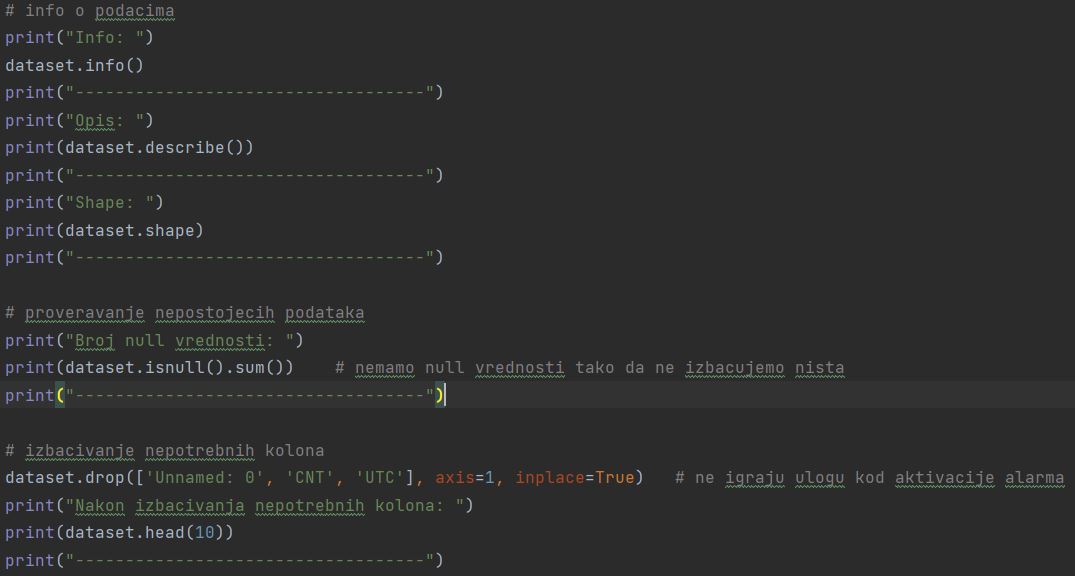
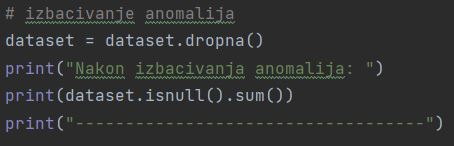
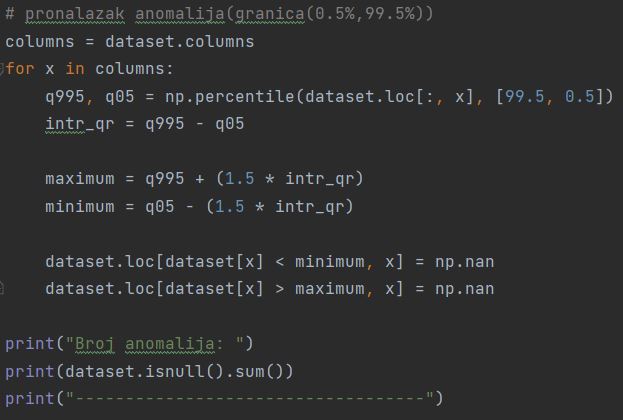
**SMOKE DETECTION**

**-Projekat iz SAUSAU-**

* **Preprocesiranje podataka**

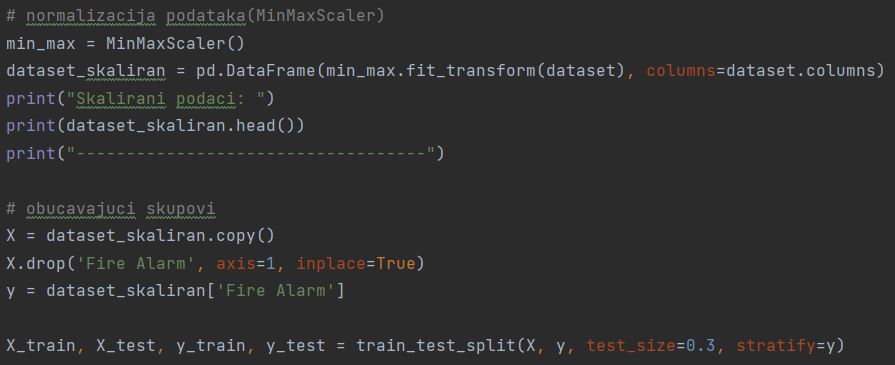


U ovom skupu podataka nemamo string vrednosti tako da nema potrebe vrsiti konverziju. Iz skupa mozemo da zakljucimo da kolone [‘Unnamed’, ’CNT’, ‘UTC’] ne uticu na to da li ce alarm biti aktiviran I mozemo ih izbaciti iz skupa.



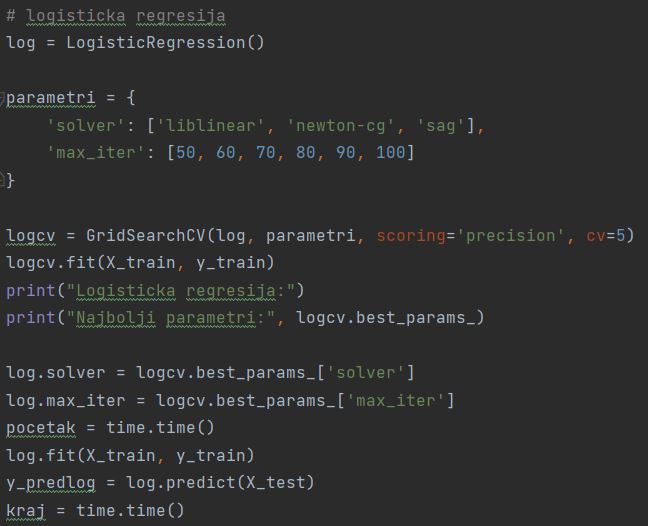
Nakon pronalska anomalija menjamo ih sa nan vrednostima i funkcijom dropna() izbacujemo iz skupa podataka.

* **Eksplorativna analiza**

Algoritmi masinskog ucenja bolje rade sa normalizovanim podacima, pa podatke normalizujemo pomocu funkcije MinMaxScaler(), koja po default vraca vrednosti (0, 1).

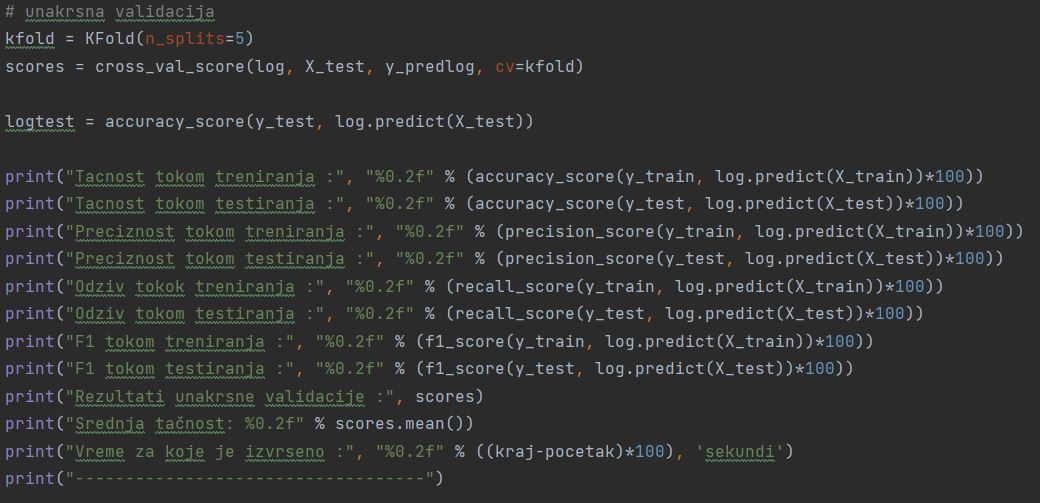


Matricu korelacija pravimo pomocu funkcije corr(), a maska nam u ovom slucaju sluzi radi lepseg prikaza na ekranu tako sto ce prikazati vrednosti ispod glavne dijagonale.

* **Kreiranje modela i podesavanje hiperparametara**

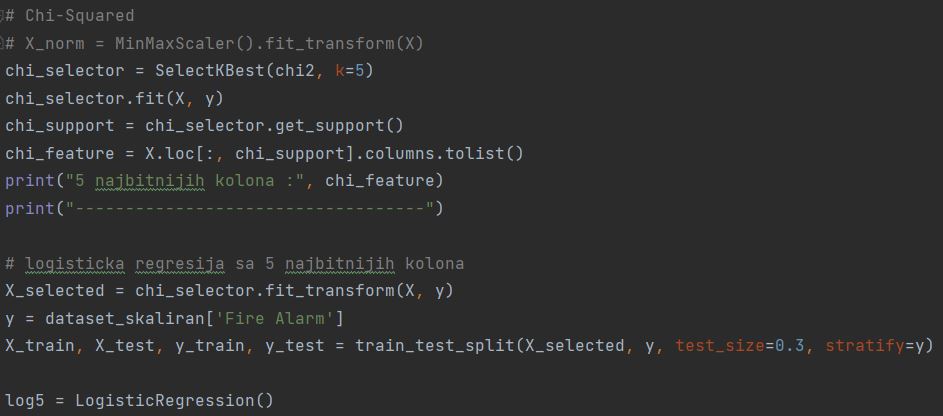
Za model vrsimo podesavanje hiperparametara (u ovom slucaju samo 2). Za scoring je podeseno da bude ‘precision’ jer dobijamo bolje rezultate nego kad je na ‘accuracy’. Kada pronadjemo najbolje parametre dodeljujemo ih nasem modelu i vrsimo fitovanje sa najboljim parametrima. Pomocu funkcije time() mozemo videti koliko je vremena bilo potrebno za fit i predict jednog modela.

U kodu su uradjena 4 modela ali radi preglednosti prikazacemo samo 2 u dokumentaciji.

* **Unakrsna validacija i analiza rezultata**

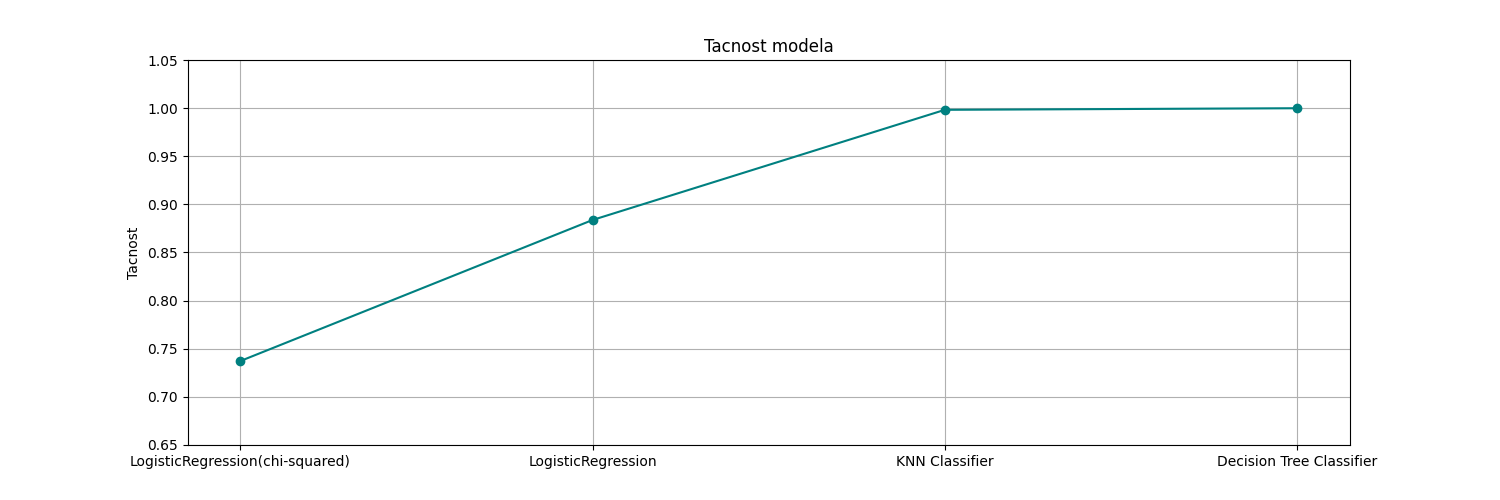
KFold nam omogucava da podelimo trening set na k-manjih delova za validaciju modela. Model se dalje trenira sa k-1 delova gde preostali delovi sluze za validaciju.

Kod ispisa ispisujemo vrednosti i tokom treniranja i tokom testiranja radi detaljnijeg prikaza i pomocu unakrsne validacije proveravamo da li je doslo do overffitinga i radi provere kako ce se model ponasati na novim podacima.

* **Odabir najbitnijih atributa**

Pomocu funkcije SelectKBest() vrsimo selektovanje najboljih k parametara koja na osnovu score\_func=chi2 vraca 5 najbitnijih kolona, koje prosledjujemo novom X, dok se y ne menja.

* **Rezultati**



Na osnovu slike iznad mozemo videti da modeli K najblizih suseda i Stabla odluke imaju veoma veliku tacnost, dok Logisticka regresija ima okej tacnost. Takodje, vazno je spomenuti da je potrebno vreme za izvrsavanje KNN algoritma od 80-95 sekundi, dok je za stablo odluke od 12-21 sekunde. Selektovanjem 5 najvaznijih kolona tacnost Logisticke regresije opada dok brzina izvrsavanja raste (u pocetku od 14-20 sekudni, a nakon selektovanja od 6-10 sekundi).

* **Napomene**

Dodatni modeli(K najblizih suseda, Stablo odlucivanja) su uradjeni u projektu kao i prikaz svih potrebnih plotova i informacija vezanih za svaki model(tacnost, preciznost, odziv…).

**Popov Petar**

**RA 24/2020**

**08.06.2023.**