

# Napoved sočnega sevanja z metodami regresije

---

## **Seminarska naloga pri predmetu Inteligentni sistemi**

Nosilca: doc. dr. Marko Meža in izr. prof. dr. Matej Zajc.

1. stopnja, 3. Letnik – Elektrotehnika-Elektronika (5. Semester)

Avtorji: Alen Čaušević, 64110008

Tine Bertoncelj, 64130066

Ljubljana, 30.12.2017

## Kazalo

Uvod.....	4
Predstavitev problema.....	4
Podatki.....	5
Opis izbranega nabora podatkov.....	5
Pridobitev podatkov.....	5
Vsebina podatkov in njihov opis.....	5
Manjkajoči del podatkov.....	6
Nadzorovano učenje.....	7
Linearna regresija.....	7
Teoretični opis modela.....	7
Opis uporabe rezultatov modela.....	8
Opis dela.....	8
Predobdelava podatkov.....	8
Gradnja modela.....	8
Validacija modela.....	10
Predstavitev interpretacija rezultatov.....	11
Linearna regresija z več spremenljivkami.....	12
Teoretični opis modela.....	12
Predstavitev in interpretacija rezultatov.....	13
Gradient Boosted Trees.....	14
Opis izbrane metode za gradnjo modela.....	14
Opis dela.....	14
Predobdelava podatkov.....	15
Gradnja modela.....	16

Validacija modela.....	17
Predstavitev in interpretacija rezultatov.....	17
Zaključek.....	19
Viri.....	20

## Uvod

### Predstavitev problema

V zadnjem stoletju je bliskovit napredek človeštva spravil raven življenjskega standarda višje kot še nikoli prej. Ker smo začeli odkrivati nove načine uporabe energije v prid nam, se je izkazalo čez čas, da to za sabo nosi tudi negativne posledice. Prvič smo bili soočeni s problemom globalnega segrevanja, ki je bila ena izmed posledic intenzivne uporabe fosilnih virov, ki so postali glavni vir energije našega napredka. Razvoj novih tehnologij nam daje upanje, da bomo nekega dne preklopili na obnovljive vire (OVE), ki zajemanje teh se ne izčrpa in uporaba njih pusti manjše negativne posledice našemu planetu.

Fotonapetostna energija ki deluje na zajemanju energije iz stalnega procesa sončnega sevanja, je najbolj obetavna tehnologija, ki zna v prihodnosti napajati svet. Današnji elektroenergetski sistem (EES) je zasnovan tako, da mora proizvodnja električne energije, vsak čas zadostiti potrebam po moči porabnikov. Težji del je, zagotoviti stabilno proizvodnjo moči fotonapetostnih modulov, ki je pa močno ovirano z vremenskimi vplivi, neenakomerno sončno radiacijo, letnimi časi...

Pri napovedi proizvodnje električne energije iz obnovljivih virov je nujno potrebno uporabiti natančno napoved vremenskih vplivov in iz njih naučiti računalniške algoritme, ki bodo iz njih znali napovedati proizvedeno električno energijo. Če bomo hoteli imeti stabilno električno omrežje, bodo dobri napovedovalni algoritmi proizvodnje energije ključ, do popolne zamenjave konvencionalnih virov z obnovljivimi.

Nalogo sva si zastavila tako, da sva po pregledu in ureditvi podatkovne množice dodelila obema po eno metodo strojnega učenja in nato primerjala rezultate in primernost izbrane metode.

## Podatki

### Opis izbranega nabora podatkov

Za dobro delujoč algoritem je potrebno dobiti bazo podatkov, ki bo imela več dnevni natančnih meritev vremenskih sprememb in izmerjeno proizvedeno fotovoltaično električno energijo. Izkaže se, da za pridobitev zabeleženih vremenskih podatkov je na spletu mnogo brezplačnih hranjen baz, ki nudijo velik razpon različnih meritev od temperature do količine zapadenega snega.

Problem je pridobitev minutne ali celo dnevne proizvodnje katere koli sončne elektrarne, ker te javno ne objavljajo podatkov na splet. Da bi se računalniško učenje lahko izvedlo, nujno potrebujemo oboje vrst podatkov.

### Pridobitev podatkov

Podatke je pridobila ameriška vesoljska agencija NASA, uporabljeni pa so bili kot del tekmovanja Space Apps Challenge 2017 v Moskvi. V obliki CSV datoteke je so podani podatki o vremenu na Havajih v obdobju med septembrom in decembrom 2016. Zajem podatkov je potekal na približno 5 minut.

### Vsebina podatkov in njihov opis

V tabeli podatkov so bile merjene naslednje veličine:

- Datum in čas -> Časovno zabeleženo, kdaj je bila izvedena meritev
- Temperatura -> Merjena v Fahrenheitovi lestvici [ $^{\circ}F$ ], ki je dober pokazatelj, koliko sončnega sevanja je prišla okolica. Je tudi eden od najbolj preprostih povezav med napovedjo sončnega sevanja in količino temperature.
- Zračni pritisk -> Merjen za barometrov [ $kPa$ ], ki meri pritisk atmosfere v milibarih. Visok povprečen pritisk je znak mirnega in sončnega vremena. Nizek zračni pritisk pa pomeni slabo vreme z možnostmi padavin.
- Vlažnost -> Merjena v procentih %, pomeni kolikšna je količina vodnih hlapov v zraku pri določeni temperaturi. Je tudi izvor vseh oblakov, megal in dežja. Vodni hlapi pomagajo ogreti tudi ozračje s tem ko absorbirajo energijo sonca.

- Smer in hitrost vetra -> Zrak se giblje iz točke višjega pritiska v območje nižjega pritiska. Enote merjene hitrosti so  $m/s$ . Če ni nobenega vetra, tudi večjih sprememb v vremenski napovedi ne bo. Pokazatelj vremena je tudi smer vetra, če ta pride iz območja kjer je velik del površin voda, bo za seboj pobral del vodnih hlapov in velika verjetnost je da, bo pripeljal padavine.
- Čas sončnega vzhoda in zahoda -> Ključen podatek pri napovedi sončne radiacije, ker pred sočnim vzhodom in po sončnem zahodu je proizvodnja sočne energije nič. Med časov sončnega sevanja pa je proizvedena energija odvisna tudi z do sedaj opisanimi parametri.
- Sončna radiacija -> Se meri kot vpadla moč na kvadratni meter.  $\frac{W_{solar}}{m^2}$ . Pri našem strojnem učenju se obnaša kot izhod algoritma in s tem tudi povezana proizvedena električna energija.

Ob vsakem merjenju so zraven podani še datum, čas in UNIX timestamp. Vse značilke v datasetu so numerične, razen datuma, časa in časa sončnega vzhoda in zahoda, ki pa so tipa DateTime. Na srečo ni manjkajočih podatkov pri posameznih meritvah, vendar pa obstajajo občasne luknje med posameznimi meritvami, ki lahko trajajo nekaj dni zapored.

## Manjkajoči del podatkov

Manjkajoči del podatkov je bil v naši bazi učnih podatkov, ki ni bil beleženj za noben parameter. Preprosto se je ta del izpustil iz nabora učnih podatkov. Končni rezultati so pokazali, da odločitev izbrisa teh podatkov ni imela večjega vpliva na rezultate.

## Nadzorovano učenje

Za napovedovanje sončnega sevanja iz nabora dodeljenih podatkov uporabljamo tehniko nadzorovanega učenja, kjer iščemo iz vhodnih podatkov  $X$ , neko "dobro" hipotetično funkcijo  $h(x)$ , ki nam da izhodno vrednost  $Y$ . V tem primeru imamo vhod in izhod zapisan kot realne vrednosti  $X = Y = \mathbb{R}$ , ki so preko hipotetične funkcije zvezno povezani. Tak problem bomo rešili z regresivno metodo.

## Linearna regresija

Za začetek smo vzeli linearno regresijo, kjer je bil  $X$  eno-dimenzionalen vektor z temperaturnim parametrom. Za izvedbo nadzorovanega učenja smo določili hipotetično funkcijo  $h(x)$ , ki bo aproksimirala rezultat  $Y$  v najbolj preprosti linearni obliki  $h(x) = \theta_0 + \theta_1 x_1$ .

### Teoretični opis modela

Tukaj so  $\theta$  parametri tako-imenovane uteži, ki dodajo parametre v prostor z linearno funkcijo in slikajo vhodne spremenljivke  $X$  v izhod  $Y$ . Sedaj ko imamo naše učne podatke, iz njih določimo parametre  $\theta_0$  in  $\theta_1$ . Poišče se ju iz funkcije  $h(x)$ , da ustvarimo čim manjšo napako med realnimi pričakovanimi podatki  $Y$  in našimi predvidenim izhodom. Formalno to napravi z funkcijo, ki meri za vsak parameter  $\Theta$  kako blizu od  $h(x^{(i)})$  so ustrezni  $y^{(i)}$ . Definirano jo kot funkcijo

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Sedaj moramo izbrati parametre  $\Theta$  tako, da bomo čim bolj uspešno minimizirali funkcijo  $J(\Theta)$ . Na začetku naključno izberemo vrednost  $\Theta$  in nato postopoma spreminjamo parametre  $\Theta$  da se nam funkcijo  $J(\Theta)$  prične zmanjševati. Uporabi se gradientni spust, ki večkratno opravi posodobitve parametre  $\Theta$  na ta način:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

kjer se alfa  $\alpha$  imenuje faktor učenja. Ko gradientni spust implementiramo v našo  $J(\Theta)$  funkcijo dobimo algoritem:

$$\theta_j := \theta_j - \alpha (y^i - h_{\theta}(x^{(i)})) x_j^{(i)}$$

, ki ga ponavljamo do konvergence.

## Opis uporabe rezultatov modela

Rezultat nam bo podal napoved sončnega sevanja iz vnesene napovedane temperature za naslednje dneve. S pomočjo linearne regresije nam bo algoritem znal povezati vhod z izhodom. Izpis bo točkovno prikazan na grafu, za vsako vrednostjo temperature bo tudi ekvivalentno temu sončna radiacija.

## Opis dela

Linearna regresija se bo izdelala v programskem okolju Octave. Orodje nam nudi, že spisane operacije med vektorji, ki pridejo prav pri implementaciji modelov regresije. Velik doprinos programa je tudi ličen izpis grafov.

## Predobdelava podatkov

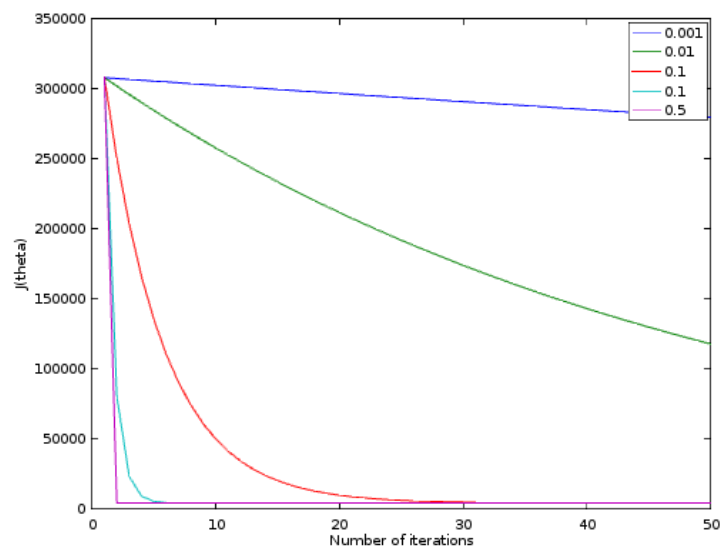
Za začetek se aplicira linearno regresijo z eno vrsto niza spremenljivk  $x^{(i)}$ . Ta pa je v našem primeru temperatura. Izkaže se, da temperatura niha med 45 in 60, in izhod  $y^{(i)}$  pa med 0 in 1200 vrednostmi, kar nam povzroča probleme pri iskanju aproksimacijske črte. Prvo je potrebno vse podatke skalirati, da se njihove vrednosti nahajajo med -1 in 1. To se napravi tako, da najprej preračunamo povprečno vrednost značilke in jo odštejemo od podanih podatkov. Nato poračunamo standardno deviacijo za vsako značilko in delimo vsako posebej z standardno deviacijo.

```
mu = mean(X);  
sigma = std(X);  
for i=1:size(X,2)  
    X(:,i) = (X(:,i) - mu(1,i))/sigma(1,i);  
end
```

## Gradnja modela

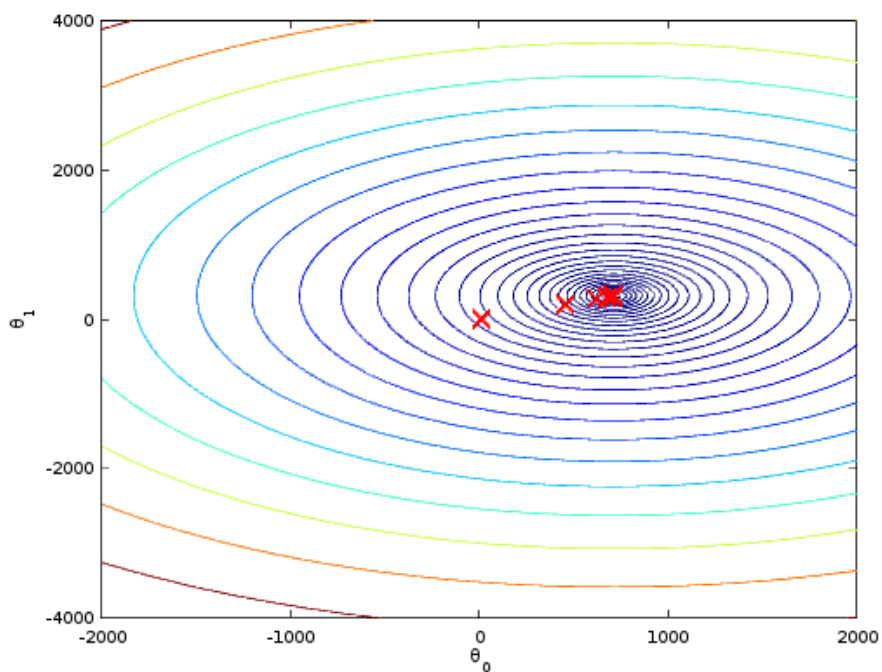
Prvo definiramo  $\theta_1$  in  $\theta_0$  pri nič in jo spustimo skozi algoritem gradientnega spusta, da se nam začnejo  $\theta$  spreminjati glede na izbrane točke. Da, bi algoritem učili, kar se da hitreje, preverimo nekaj možnih  $\alpha$  ali tako imenovane hitrosti učenja (*Slika 1*).





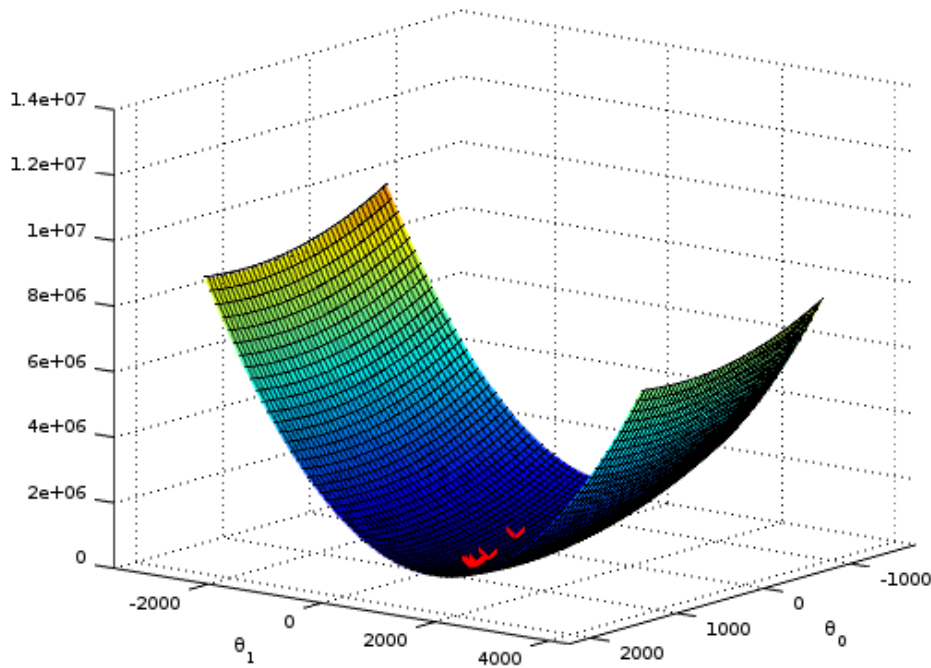
Slika 1: Prikaz različnih hitrosti učenja

Iz grafa je razvidno, da najbolj primer korak učenja je  $\alpha = 0.1$ , kjer se računalnik uči ne prehitro ali ne prepočasi. Da pride do svojega minimuma algoritem potrebuje približno 40 ponovitev gradientnega spusta. Iz (Slika2) se opazi, kako na vsako dekada algoritem počasi potuje proti nižjim vrednosti funkcije v 2d prostoru izrisanih vrednosti, ki jih lahko zavzema  $J(\theta)$ .



Slika 2: Prikaz hitrosti upadanja  $J(\theta)$  proti minimumu

Za prikaz učinkovitosti gradientnega spusta, se lahko uporabi tudi (*Slika3*) graf v 3dimenzionalnem prostoru. Križci so vrednosti  $\theta_0$  in  $\theta_1$ , ki jih ima algoritem pri različnih korakih učenja. Če nam algoritem deluje pravilo, nam vsi križci, kot je razvidno iz obeh slik, potujejo proti najnižji točki, ki jo lahko zasede funkcija  $J(\theta)$  in se tam ustalijo.



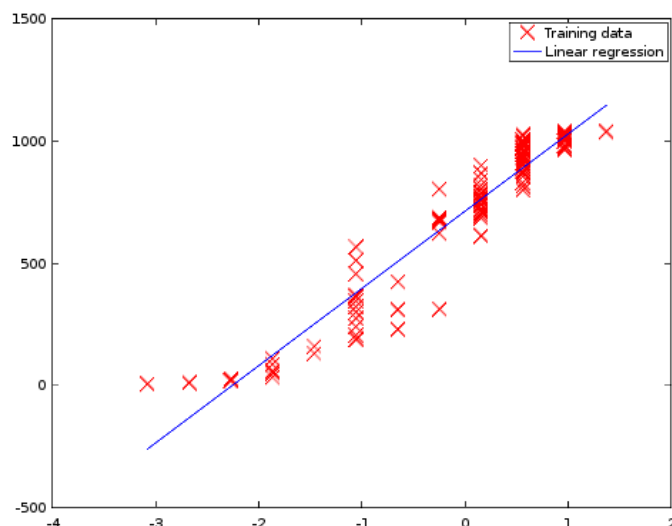
*Slika 3: Prostorski prikaz upadanja theht*

### Validacija modela

Po uspešnem učenju, dobimo nazaj našo hipotetično funkcijo  $h(x) = \theta_0 + \theta_1 x_1$ , s preračunanima  $\theta$ .

$$\theta = \begin{bmatrix} 713.06 \\ 315.84 \end{bmatrix}$$

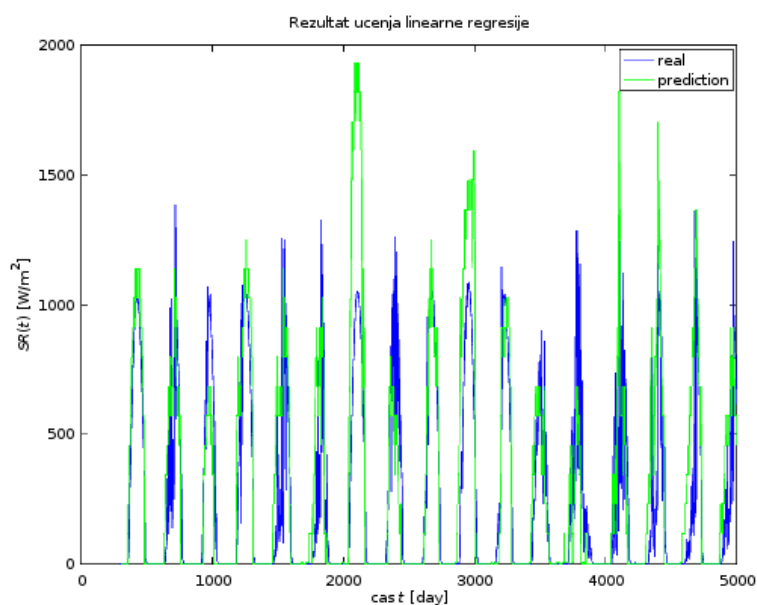
Ta nam omogoči vris premice v graf (*Slika4*). Algoritem je sedaj na podlagi podatkov za učenje izračunal linearno povezavo med temperaturo in sončnim sevanje. Sedaj bo ne glede na kakršno koli podano temperaturo znal poiskati rezultat za iskano  $Y$  vrednost.



Slika 4: Aproksimacijska premica linearne regresije

### Predstavitev in interpretacija rezultatov

Za učinkovitost modela izrišemo prikaz pravih vrednosti in naših predvidenih napovedi, ki nam jih vrne algoritem (*Slika5*). Za vhodne podatke uporabimo naše testne podatke. Kot je očitno razvidno iz grafa, da za večino dni se pojavijo odstopanja in prileganja potekov grafa med pravim in napovedanim se razlikujeta, kar za dvakratno svojo vrednostjo. Omeniti je še potrebno, da za boljši rezultat imamo algoritem prilagojen tako, da nam ne izrisuje negativnih vrednosti. S tem dobimo boljšo obliko iz pričakovanega grafa in fizikalno je tudi nemogoče, da bi sevanje padlo na negativno vrednost.



Slika 5: Rezultat linearne regresije

Iz tega primera opazimo (*Slika5*), da napovedovanje sončnega sevanja ni tako linearno povezano z samo temperaturo. Vreme na sploh je zelo kompleksen rezultat mnogih parametrov, ki pa jih v našem podatkovnem naboru nismo imeli. Algoritem se bi dalo izboljšati, če bi uporabili več parametrov in boljše aproksimacijske funkcije za prilaganje podatkov. Trenutni model ima v povprečju čez 17 dni uspešnost prave napovedi 125.11%, kar pomeni, da je napovedal previsoko od prave napovedi.

## Linearna regresija z več spremenljivkami

Vreme seveda ni odvisno smo od temperature zraka, pač pa je rezultat več parametrov. Da bi rezultat izboljšali, dodamo nove značilke v našo hipotetično funkcijo.  $h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$ .

### Teoretični opis modela

Algoritem smo sedaj ponovno zagnali s to spremembo, da smo vzeli nova dodatna parametra, vlažnost in hitrost vetra. Naša hipotetična funkcija sedaj zgleda takole  $h(x) = \theta_0 + x\theta_1 + x\theta_2 + x\theta_3$ . Opisan postopek zgoraj se ponovi še enkrat in dobimo nove izračunane  $\theta$ :

$$\theta = \begin{bmatrix} 700.2 \\ 286.8 \\ -5.7 \\ 29.2 \end{bmatrix}$$

Za izračun  $\theta$  se nam enačba poenostavi, če imamo  $x$ , zapisan v vektorski obliki kot,

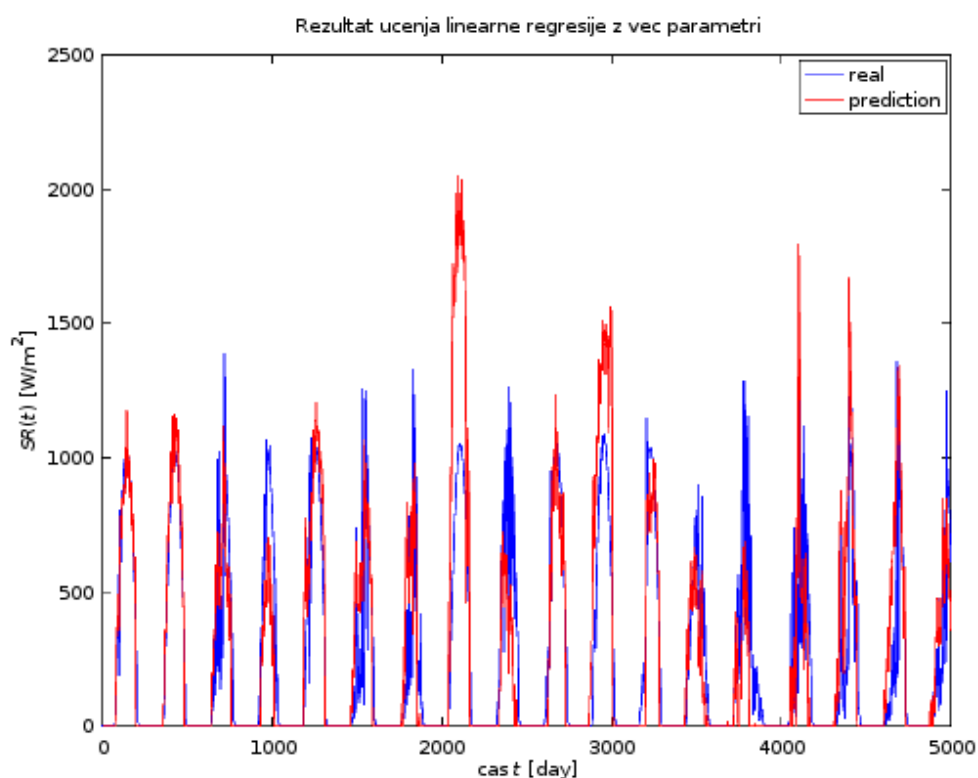
$$X = \begin{bmatrix} -(x^{(1)})^T - \\ -(x^{(2)})^T - \\ \dots \\ -(x^{(m)})^T - \end{bmatrix}$$

,kjer je  $m$  pa število vhodnih podatkov. Enačba je potem zapisana v preprostejši obliki za poljubno število  $\theta$ :

$$\theta = (X^T X)^{-1} X^T \vec{y}$$

## Predstavitev in interpretacija rezultatov

Uspešnost prave od napovedane (*Slika6*) sedaj popravimo na 109.38%. Razvidno je, da algoritem pri zajemu mnogih parametrov zna bolje poiskati povezave med vremenskimi spremembam in sončno energijo. Če bi nam naši podatki nudili več značilk kot so na primer oblačnost, količina padavin, bi nam algoritem že kar dobro znal napovedati željeno veličino.



*Slika 6: Rezultat učenja linearne regresije z več parametri*

## Gradient Boosted Trees

Druga metoda, ki sva jo uporabila pri napovedovanju sončnega sevanja, pa se imenuje Gradient Boosted Trees.

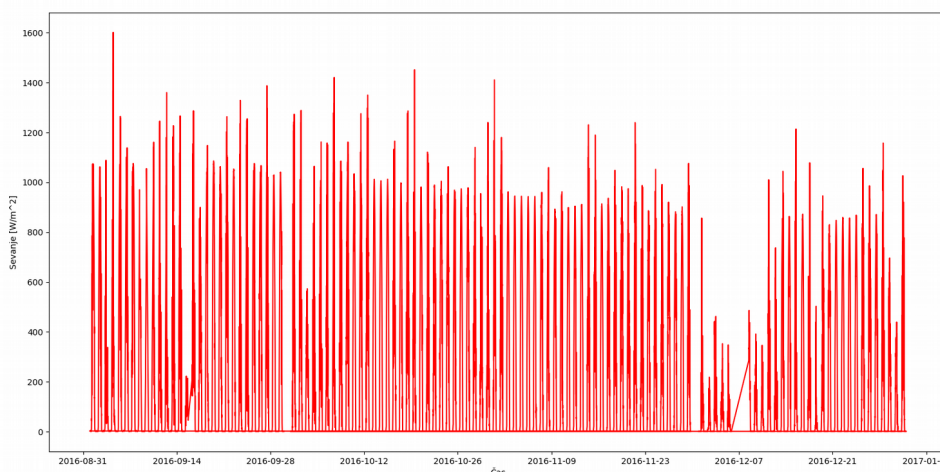
### Opis izbrane metode za gradnjo modela

Za izdelavo modela sva izbrala metodo Gradient Boosted Trees v orodju RapidMiner. Gradient boosting je metoda strojnega učenja za klasifikacijske in regresijske probleme, ki ustvari model v obliki skupine šibkih modelov, običajno odločitvenih dreves. Deluje nekako tako:

Imamo nek nepopoln model  $F(x)$ , ki iz parametrov  $x$  skuša napovedati vrednost  $y$ . Želimo ga izboljšati brez da bi mu odvzeli ali spremenili kak parameter. To storimo tako, da vpeljemo nek nov parameter ocene  $h$ , tako da velja  $F(x_i) + h(x_i) = y_i$  ali  $h(x_i) = y_i - F(x_i)$ . Razliki  $y_i - F(x_i)$  pravimo ostanek ali residual, to so deli, jih trenutni model ne zna dobro napovedati. Naloga parametra  $h$  pa je kompenzacija slabosti obstoječega modela. Ta parameter  $h$  nato priležemo našim podatkom in s tem izboljšamo model. V kolikor nismo zadovoljni z novimi rezultati, ta proces lahko ponavljamo dokler ne dobimo želenega rezultata.

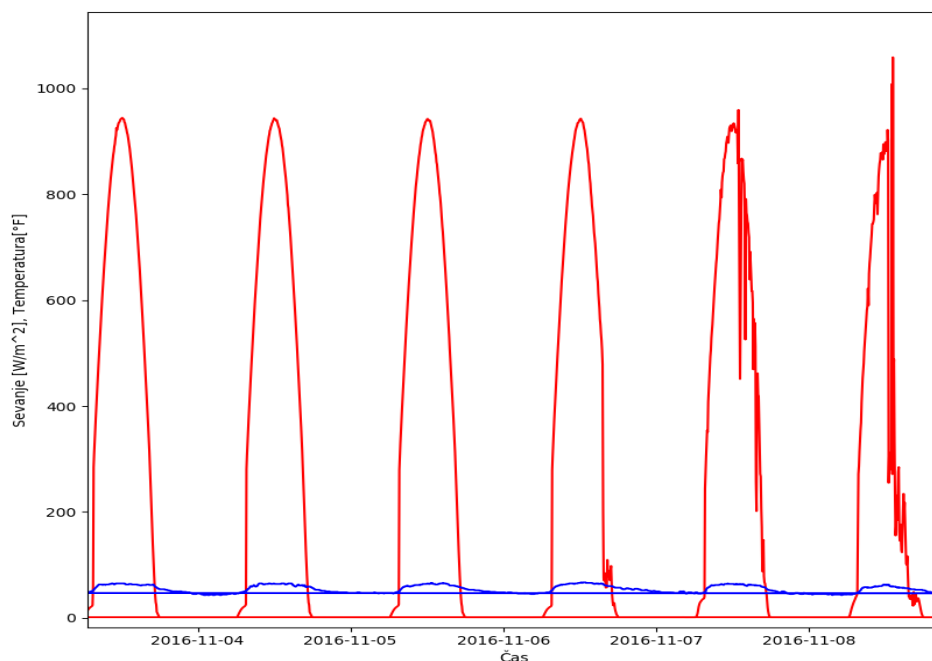
### Opis dela

Najprej sva si podatke iz podatkovne množice izrisala v obliki grafa sevanja v odvisnosti od časa.



Slika 7: potek sevanja celotne podatkovne množice

Iz zgornje slike je opazno da ima vsak dan svoj vrh, vendar so podatki nepregledni, zato naju je zanimal tudi tedenski potek sevanja skupaj s temperaturo.



Slika 8: tedenski potek sevanja s temperaturo

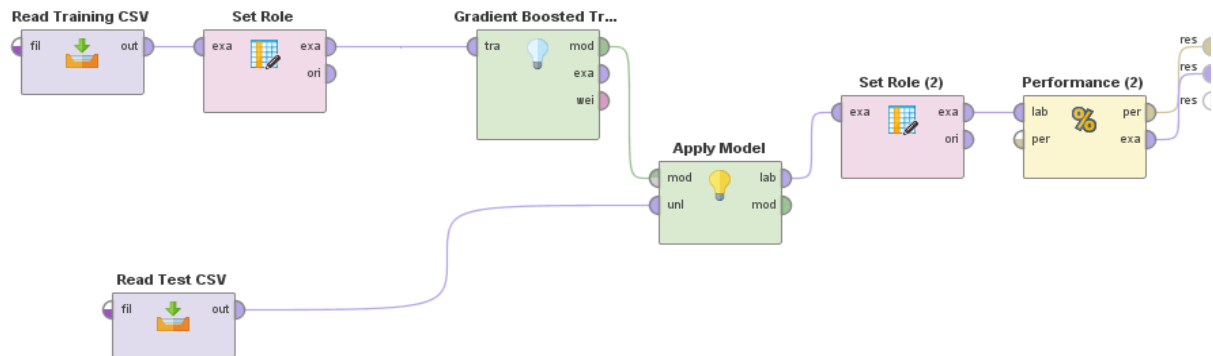
Na zgornji sliki se vidi jasna povezava med sevanjem in temperaturo, za kar meniva da je ključna značilka pri napovedovanju sevanja.

### **Predobdelava podatkov**

Metodo Gradient Boosted Trees sva zaradi enostavnosti uporabe implementirala v orodju RapidMiner. Podatkovno množico sva razdelila na dva dela, učno in testno množico. Iz prve slike se vidi, da manjkajo podatki ob koncu novembra, in nekje med 8. in 10. decembrom, zato sva kot učno množico vzela vse podatke od začetka, do 29. novembra, in testno množico vse podatke od 10. decembra do konca. Ker dobljeni rezultati prvega modela se niso zdeli najboljši, pa je pri drugem modelu izbran način naključnega deljenja učne in testne množice v razmerju 70% za učno in 30% za testno množico.

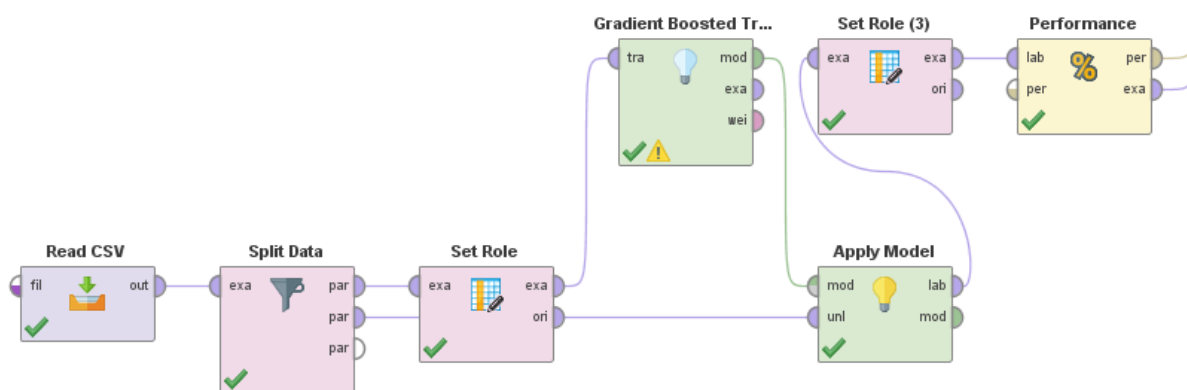
## Gradnja modela

V RapidMinerju začnemo nov proces, in z operatorjem "Read CSV" naložimo učno in testno množico. Ker napovedujemo sevanje, pripišemo z operatorjem "Set Role" značilki "Radiation" vlogo "label". Nato povežemo našo testno množico z operatorjem "Gradient Boosted Tree" in vse skupaj z operatorjem "Apply Model". Na koncu je tu še operator "Performance", v katerem je nastavljeno, da vrne vrednost kvadrirane korelacije, ali  $R^2$ .



Slika 9: gradnja prvega modela v orodju RapidMiner

Drugi model pa je zgrajen tako, da celotno podatkovno množico naložimo z operatorjem "Read CSV" in potem z operatorjem "Split Data" razdelimo podatke tako da podamo razmerje delitev in izberemo način "shuffled sampling". Naprej model poteka enako kot prejšnji.



Slika 10: gradnja drugega modela v orodju RapidMiner

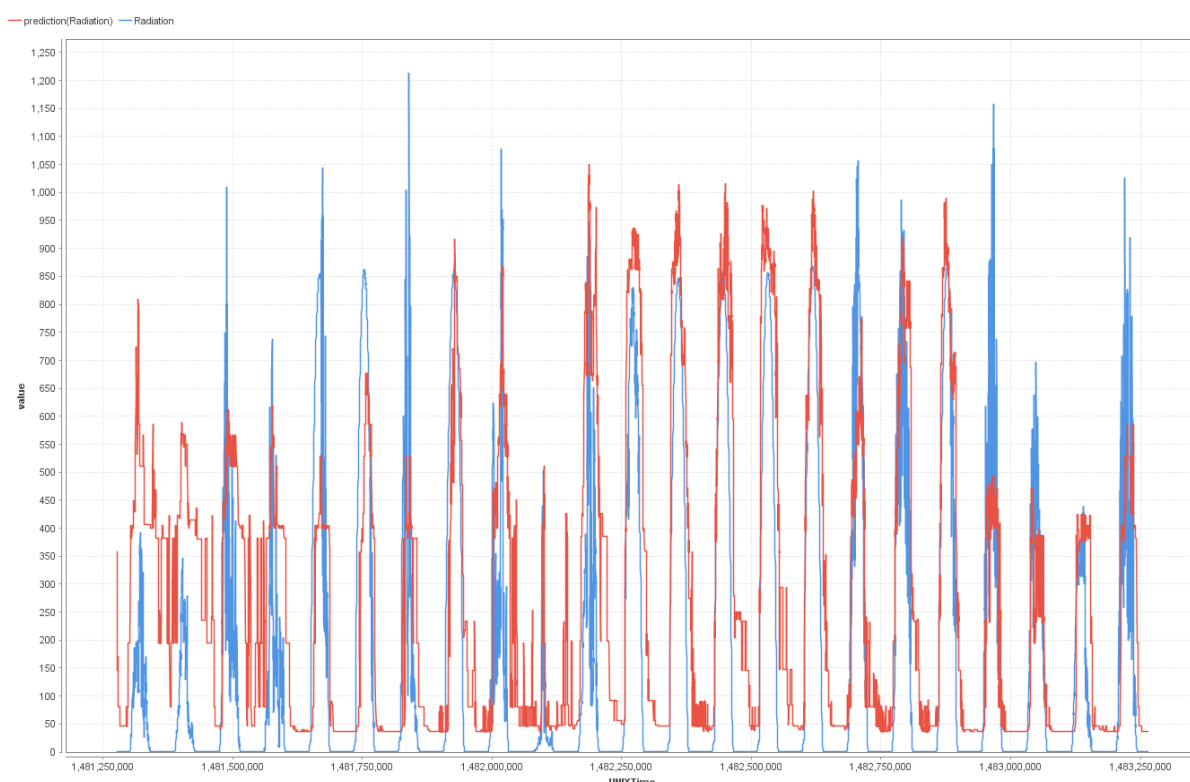


## Validacija modela

Model Gradient Boosted Trees sva validirala z operatorjem Performance (Regression), ki je namenjen za regresijske modele. V prvem primeru, ko je bila množica razdeljena linearno, je bil rezultat  $R^2 = 58.4\%$ . Ta rezultat sva skušala izboljšati, zato sva ubrala drug pristop za izbiro podatkovne in testne množice. Drugi model, ko sta učna in testna množica naključno izbrani se izkaže za boljšega, pri tem pristopu je  $R^2 = 83.7\%$ .

## Predstavitev in interpretacija rezultatov

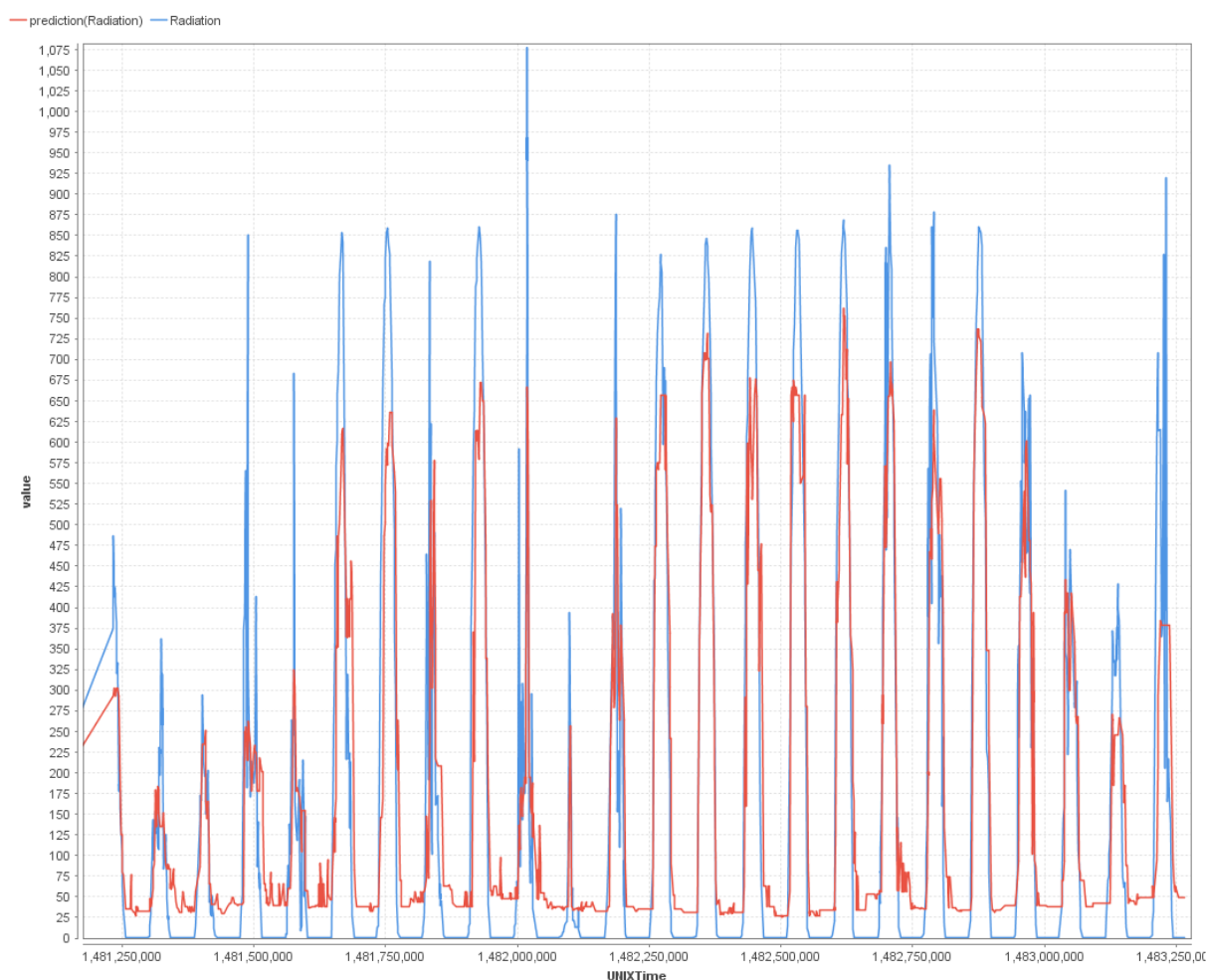
Rezultati so predstavljeni v obliki grafa sevanja v odvisnosti od časa, zraven pa je tudi prikazan graf dejanske vrednosti sevanja, zato da lažje vidimo odstopanja.



Slika 11: graf napovedanega in dejanskega sevanja pri linearnem deljenju podatkov

Zgornja slika kaže, da model nekako zajame dnevna nihanja, vendar pa lahko močno odstopa, še posebej če se vreme naglo spreminja iz dneva v dan, medtem ko relativno dobro napoveduje, ko je vreme nekaj dni zapored stabilno.

Pri naključni izbiri učne in testne množice so napovedane vrednosti veliko boljše, vendar kljub temu ne zadenejo višine sevanja ponoči, ko model napoveduje rahlo previsoko in podnevi ko so vrednosti nekoliko nizke. Model pa še vedno načeloma lahko služi kot v redu pokazatelj kateri dnevi imajo lepo vreme in kateri malo manj.



*Slika 12: graf napovedi sevanja pri naključni izbiri testne in učne množice*

## **Zaključek**

Predstavljene metode v projektu nam podajo rezultate, ki pa so za implementacijo v praksi pomanjkljive. Če bi hoteli vzpostaviti elektroenergetski sistem, kjer bi porabnikom nudili le fotoelektrično energijo, bi bili problemi zaradi netočnosti napovedovanja pridobljene energije. Dogajalo bi se, da bi bili nepripravljeni na drastične spremembe v vremenu, ki jih je s tem opisanimi metodami nemogoče napovedati. V najslabšem primeru, elektroenergetski sistem nebi dovedel dovolj energije uporabnikom, kar bi nosilo posledice v gospodarstvu.

Metode se bi dalo seveda tudi izboljšati. Eden od načinov bi bil, da bi beležili več parametrov vremena, kot je npr. oblačnost, količina padavin,... in z njimi poizkušali dobiti boljše hipotetične funkcije. Pri metodah z regresijami, se bi dalo funkcijo prilagati z drugimi oblikami, ne samo z premico.

Obstajajo že obstoječe metode dnevnega napovedovanja sončne energije, ki so tudi v določenih dnevih več kot 90% uspešne. Učenja teh metod je zajemalo večje baze podatkov iz več vremenskih postaj in učene so bile na principu Deep learninga. Z boljšimi algoritmi napovedovanja, nam da upanje, da v prihodnosti bo postala sončna elektrarna zanesljiv vir električne energije.

## Viri

- Ritchie Ng. Machine learning: One Variable Linear Regression. [Dec 29, 2017]  
Dostopno na: (<http://www.ritchieng.com/one-variable-linear-regression/>)
- CS229. Syllabus and Course Schedule: Supervised Learning [Dec 29, 2017]  
Dostopno na: (<http://cs229.stanford.edu/syllabus.html>)
- Andrew Ng. OpenClassroom: Linear Regression [Dec 29, 2017]  
Dostopno na:  
(<http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning>)
- DataBase: OpenDatabase. Solar Radiation Prediction [Dec 29, 2017]  
Dostopno na: (<https://www.kaggle.com/dronio/SolarEnergy>)
- Cheng Li. A Gentle Introduction to Gradient Boosting [Dec 30, 2017]  
Dostopno na: ([http://www.chengli.io/tutorials/gradient\\_boosting.pdf](http://www.chengli.io/tutorials/gradient_boosting.pdf))
- Gilbert Wasserman. Predicting Solar Radiation [Dec 30, 2017]  
Dostopno na: (<https://www.kaggle.com/gilwassermann/solar-radiation-prediction-nb>)
- [https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/gradient\\_boosted\\_trees.html](https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/gradient_boosted_trees.html) [Dec 30, 2017]
- [https://docs.rapidminer.com/latest/studio/operators/validation/performance/predictive/performance\\_regression.html](https://docs.rapidminer.com/latest/studio/operators/validation/performance/predictive/performance_regression.html) [Dec 30, 2017]
- [https://en.wikipedia.org/wiki/Gradient\\_boosting](https://en.wikipedia.org/wiki/Gradient_boosting) [Dec 30, 2017]