridge().fit(X_train, Y_train)
Y_pred_ridge = ridge.predict(X_test)
Y_pred_ridge_df = pd.DataFrame(Y_pred_ridge, columns = ['Ridge'])
```

## Lasso регресија

```
lasso = Lasso().fit(X_train, Y_train)
Y_pred_lasso = lasso.predict(X_test)
Y_pred_lasso_df = pd.DataFrame(Y_pred_lasso, columns = ['Lasso'])
```

# Регресиони модели

---

## ElasticNet регресија

```
elasticNet = ElasticNet().fit(X_train, Y_train)
Y_pred_elasticNet = elasticNet.predict(X_test)
Y_pred_elasticNet_df = pd.DataFrame(Y_pred_elasticNet, columns = ['ElasticNet'])
```

## Полиномна регресија

```
polynomial = PolynomialFeatures(degree=4)
X_polynomial = polynomial.fit_transform(X_train)
polynomial.fit(X_polynomial, Y_train)
linear2 = LinearRegression().fit(X_polynomial, Y_train)
Y_pred_linear2 = linear2.predict(polynomial.fit_transform(X_test))
Y_pred_polynomial_df = pd.DataFrame(Y_pred_linear2, columns = ['Polynomial'])
```

## Баесова Ridge регресија

```
bayesianRidge = BayesianRidge().fit(X_train, Y_train)
Y_pred_bayesianRidge = bayesianRidge.predict(X_test)
Y_pred_bayesianRidge_df = pd.DataFrame(Y_pred_bayesianRidge, columns = ['BayesianRidge'])
```



# Регресиони модели

---

## LSTM

```
lstm = Sequential()  
lstm.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], 1)))  
lstm.add(LSTM(units=25))  
lstm.add(Dense(units=1))  
  
lstm.compile(optimizer='adam', loss='mean_squared_error')  
lstm.fit(X_train, Y_train, epochs=20, batch_size=32)
```

```
Y_pred_lstm = lstm.predict(X_test)  
Y_pred_lstm_df = pd.DataFrame(Y_pred_lstm, columns = ['LSTM'])
```

# Заклучок

Врз основа на добиените резултати може да се заклучи дека моделите за линеарна, Ridge, Lasso, ElasticNet и баесов Ridge даваат слични предвидувања, додека моделите за полиномна регресија и LSTM даваат малку поразлични резултати.

Со овие модели од МУ може да се предвиди потрошувачката на електрична енергија во наредните денови и седмици, бидејќи временските услови се познати неколку седмици однапред. Со тоа може да се контролира производството, т.е. да нема недостаток или вишок на електрична енергија.

	datetime	Linear	Ridge	Lasso	ElasticNet	Polynomial	BayesianRidge	LSTM
0	2013-07-24 00:00:00	236.391505	236.390727	235.330975	232.497761	218.846710	236.328196	209.354782
1	2013-07-24 01:00:00	207.786464	207.786080	207.337659	205.813551	229.994408	207.755171	207.816788
2	2013-07-24 02:00:00	238.781049	238.780204	237.467120	234.326758	219.346613	238.712218	209.113754
3	2013-07-24 03:00:00	236.726104	236.725383	235.603271	232.726572	236.608766	236.667336	208.321808
4	2013-07-24 04:00:00	335.290346	335.290169	335.028741	334.274316	296.108050	335.275956	271.245575
...	...	...	...	...	...	...	...	...
8563	2017-06-30 19:00:00	203.029452	203.029214	202.716372	201.891047	209.632051	203.010103	207.320679
8564	2017-06-30 20:00:00	211.857301	211.856888	211.223963	209.797378	210.657381	211.823675	207.559158
8565	2017-06-30 21:00:00	203.820137	203.819867	203.514979	202.521080	214.438002	203.798138	207.448410
8566	2017-06-30 22:00:00	195.217499	195.217309	195.122615	194.455658	211.894813	195.202076	207.367371
8567	2017-06-30 23:00:00	203.688087	203.687683	203.187206	201.853556	209.982771	203.655212	207.662399

8568 rows × 8 columns