

**PREDVIĐANJE POTROŠNJE ELEKTRIČNE ENERGIJE KORIŠĆENJEM LGBM ALGORITMA U ML.NET-U I PYTHON-U****ELECTRICITY CONSUMPTION PREDICTION USING LGBM ALGORITHM IN ML.NET AND PYTHON**

Nemanja Simić, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** – Predviđanjem potrošnje električne energije, elektrodistribucije mogu efikasnije da planiraju režim upotrebe generatorskih postrojenja, redovna održavanja elemenata mreže kao i potencijalnu trgovinu na marketu električne energije. Algoritmi mašinskog učenja mogu poslužiti kao alat za precizno predviđanje potrošnje u elektroenergetskim sistemima. Kroz ovaj rad implementirana su dva softverska rešenja za prognozu potrošnje električne energije na osnovu podataka vremenskih prilika koristeći LGBM algoritam u ML.NET-u i Pythonu, dok su rezultati predikcija opisani i analizirani.

**Ključne reči:** LGBM, ML.NET, Python, Mašinsko učenje, Algoritam

**Abstract** – By predicting electricity consumption, electricity distribution companies can more efficiently plan the mode of use of generating plants, regular maintenance of network elements, as well as potential trade on the electricity market. Machine learning algorithms can serve as a tool for accurate consumption forecasting in power systems. Through this work, two software solutions for forecasting electricity consumption based on weather data were implemented using the LGBM algorithm in ML.NET and Python, while the prediction results were described and analyzed.

**Keywords:** LGBM, ML.NET, Python, Machine Learning, Algorithm

**1. UVOD**

Stabilan elektroenergetski sistem odlikuje, između ostalog, pouzdana proizvodnja električne energije i redovno održavanje postrojenja. Precizna prognoza potrošnje unapređuje proces planiranja upotrebe i održavanja proizvodnih postrojenja, kao i kupovine električne energije. Stoga, počeo je razvoj raznih modela mašinskog učenja koji predviđaju potrošnju električne energije u stambenim i poslovnim zgradama koristeći karakteristike kao što su podaci o vremenskoj prognozi ili istorijski podaci o računima za električnu energiju [1]. U radu [2], Kolter i Ferreira koriste mesečne račune električne energije i gasa, kao i karakteristike zgrade kako bi modelovali potrošnju. Raspostranjena upotreba mašinskog učenja dovela je do razvoja različitih tehnologija u

ovom polju. U programskom jeziku *Python* napisane su popularne biblioteke za obradu i analizu podataka *Numpy* i *Pandas*, zajedno sa velikim brojem algoritama mašinskog učenja različitih namena, od kojih je većina deo *scikit-learn* paketa. Tehnološki gigant *Microsoft*, prateći trendove tržišta, C# programerima nudi mogućnost upotrebe ovakvih algoritama pomoću *ML.NET* paketa koji je razvijen u kao deo *.NET* radnog okvira-a.

**2. TEORIJSKE OSNOVE****2.1. Prognoza potrošnje u elektroenergetskim sistemima**

Predviđanje potrošnje električne energije ima za cilj da obezbedi neprekidno snabdevanje i smanji operativne troškove preduzeća [3]. Predstavlja proračun koji pruža kompanijama jasniju sliku sistema u budućnosti i daje indikatore o potrebnim intervencijama. Poznavanje stanja mreže u određenom trenutku u budućnosti pruža uvid u potencijalne probleme sistema i olakšava donošenje biznis odluka [4]. Ovakve informacije su od velikog značaja prilikom kupovine i prodaje električne energije [5], kao i tokom planiranja radova i proširenja infrastrukture mreže [6]. Na osnovu vremenske skale, prognoza potrošnje električne energije se klasifikuje na:

- Veoma kratkoročno predviđanje potrošnje (*VSTLF*) koristi se za vremenski period od 10 do 30 minuta [7].
- Kratkoročno predviđanje potrošnje (*STLF*) koristi se za vremenski period od nekoliko sati, dana ili jedne nedelje [8].
- Srednjoročno predviđanje potrošnje (*MTLF*) koristi se za vremenski period od jedne nedelje do jedne godine [9].
- Dugoročno predviđanje potrošnje (*LTLF*) koristi se za vremenski period od jedne do 20 godina [10].

**2.2. Mašinsko učenje**

Arthur Samuel, još 1959. godine, mašinsko učenje definiše kao oblast studija koja računarima daje mogućnost učenja koje nije eksplicitno programirano. Sposobnost generisanja znanja na osnovu od ranije poznatih podataka i korišćenja tog znanja u pronalaženju odgovora na pitanja o novim, do tada nepoznatim, podacima.

Tom Mitchell 1998. godine iznosi moderniju definiciju: „Kaže se da kompjuterski program uči iz iskustva E u odnosu na neku klasu zadatka T i meru učinka P, ako se njegov učinak na zadacima T, mereći sa P, poboljšava iskustvom E.“

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Sebastijan Stojan, docent.

Drugim rečima, mašinsko učenje predstavlja podoblast veštačke inteligencije u kojoj algoritmi imaju sposobnost iterativnog otkrivanja pravila među podacima iz kojih donose konačne zaključke spremne za korišćenje u obliku modela. Proces iterativnog obrađivanja ulaznih podataka od strane algoritma naziva se treniranje modela. Ovo nije egzaktna nauka, što znači ne postoji jedno tačno rešenje niti jedan savršeno istreniran model. Više različitih algoritama se mogu koristiti u rešavanju istog problema, ne postoji jedan algoritam koji odgovara samo jednom problemu. Ono što će odlučiti koji će se algoritam koristiti u konačnom rešenju jeste sama preciznost konačnog zaključka tog algoritma, odnosno istreniranog modela. Na preciznost predikcije može uticati sam oblik podataka, količina i kompleksnost, ali i priroda algoritma kao i podešavanje njegovih parametara.

Prema načinu obučavanja modela, prepoznamo tri tipa mašinskog učenja:

1. Nadgledano učenje (eng. *supervised learning*) podrazumeva skup ulaznih podataka ( $x_1, x_2, \dots, x_n$ ) i skup njima odgovarajućih izlaznih podataka ( $y_1, y_2, \dots, y_n$ ). Drugim rečima, model obučavamo trening podacima čija tražena prognoza je poznata unapred algoritmu i na osnovu koje se traži logička veza sa ulaznim podacima. Algoritmi sa ovakvim načinom učenja se dele na regresione i klasifikacione.
2. Nenadgledano učenje (eng. *unsupervised learning*) pruža mogućnost izračunavanja izlaznih podataka kada su poznati samo ulazni. Kod ovakvog pristupa rešavanja problema algoritam ima zadatak otkrivanja zakonitosti između ulaznih podataka. Zaključci koje donosi algoritam su do tog trenutka nepoznati i ne mogu se znati efekti ulaznih podataka pre obučavanja. Algoritmi se dele na algoritme za klasterovanje i za pronalazak novih struktura.
3. Učenje podsticajem (eng. *reinforcement learning*) je oblast mašinskog učenja u kojoj se algoritam „nagrađuje“ kada se u iteraciji izračuna dovoljno precizna prognoza, dok se u suprotnom „kažnjava“. Metafora nagrade i kazne ogleda se u jednostavnom dodavanju i oduzimanju od sume za koju algoritam teži da bude što veća. Na taj način algoritam podešava svoje parametre kroz cikluse sa ciljem što veće preciznosti.

### 2.2.1 Stablo odluke

Stablo (eng. *Tree*) je oblik strukture podataka u računarstvu. Podaci su uvezani u hijerarhiju imitirajući obrnuto drvo, koren je gore dok se grane i listovi razvijaju ka dole. Osnovni elementi ove strukture jesu čvorovi i grane. Prvi čvor u stablu se naziva koren, dok čvorovi ispod kojih ne postoji grananje se nazivaju listovi.

Stablo odluke (eng. *Decision tree*) [11] je struktura podataka u čiji čvorovi predstavljaju odluke, svaka grana je odluka ili pravilo, dok je svaki list konačan ishod. Smatra se kao jedan od najpopularnijih pristupa prikaza klasifikacije. Pronalazi upotrebu u raznim istraživačkim disciplinama kao što su statistika, mašinsko učenje, prepoznavanje obrazaca kao i rudarenju podataka (eng. *data mining*) [12]. Čitav proces prolaska kroz stablo odluke podseća na način ljudskog odlučivanja i lako ga je razumeti.

Primena ove strukture je popularna među algoritmima nagledanog učenja. Svoju upotrebu pronalazi i u regresiji i u klasifikaciji. Ovakvi algoritmi imaju mnogo „šta ako“ ograničenja i ovo stablo im pomaže tokom izvršavanja.

### 2.2.2 Light Gradient Boosting Machine

*Light Gradient Boosting Machine (LGBM)* [13] je algoritam mašinskog učenja zasnovan na stablu odluke. „Laka“ varijanta *gradient boosting machine* algoritma koja je preciznija i boljih performansi od ostalih iz ove porodice algoritama. Svoj epitet je dobio zahvaljujući smanjenoj potrošnji memorije tokom izvršavanja i brzim procesom obučavanja modela. Glavna razlika između ovog i ostalih *Gradient Boosting Machine (GBM)* algoritama jeste mogućnost vertikalnog širenja stabla, što znači gradi nove listove. Izlaz algoritma je maksimalno efektivan list sa najmanjom greškom. Pored brzine, preciznosti, ekonomičnog trošenja memorije, ovaj algoritam karakteriše i mogućnost paralelizacije, distribucije i učenja koristeći resurse grafičkog procesora. Takođe, pogodan je za rešavanje problema sa velikom količinom podataka. Sama implementacija algoritma je jednostavna za korišćenje tokom softverskog razvoja.

## 3. IMPLEMENTACIJA REŠENJA

### 3.1. Priprema podataka

U ovom radu se rešava problem predviđanja potrošnje električne energije na osnovu podataka vremenskih prilika. Stoga, prvi koraci su pronalazanje javno dostupnih pouzdanih izvora, analiza, obrada i spajanje dva seta podataka.

#### 3.1.1 Potrošnja električne energije

Nakon analize javno dostupnih izvora odabran je set podataka koji predstavlja potrošnju električne energije u Njujorku. Ovaj set je dostupan na veb sajtu *NYC Open Data* [14] generisan od strane uprave za stambena pitanja grada Njujorka (eng. *New York City Housing Authority (NYCHA)*). Sadrži različite informacije vezane za potrošnju električne energije na području grada u periodu od 2010. do 2021. godine. U tabeli se nalazi 362,630 zapisa koji su opisani sa 27 atributa.

#### 3.1.2 Vremenske prilike

Kao izvor za skup podataka o vremenskim prilikama korišćen je veb sajt nacionalnog centra za informacije o životnoj sredini *National Oceanic and Atmospheric Administration (NOAA)* [15], američka naučna regulatorna agencija u okviru Ministarstva trgovina Sjedinjenih Američkih Država. Podaci su dostupni u obliku prosečnih mesečnih vrednosti za odabrani grad.

### 3.2. Model mašinskog učenja u C#-u

*Visual Studio 2019* nudi proces kreiranja modela mašinskog učenja sa korisničkim interfejsom korišćenjem *ML.NET Model Builder*-a [16]. Korišćenjem ove funkcionalnosti moguće je učitati obrađene podatke, istrenirati model sa algoritmom koji daje najpreciznije rezultate, testirati i isporučiti rešenje bez kucanja koda, u šest koraka:

1. Scenario – Odabir tipa problema, čime se automatski određuje grupa adekvatnih algoritama.

2. Okruženje – Odabir mašine na kojoj se izvršava obučavanje, lokalna ili mašina na Azure-u.
3. Podaci – Učitavanje podataka iz fajla ili sa povezanog SQL servera. Odabir kolona za ignorisanje tokom treniranja kao i kolone čija se vrednost prognozira.
4. Treniranje modela – Grupa algoritama, adekvatnih za odabrani scenario, se iterativno treniraju pri čemu se nakon svake iteracije beleži preciznost modela. Tokom ovog repetitivnog procesa parametri algoritama se automatski podešavaju sa ciljem da svaki sledeći ciklus bude precizniji od prethodnog.
5. Evaluacija modela – Manuelno testiranje preciznosti najefikasnijeg modela iz prethodnog koraka.
6. Generisanje koda – Gotovo rešenje se može izgenerisati u obliku konzolne aplikacije ili kao Web API.

### 3.3. Model mašinskog učenja u Python-u

Python model kreiran je uz pomoć razvojnog okruženja Jupyter Notebook i metode `lightgbm.LGBMRegressor` [17] u okviru `scikit-learn` biblioteke. Parametri algoritama zasnovanih na strukturi drveta (eng. *tree based*), kao što je i *LGBM*, se dele u četiri glavne kategorije:

- Parametri koji utiču na strukturu stabla (`num_leaves` i `max_depth`)
- Parametri koji utiču na brzinu učenja (`num_threads` i `use_gpu`)
- Parametri koji utiču na preciznost (`n_estimators` i `learning_rate`)
- Parametri koji štite od overfit-a (`lambda_1` i `lambda_2`)

### 4. REZULTATI TESTIRANJA

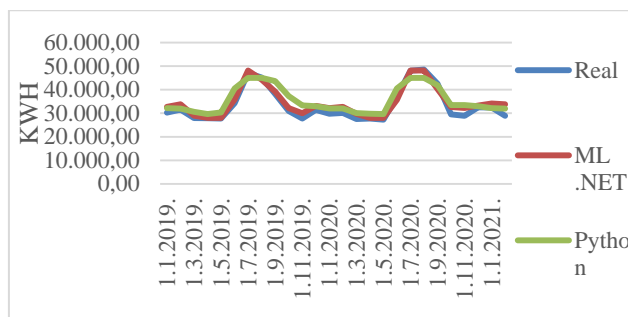
Prilikom kreiranja modela mašinskog učenja i samog testiranja rešenja, u ovom radu korišćeno je isto radno okruženje za Python i ML.NET kako bi poređenje dve implementacije bilo verodostojno, čije se karakteristike mogu videti u tabeli:

Tabela 1. Karakteristike testnog okruženja korišćenog u ovom istraživanju.

OS	Windows 10 Pro 21H2, 64-bit
Procesor	AMD Ryzen 5 2600 Six-Core 3.40 GHz
RAM	16.0 GB

Nakon završetka procesa obučavanja, oba modela mašinskog učenja su izvršavana nad istim setom ulaznih podataka, a njihovi rezultati upoređivani i analizirani. Podaci koji se nisu našli u trening setu, period od januara 2019. do februara 2021. godine, a za koje su poznate mesečne prosečne vrednosti potrošnje zgrada, su korišćeni u testiranju modela. Mesečne prognoze oba modela zajedno sa realno izmerenom potrošnjom se mogu videti na slici 1.

Maksimalna i minimalna odstupanja u prognozi izračunata u KWH a potom predstavljena u % za taj mesec za oba modela se mogu videti tabeli 2.

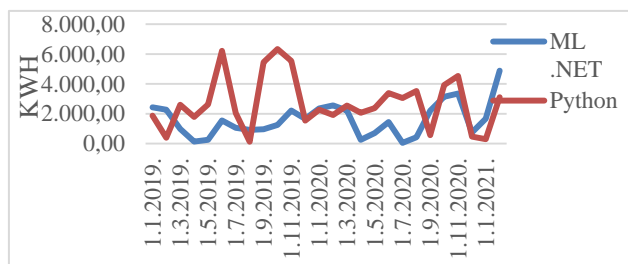


Slika 1. Vizuelni prikaz prognoziranih i realnih vrednosti potrošnje.

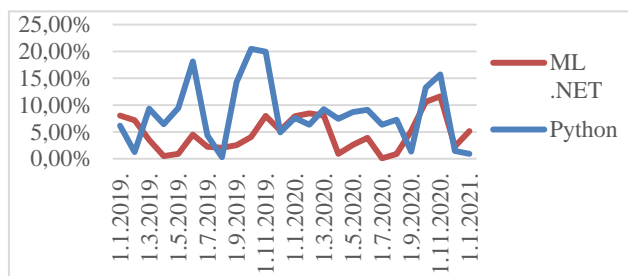
Tabela 2. Maksimalna i minimalna odstupanja u prognozi oba modela.

	MAX KWH	MAX %	MIN KWH	MIN %
ML.NET	4885.79	16.92	43.04	0.09
Python	6334.9	20.47	128	0.28

Preciznosti modela za svaki mesec se razlikuju. Drugim rečima, ne postoji fiksno odstupanje od tražene vrednosti za svaku prognoziranu već to u mnogome zavisi od ulaznih podataka i analogije koju je napravio algoritam na osnovu njih. Na slici 2 se mogu videti odstupanja u KWH oba modela za svaki prognozirani mesec, dok se na slici 3 može videti to isto odstupanje u procentima.



Slika 2. Odstupanja prognozirane potrošnje oba modela u KWH.



Slika 3. Odstupanja prognozirane potrošnje oba modela u %.

Matematički izračunata preciznost prognoza može se videti u tabeli 3.

Tabela 3. Metrike preciznosti prognoza dva modela.

	MAPE	ME	MAE	MPE	CORR
ML.NET	0.05	1218.34	1602.77	0.042	0.97
Python	0.08	1974.66	2713.95	0.069	0.93

Tumačenjem ovih vrednosti, izračunatim statističkim formulama, utvrđuje se da li prognoze mogu imati upotrebnost u rešavanju konkretnog problema.

## 5. ZAKLJUČAK

U ovom radu korišćene su dve implementacije *LGBM* algoritma sa ciljem predviđanja prosečne potrošnje zgrade na Menhetnu na osnovu podataka vremenskih prilika. Model kreiran uz pomoć *ML.NET*-a pokazao je za 3% preciznije rezultate od modela implementiranog u *Python*-u i na taj način dokazao da algoritmi pisani u *C#* jesu konkurencija na tržištu.

Važan deo procesa stvaranja modela mašinskog učenja jeste priprema podataka. Veliki broj praktičnih i efikasnih biblioteka napisanih u *Python*-u čine ovaj jezik liderom u obradi podataka (eng. *Data processing*).

Proces kreiranja modela u *C#* je znatno olakšan korišćenjem *Model Builder*-a, ali njegovo glavno ograničenje jeste to što na raspolaganju ima samo one algoritme koje je *Microsoft* podržao. Drugim rečima *LGBM* algoritam se za potrebe ovog rada jeste pokazao preciznijim kada se koristi njegova *C#* implemetacija, međutim *Python* ima veći broj podržanih algoritama i postoji mogućnost da korišćenjem nekih od njih koji nisu podržani u *.NET*-u imaju još preciznije rezultate. Takođe, radno okruženje *Jupyter Notebook* koje se koristi tokom obrade podataka i kreiranjem modela u *Python*-u je jednostavnije i lakše za korišćenje od *Visual Studio*-a, što ovom jeziku daje prednost.

Dalje usavršavanje rešenja bi moglo ići u pravcu novih tehnika za podešavanje parametara. Sve češće se može videti upotreba algoritama veštačke inteligencije baziranih na učenju nagradom i kaznom (eng. *reinforcement learning*) u podešavanju parametara drugih algoritama. Na taj način mašinskim učenjem stvaramo novi što precizniji model mašinskog učenja. Algoritam kroz iteracije podešava različite vrednosti ulaznih promenljivih, i u zavisnosti od preciznosti novodobijenog modela biva nagrađen ili kažnjen. Takođe, *Python* zajednica nudi širok spektar algoritama regresije, gde bi neki od njih mogao dati preciznije rezultate od *LGBM* implementiranog u *.NET*-u. Konačno, *Python* i *.NET* možda jesu popularne i velike zajednice ali nisu jedine. Istraživanja treba proširivati novim tehnologijama, davati šansu novim algoritmima.

## 6. LITERATURA

- [1] Zhao, Hai-xiang & Magoulès, Frédéric, „A review on the prediction of building energy consumption“, *Renewable and Sustainable Energy Reviews*. 16. 3586-3592. 10.1016/j.rser.2012.02.049.
- [2] J. Kolter, J. Ferreira, „A Large-Scale Study on Predicting and Contextualizing Building Energy Usage“, *AAAI*, vol. 25, no. 1, pp. 1349-1356, Aug. 2011.
- [3] Slobodan Ilić, „Kratkoročno predviđanje potrošnje električne energije u velikim elektroenergetskim sistemima“, Fakultet tehničkih nauka, Univerzitet u Novom Sadu.
- [4] Sebastijan Stoja, „Arhitektura softverskog sistema za elektroenergetske proračune zasnovana na mikroservisima“, Fakultet tehničkih nauka, Univerzitet u Novom Sadu.

- [5] Gama João, Rodrigues Pedro, „Stream-Based Electricity Load Forecast“, 446-453. 10.1007/978-3-540-74976-9\_45, 2007.
- [6] Khuntia, Swasti R. & Rueda, José & Meijden, Mart, „Forecasting the Load of Electrical Power Systems in Mid- and Long-term Horizons - A Review“, *IET Generation Transmission & Distribution*. 10, 2016. 10.1049/iet-gtd.2016.0340.
- [7] J. W. Taylor, „An evaluation of methods for very short-term load forecasting using minute-by-minute British data“, *International Journal of Forecasting*, vol. 24, no. 4, pp. 645-658, 2008.
- [8] G. Gross, F.D. Galiana, „Short-term load forecasting“, *Proceedings of the IEEE*, vol. 75, no.12, pp. 1558-1572, 1987.
- [9] N. Amjady, F. Keynia, „Mid-term load forecasting of power systems by a new prediction method“, *Energy Conversion and Management*, vol. 49, no. 10, pp. 2678-2687, 2008.
- [10] H. Daneshi, M. Shahidehpour, A. L. Choobbar, „Long-term load forecasting in electricity market“, *IEEE International Conference on Electro/Information Technology, USA*, 2008.
- [11] Harsh H. Patel, Purvi Prajapati, „Study and Analysis of Decision Tree Base Classification Algorithms“, Dept. Of Information Technology, CSPIT, Charotar University of Science and Technology, Changa, Gujarat, India.
- [12] Lior Rokach, Oded Maimon, „The Data Mining and Knowledge Discovery Handbook Chapter: Decision Trees“, DOI:10.1007/-387-25465-X\_9.
- [13] <https://lightgbm.readthedocs.io/en/v3.3.2/> (pristupljeno u martu 2023.)
- [14] <https://data.cityofnewyork.us/Housing-Development/Electric-Consumption-And-Cost-2010-April-2020-/jr24-e7cr> (pristupljeno u martu 2023.)
- [15] <https://www.ncdc.noaa.gov/cdo-web/datasets/GSOM/locations/CITY:US360019/detail> (pristupljeno u martu 2023.)
- [16] <https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet/model-builder> (pristupljeno u martu 2023.)
- [17] <https://lightgbm.readthedocs.io/en/latest/index.html> (pristupljeno u martu 2023.)

### Kratka biografija:



**Nemanja Simić** rođen je u Šapcu 1996. godine. Osnovne akademske studije na Fakultetu tehničkih nauka Univerziteta u Novom Sadu upisao je 2015. godine. Diplomirao je 2019. godine na smeru Primenjeno softversko inženjerstvo i iste godine upisao master akademske studije na istom smeru.