# Linearna regresija

**Jednostavna linearna regresija**

Je statistički metod koji omogućava proučavanje veze između dve kontinualne promenljive.

Prva promenljiva, x, se naziva **nezavisna promenljiva**. Druga promenljiva, y, se naziva **zavisna promenljiva**.

Jednostavna linearna regresija ima pridev “jednostavna” zato što ima samo jednu nezavisnu promenljivu .  
Postoji i višestruka linearna regresija, koja se odnosni na linearnu regresiju sa više nezavisnih promenljivih.

Pre opisa same linearne regresije, važno je napomenuti kakve veze/zavisnosti između promenljivih su od interesa.

**Deterministička (funkcionalna) zavisnost** nam nije od interesa jer njom možemo odrediti **tačnu** vrednost zavisne promenljive na osnovu vrednosti nezavise promenljive .

Za svaku od ovih determinističkih zavisnosti, jednačina tačno opisuje odnos dve promenljive. Ne razmatramo determinističke veze, već **statističku zavisnost**, gde veza između dve promenljive nije savršeno tačna.

**Koja je "najpogodnija linija"?**

Pošto je neophodno pronaći neku linearu zavisnost između dve promenljive, nameće se pitanje koja je to najpogodnija linija koji opisuje tu zavisnost? U principu, zanima nas da pronađemo liniju y\_pred = a x + b koja na najbolji način opisuje date podatke. Podaci su dati kao:

\* x\_i - vrednost nezavisne promenljive za svako merenje i,

\* y\_i - vrednost zavisne promenljive za svako merenje i,

\* y\_pred - \*predviđena\* vrednost zavisne promenljive za svako merenje i.

Očigledno je da će se y\_i i y\_pred razlikovati u određenoj meri (u nekim merenjima manje u nekim merenjima više), ali je potrebno \*u proseku\* smanjiti ovu razliku. Jednostavna linearna regresija je postupak koji pronalazi najpogodniju liniju koja opisuje zadate podatke x\_i i y\_i, minimizujući grešku predikcije e\_i = y\_i - y\_pred.

**Metod najmanjih kvadrata**

Linija koja najbolje "fituje" (od eng. \*fit\* - pristajati) podatke će biti ona linija koja najbolje odgovara podacima, tj. najbolje opisuje vezu između nezavisne x i zavisne promenljive y.

Najbolje fitovana linija ima najmanju grešku predikcije za svako merenje i.

Jedan od metoda za pronalaženje ovakve linije je \*\*metod najmanjih kvadrata\*\* (eng. \*OLS - Ordinary Least Squares\*). Ovaj metod minimizuje sumu kvadrata grešaka predikcije (\*SSE - Sum of Squares Error\*). SSE je metrika koja govori koliko dobro data linija opisuje podatke.

\* Jednačina linije koja najbolje "fituje" je: y = a x + b. Gde je \*a\* je nagib linije (\*slope\*) i \*b\* odsečak na y-osi (\*intercept\*).

\* Potrebno je odrediti vrednosti \*a\* i \*b\* tako da je suma kvadratnih grešaka predikcije što manja.

\* Dakle, potrebno je pronaći \*a\* i \*b\* koji minimizuju:

SSE=sum{i=1}^{n}(y\_i - y\_pred)^2 = sum\_{i=1}^{n}(y\_i - a x\_i - b)^2

**Pretpostavke linearne regresije**

Pretpostavke linearne regresije - su osnovni uslovi koji se moraju ispuniti kako bi dobili pouzdane i validne rezultate u analizi odnosa između nezavisne promenljive X i zavisne promenljive Y. Pretpostavke osiguravaju da naš odabrani model linearne regresije odgovara podacima i da će predikcije biti precizne i pouzdane unutar određenih intervala.

Pretpostavke se mogu proveriti vizualno koristeći grafike, ili koristeći statističke testove (vise o statističkim testovima na sledećim vežbama).

\*\*L.I.N.E. pretpostavke\*\* - 4 glavne pretpostavke moraju da važe, kako bi bili sigurni da je regresioni model pogodan za dati problem:

- [Linearnost](#1.-Linearity)

- [Independece of errors](#2.-Independece-of-errors)

- [Normality of errors](#3.-Normality-of-errors)

- [Equal variance](#4.-Equal-variance)

## 1. Linearity

Linearnost - pretpostavka važi kada je odnos izmedju x i y linearan ✅.

Pravimo model linearne regresije koji predviđa cenu kuće na osnovu površine placa .  
Određujemo parametre modela, presek i nagib, kao i na prethodnim vežbama.

Y vs Y\_pred

Sa prethodnog grafika vidimo da je odnos između X i Y linearan. Da bismo bili sigurni, uporedićemo `y` i `y\_pred`, odnosno Y i Y\_pred.

Tačke na grafiku treba da budu blizu prave x=y kako bi predikcije bile tačne. Velika odstupanja bi ukazivala da linearni model nije adekvatan ❌, a mala odsupanja da jeste adekvatan ✅.

Pošto nema većih odstupanja, linearnost je zadovoljena ✅.

(Ukoliko bismo na grafiku uočili velika odstupanja, zaključili bi da je linearni model pogrešan izbor.)

## 2. Independece of errors

Nezavisnost grešaka - pretpostavka važi kada su greške statistički nezavisne.

Testiramo tako što: crtamo grafik reziduala.  
Ako grafik reziduala ima uočljiv šablon, onda je pretpostavka narušena ❌. Pretpostavka vazi kada nema šablona ✅.  
Testira se nad **rezidualima**, ne nad osobinama.

### Reziduali

Podsedimo se: reziduali su procene grešaka iz uzorka podataka. To je razlika između prave i procenjene vrednosti e = y - y\_pred.  
Primer reziduala: Narandžastom linijom pokazujemo odstupanje yi\_pred od stvarne vrednosti yi .

Enkapsuliramo kod u funkciju koja računa rezidual za svako y i y\_pred.  
Prikazujemo grafik reziduala; na x-osi se nalazi prediktovana vrednost , a na y-osi su reziduali.

Ovaj grafik nema uočljiv šablon, pa zaključujemo da su greške nezavisne, odnosno pretpostavka važi ✅.

Durbin-Watson statistički test

Alternativni način da proverimo nezavisnost grešaka je koristeci \*Durbin-Watson\* statistički test. Ovaj test proverava autokorelaciju izmedju reziduala; proverava da li postoji šablon u podacima.

Tumačenje autokorelacije za vrednost d:

- 1.5 <= d <= 2: Nema autokorelacije - pretpostavka važi ✅.

- d < 1.5: Pozitivna autokorelacija - pretpostavka ne važi ❌.

- d > 2: Negativna autokorelacija - pretpostavka ne važi ❌.

Reziduali nisu autokorelirani, pa pretpostvka važi ✅.

\*Napomena: ovo smo već prikazali preko grafika reziduala, sada smo samo potvrdili.

## 3. Normality of errors

Normalnost grešaka - pretpostavka važi kada su zadovljene dve potpretpostavke:

1. Srednja vrednost 0 za greške.
2. Greške su normalno distribuirane.

Napomena: Bitnije je da je srednja vrednost 0 za greške, nego da su greške normalno distribuirane jer distribucija grešaka zavisi od uzorka podataka (detaljnije na predavanjima).

### 3.1 Srednja vrednost 0 za greške

Testiramo prvu pretpostavku da greške imaju srednju vrednost 0.  
Greške već po definiciji imaju srednju vrednost 0.  
Možemo se i sami uveriti da ’e srednja greška biti 0.

Prethodni broj je veoma blizu 0. Zašto prethodni broj nije tačno 0? - Zbog zaokruživanja brojeva (numerička nestabilnost).

Greške po definiciji imaju srednju vrednost 0, i nije dovoljno da samo izračunamo srednju vrednost. Zato koristimo metodu - \*LOWESS (locally weighted scatterplot smoothing)\* - gde želimo da dobijemo što ravniju liniju.

Reziduali ne odstupaju mnogo od nule, pa pretpostavka da su greške distribuirane oko 0 važi ✅.

### 3.2 Greške su normalno distribuirane

Pretpostavku testiramo pomoću histograma reziduala.

Reziduali na histogramu treba da budu normalno distribuirani - puno malih i malo velikih grešaka, podjednako pozitivne i negativne.

\*Histogram prikazuje broj primera (frekvenciju) unutar unapred definisanih intervala (binova).\*

Izgleda da greške nisu normalno distribuirane. Da bi bili sigurni, koristimo \*Anderson-Darling\* statistički test. Ovaj test vraća p-vrednost. Tumačenje:

- ako je p-vrednost >= praga - distribucija je normalna ✅.

- ako je p-vrednost < praga - distribucija nije normalna ❌.

Iz priloženog vidimo da reziduali imaju srednju vrednost 0, ali nisu normalno distribuirani, pa pretpostavka ne važi ❌.

Ipak, ukoliko je uzorak podatak velik (što jeste u našem slučaju), možemo zanemariti to što reziduali nisu normalno distribuirani smatrati ovu pretpostavku validnom ✅.

## 4. Equal variance

Jednaka varijansa - pretpostavka važi kada gausijane grešaka oko regresione prave imaju jednaku varijansu ✅.

Jednaku varijansu vidimo sa grafika reziduala - pretpostavka važi kada nema šablona u rezidualima.

Tada distribucija reziduala oko srednje vrednosti 0 ima jednaku varijansu za svako x\_i.

Pošto reziduali nemaju šablon, zaključujemo da pretpostavka važi ✅.

Jednaku varijansu možemo potvrditi i \*Goldfeld-Quandt\* testom. Ovaj test vraća p-vrednost. Tumačenje:

- ako je p-vrednost >= praga - varijansa je jednaka ✅.

- ako je p-vrednost < praga - varijansa nije jednaka ❌.

Dobijamo jednaku varijansu pa zaključujemo da ova pretpostavka važi.

**Zaključivanje o modelu**

Nakon testiranja pretpostavki o regresionom modelu (L.I.N.E. pretpostavke) želimo da tumačimo rezultate; ispitujemo kvalitet i pouzdanost modela.

Tako ćemo u nastavku:

- ispitati linearnost model preko [t-testa](##testiranje-linearnosti-pomocu-t-testa).

- naći raspon mogućih vrednosti za parametre modela (nagib i odsečak) koristeći [interval poverenja](##interval-poverenja).

- naći raspon mogućih vrednosti za Y koristeći [interval predikcije](##interval-predikcije).

## 1. Testiranje linearnosti pomoću t-testa

Linearnost - pretpostavka važi kada je odnos između X i Y linearan ✅.

Ovu pretpostavku smo do sad testirali vizualno preko grafika.

Drugi nacin da testiramo je koristeći statističke testove, konkretno \*\*t-test\*\* (slično kao što smo na prethodnim vežbama testirali ostale L.I.N.E. pretpostavke putem statističkih testova).

T-test testira da li postoji linearan odnos između X i Y.

Hipoteza kaže (test pretpostavlja) da: \*\*ne postoji\*\* linearna veza između X i Y tj. a=0 u y = ax + b.

Ovu hipotezu testiramo koristeći \*\*T-vrednost\*\* iz t-distribucije:

- Ako T-vrednost nije česta (T-vrednost je daleko od 0), onda odbacujemo hipotezu i zaključujemo da odnos između X i Y \*\*jeste\*\* linearan ✅.

- Ako T-vrednost jeste česta (T-vrednost je oko 0), onda ne možemo da odbacimo hipotezu.

\*Napomena: ako je T-vrednost česta, to ne znači da je veza nelinearna; ne znamo da li veza linearna ili nelinearna.\*

Zanima nas konkretna T-vrednost za naš regresioni model. T-vrednost ne računamo ručno već koristimo statsmodels koji automatski računa T-vrednost.  
Crtamo t-distribuciju i konkretnu T-vrednost za površinu placa (dobijamo iz statsmodels paketa).  
Gledamo grafik koliko je česta konkretna T-vrednost za površinu placa.

Alternativno, T-vrednost možemo da pročitamo i pozivom *summary()* metode nad modelom (obelezeno sa *t*).

Podsetimo se, hipoteza pretpostavlja da \*\*ne postoji\*\* linearna veza između X i Y. Ako T-vrednost nije česta, hipoteza ne važi => zaključujemo da je odnos linearan.

Pošto konkretna T-vrednost (20.76) nije česta, hipoteza ne važi => postoji linearna veza između X i Y ✅.

**P-vrednost**

Umesto T-vrednosti, linearnost možemo da proverimo koristeći p-vrednost.

P-vrednost je verovatnoća da ćemo iz t-distribucije izvući konkretnu T-vrednost (20.76) ili neku još manje verovatnu (npr. 28, 50, -50, 123,...).

Ako je p-vrednost mala, onda T-vrednost nije česta i veza između X i Y jeste linearna.

\*\*Šta je \*mala\* p-vrednost?\*\*

Kao prag se obično uzima vrednost alpha = 0.05 = 5% (eng. \*significance level\*).

Ako je konkretna `p-vrednost ≤ 0.05` onda hipoteza ne važi => 95% smo sigurni da veza jeste linearna ✅.

Pogledaj gore u pozivu metode `summary()` kolika je p-vrednost (obeleženo sa `P>|t|`). Vidimo da je mali broj, blizu 0.

Vidimo da je p-vrednost izuzetno mala, daleko ispod praga `alpha`, pa smo pokazali da linearnost postoji ✅.

Proširujemo funkciju (sa prošlih vežbi) koja proverava linearnost, dodajući t-test.

Više ne moramo da crtamo grafik da bi proverili linearnost veze, vež linearnost proveravamo koristeći t-test.

## 2. Interval povjerenja

Skup podataka sa kojim radimo sadrži podatke o 280 kuća u gradu Windsor. Koristimo linearni model y = ax + b kako bismo procenili stvarne vrednosti kuća. Važno je napomenuti da nemamo podatke o cenama svih kuća u tom gradu, već samo o određenom broju kuća. U stvarnom svetu to je redovna pojava i retko imamo pristup svim podacima, već samo određenom broju.

Da smo imali drugačije podatke dobili bi nešto drugačiji model sa drugačijim vrednostima za nagib i presek.

Postavlja se pitanje: koliko bi se promenili parametri modela kada bi proširili skup podataka? Koliko je trenutni model dobar?

Da bismo na ovo odgovorili treba nam \*sigurnost\* koju zovemo interval poverenja.

Interval poverenja (eng. \*confidence interval\*) je raspon u kom će se naći prava vrednost parametara. Pravu vrednost parametara bi dobili kada bi imali celu populaciju u skupu podataka, a ne samo uzorak iz populacije. Npr. kada bi imali popisane sve kuće iz grada Windsor, a ne samo njih 280.

Interval poverenja ne računamo ručno već koristimo `statsmodels` koji automatski računa interval.

Interval poverenja se kreira za neki procenat pouzdanost, slično kao što smo kod [p-vrednosti](###p-vrednost) uzimali prag od 95%.

Tako i ovde želimo sa 95% tačnosti da odredimo između kojih vrednosti se nalazi npr. nagib?

Intervale poverenje dobijamo pozivom metode `model.conf\_int()`, ili alternativno citamo iz `model.summary()` ( obelezeno sa `[0.025 0.975]`).

Tumačenje prethodnog ispisa u konzoli: sa pouzdanošću od 95% tvrdimo da se cena kuće poveća za iznos između 72.4$ i 87.6$ kada se površina placa poveća za 1m^2.

Vizualizujemo interval poverenje za nagib:

- plava linija - prediktovana regresija (kao i do sad).

- zelena linija - cena kuće se poveća za minimum 72.4$ kada se površina placa poveća za 1m^2.

- crva linija - cena kuće se poveća za maksimum 87.6$ kada se površina placa poveća za 1m^2.

Sa 95% tvrdimo da se nagib regresionog modela nalazi između zelene i crvene linije.

Slično kao i za nagib, možemo vizualizovati interval poverenja za presek (probaj samostalno da uradiš).

## 3. Interval predikcije

Do sad smo intervale poverenja tražili za parametre (nagib i odsečak).

Intervale poverenja možemo da definišemo i za same predikcije.

Interval predikcije odgovara na pitanje - između kojih vrednosti se nalazi Y\_i za dato X\_i? Npr. koji je raspon cene kuće sa placom od 450m^2 ?

Interval predikcije dobijamo pozivom metode `model.get\_prediction(x\_with\_const).summary\_frame()`.

**Višestruka linearna regresija**

Do sada smo koristili jednostruku (jednostavnu) linearnu regresiju, gde postoji samo jedna nezavisna promenljiva x.

\*Višestruka\* linearna regresija se odnosni na linearnu regresiju sa \*više\* nezavisnih promenljivih x\_1, x\_2,...x\_n.

Primer višestruke linearne regresije: y = 2.7 x\_1 + 3.4 x\_2 + 5.

U nastavku prikazujemo:

- primer [višestruke linearne regresije](#primer-višestruke-regresije),

- dodatnu [pretpostavka o savršenoj kolinearnosti](#kolinearnost),

- proveru linearnosti pomoću [t-testa](#t-test) i [f-testa](#f-test),

- kako [evaluiramo model](#evaluacija-modela).

## 1. Primjer višestruke linearne regresije

Do sad smo pravili model koji predviđa cenu kuće Y na osnovu površine placa X. Sada uvodimo dodatnu nezavisnu promenljivu i pravimo model višestruke linearne regresije koji predviđa cenu kuće Y na osnovu površine placa X\_1 i broja spavaćih soba X\_2.

Pravimo model gde koristimo sve nezavisne promenljive, tako što pozovemo `x = df.drop(columns=['price'])` kako bi učitali sve nezavisne promenljive koje su nam na raspolaganju.

## 2. Kolinearnost

Pored L.I.N.E. pretpostavki, u višestrukoj regresiji moramo da zadovoljimo još jednu pretpostavku da: \*\*ne postoji savršena kolinearnost između 2 ili više nezavisnih promenljivih\*\*.

\*Savršena kolinearnost između dve promenljive x\_1 i x\_2 postoji kada su povezane linearnom funkcijom: x\_2 = a x\_1 + b.\*

Primer: recimo da u podacima postoji još jedna kolona koja se zove `toilet` (kopija kolone `bathrms`). Ta promenljiva je u linearnoj relaciji sa promenljivom `bathrms`, tako što je x\_2 = 1 x\_1 + 0. U ovom slučaju postoji savršena kolinearnost i treba izbaciti tu kolonu `toilet` zbog koje nastaje savršena kolinearnost.

\*Napomena: posmatraćemo savršenu kolinearnost između tačno 2 promenljive.\*

Kako da uočiomo kolinearnost? - preko \*\*matrice korelacije\*\*.

Kolinearnost i korelacija su dva slična pojma. \*\*Korelacija\*\* se odnosi na linearnu vezu dve promenljive, a \*\*kolinearnost\*\* se odnosi na linearnu vezu dve \*nezavisne\* promenljive u \*linearnoj regresiji\*.

\*\*Kovarijansa\*\* - \*\*koeficijent korelacije\*\* - konkretan broj u rasponu [-1,1] koji meri linearnost između promenljivih. Vrednost -1 znači savršena negativna kolinearnost, a +1 savršena pozitivna kolinearnost. Jaka korelacija (ne savršena) - vrednosti u intervalu (-1, -0.8] U [0.8, 1).

Matrica korelacija pokazuje vrednosti koeficijenata korelacije za sve parove promenljivih.

- Ako matrica korelacija pokaže savršenu pozitivnu ili negativnu kolinearnost (+1 ili -1) između nezavisnih promenljivih (u našem primeru za kolonu `toilet`), onda tu kolonu treba ukloniti.

- Ako matrica korelacija pokaže jaku korelaciju za nezavisne promenljive, tada pretpostavka o kolinearnosti nije narušena.

\*Savet\*: kada se uoči jaka korelacija između nezavisnih promenljivih, vredi ispitati model bez tih promenljivih jer će se možda dobiti bolje performanse, ali nema garancije.

- Ostale korelacije (makar one bile jake) nisu problematične.

## 3. T-test

U jednostrukoj regresiji, linearnost smo testirali vizualno preko grafika ili koresteći \*\*t-test\*\*.

T-test se koristi i u višestrukoj regresiji, i računa se za svaku promenljivu posebno.

U višestrukoj regresiji, t-test proverava značaj pojedinačnih promenljivih \*\*u prisustvu ostalih promenljivih\*\*.

Za svaku promenljivu zasebno testiramo da li je u linearnoj vezi sa Y, ali u prisustvu ostalih promenljivih.

Može se ispostaviti da neka promenljiva nije značajna kada model ima pristup drugim promenljivim. Ova situacija se javlja kada su dve promenljive u korelaciji (npr. `toilet` i `bathrms` ili kao `bedrooms` i `stories`).

Primer gde poredimo 2 modela:

1. model jednostruke regresije koji predviđa cenu Y na osnovu broja soba X. Vidimo da je `p-vrednost ≤ 0.05` za promenljivu `bedrooms` => 95% smo sigurni da veza jeste linearna ✅.

2. model višestruke regresije koji predviđa cenu Y na osnovu svih promenljivih. Vidimo da je `p-vrednost > 0.05` za promenljivu `bedrooms` => ne mozemo zaključiti linearnost, jer t-test nije validan. Zaključujemo da promenljiva `bedrooms` nema veliki uticaj u prisustvu ostalih promenljivih.

\*Napomena:\* kada je `p-vrednost > 0.05` za neku nezavisnu promenljivu, verovatno možemo dobiti bolji model izbacivanjem te promenljive.

\*Podsetnik: da vidimo p-vrednost čitamo kolonu `P>|t|` za svaku promenljivu.\*

## 4. F-test

F-test provera da li je bar jedna nezavisna promenljiva u linearnoj vezi sa Y.

F-test je jednostavniji od t-testa jer ne testira linearnost u prisustvu ostalih promenljivih kao što to t-test radi.

Slično kao kod t-testa gde postoji p-vrednost, tako postoji i p-vrednost kod F-testa. Ako je p-vrednost mala, onda F-vrednost nije česta i bar jedna promenljiva X ima linearnu vezu sa promenljivom Y.

\*\*Šta je \*mala\* p-vrednost?\*\*

Kao prag se obično uzima vrednost alpha = 0.05 = 5% (eng. \*significance level\*).

Ako je konkretna `p-vrednost ≤ 0.05`, tada smo 95% sigurni da bar jedna promenljiva X ima linearnu vezu sa promenljivom Y ✅.

Vrednosti čitamo iz `statsmodels` tabele pozivom metode `summary()`: F-vrednost je obelezena sa `F-statistic`, dok je p-vrednost je obelezena sa `Prob (F-statistic)`.

U konkretnom primeru vidimo da je p-vrednost F-testa mala i zaključujemo da bar jedna promenljiva X ima linearnu vezu sa promenljivom Y. ✅.

## 5. Evaluacija modela

Kako da znamo koji model je bolji: model sa jednom promenljivom, sa dve, ili više promenljivih?

Pretpostavljamo da što bolje opisemo kuću (što više znamo o nekoj kući) bolje ćemo prediktovati cenu kuće, ali nismo sigurni da je ovo uvek tačno. Zato nam treba način da izmerimo koliko je neki model dobar.

Koeficijent determinacije - r^2 - je način da izmerimo kvalitet modela. Uzima vrednosti u opsegu [0, 1]. Što je r^2 bliže 1, to je model bolji; što je dalje od 1, to je model gori.

Poredimo r^2 na primeru dva modela: prvi predviđa cenu na osnovu površine placa, a drugi na osnovu broja soba.

Vidimo da je r^2 veći za `lotsize` model, sto znači da površina placa bolje opisuje cenu kuće, nego broj soba.

**Prilagođeni r^2**

Kada dodajemo nezavisne promenljive u model, pravilo je da vrednost r^2 nikada ne opada. Pošto r^2 ne opada, mogli bismo zaključiti da je uvek dobro dodati nezavisnu promenljivu u model ➡️ Ovaj zaključak je pogrešan! ❌

Nova nezavisna promenljiva koju dodajemo u model može biti od pomoći, ali isto tako može biti beskorisna ili sadržati pogrešne vrednosti. Uvođenjem \*loše\* nezavisne promenljive se smanjuje kvalitet modela, a r^2 mera to neće pokazati (jer se nikada ne smanjuje).

Zato ćemo za višestruku regresiju koristiti \*\*prilagođeni r^2\*\* ✅. Prilagođeni r^2 je modifikacija r^2 mere koja uzima u obzir kompleksnost modela i broj podataka. Pouzdan je pokazatelj kvaliteta modela.

## 6. Train/val/test

## Evaluacija modela sa \*train/val/test\* podelom podataka podrazumeva razdvajanje dostupnih podataka na trening, validacioni i test skup. Najčešće se podaci dele u odnosu: 80/20/20. Nekad je drugačiji odnos bolji, a to zavisi od količine podataka i problema koji se rešava.

## \*\*Trening skup\*\*

## Model se trenira na trening skupu. Treniranje modela podrazumeva traženje parametara modela kako bi minimizovao gresku. U našem slučaju parametri modela su koeficijenti uz nezavisne promenljive x\_1, x\_2,...x\_n.

## \*\*Validacioni skup\*\*

## Validacioni skup je odvojen skup podataka koje model nije video (nisu u trening skupu). Ove podatke model \*ne vidi\* u toku treninga.

## Nad podacima iz validacionog skupa merimo performanse modela nekom metrikom (npr. prilagođeni r^2, SSE, ...). Performanse nam govore koliko model dobro generalizuje nad podacima koje model nije video do sad.

## Primer: validacioni skup služi da pogledamo r^2 meru nad kućama koje model do sad nije video (metrika se računa nad validacionim skupom). Kada dobijemo neku meru, ta mera govori koliko dobro trenirani model generalizuje nad novim podacima. Zatim možemo da probamo da dodamo ili izbacimo neku kolonu i gledamo kako će se mera promeniti. Iterativno dobijamo model koji najbolje generalizuje.

## \*\*Test skup\*\*

## Test skup služi da evaluira model nad nepoznatim podacima. Test skup ocenjuje moć generalizacije modela nad novim podacima. Slično kao i validacini skup, sa razlikom da test skup evaluiramo \*samo jednom\*, u slučaju da uopšte imamo pristup test skupu. Često nemamo pristup test skupu nego će to biti podaci koje dobijamo u realnom vremenu nad kojima trenirani model prediktuje.

**Nedostajuće vrijednosti u podacima**

Kada postoje nedostajuće vrednosti u podacima, nije moguće pronaći parametre modela. Zbog toga je potrebno obraditi nedostajuće vrednosti. Metode se dele u dve glavne grupe:

* za [uklanjanje podataka](about:blank#uklanjanje-podataka) i
* [popunjavanje podataka](about:blank#popunjavanje-podataka).

Vidimo da postoje nedostajuće vrednosti u dve kolone lotsize(m^2) i stories. Ako bismo probali da nađemo regresioni model koji koristi bar jednu od ovih kolona dobili bi grešku.

**Uklanjanje podataka**

Prva grupa metoda su metode za uklanjanje podataka. Možemo ukloniti čitav red ili čitavu kolonu.

**Uklanjanje redova**

U praksi najčešći metod za obradu nedostajućih vrednosti jer je najlakši za implementirati. Uklanjaju se svi redovi kojima nedostaje bar jedna vrednost.

Problematično je ako ima puno redova koje treba ukloniti, jer tada gubimo puno podataka (smanjujemo uzorak), što može dovesti do lošeg modela.

Da izbacimo sve redove koji sadrže bar jednu nedostajuću vrednost koristimo metodu `dropna()`

**Uklanjanje kolona**

Uklanjanjem kolone koja sadrži puno nedostajućih vrednosti možemo da pojednostavimo i poboljšamo performanse modela.

Ako kolona ima puno nedostajućih vrednosti, verovatno je cela kolona neinformativna i bolje je izbaciti tu kolonu (nego npr. sve redove).

Ukoliko je kolona veoma informativna(kao što je npr. `lotsize(m^2)`), onda izbacivanje kolone nije preporučljivo jer dobijamo lošiji model, pa se takva kolona zadržava u podacima.

Da izbacimo kolonu koristimo metodu `drop(columns=[])`.

**Popunjavanje podataka**

Brisanje podataka može dovesti do smanjivanja uzorka podataka i samim tim do lošijeg modela. Umesto da brišemo podatke, nedostajuće vrednosti možemo da popunimo (eng. \*data imputation\*).

Ne postoji univerzalno najbolji način za popunjavanje, jer zavisi od problema koji se rešava.

Iako ne znamo koja konkretna vrednost treba da stoji (umesto nedostajuće vrednosti), možemo da procenimo.

**Popunjavanje prosjekom**

Da nedostajuće vrednosti zamenimo prosekom vrednosti te kolone koristimo metodu `fillna()`.

\*Napomena: ne mora nužno da se koristi srednja vrednost, nekada medijan ili mod daju bolje rezultate.\*

**Popunjavanje polinomom**

Interpolacija je tehnika koja pronalazi "skrivenu" funkciju koja najbolje opisuje podatke. Interpolacija funkcioniše tako što pronalazi polinom koji prolazi kroz tačke (podatke).

U narednim primerima pokazaćemo interpolaciju za površinu placa. Ekvivalentno bismo mogli primeniti interpolaciju i na kolonu `stories` koja ima nedostajuće vrednosti.

Primer: interpoliramo (i ekstrapoliramo) vrednosti u koloni `lotsize(m^2)` polinomom 1, 3, i 10. stepena. Na sledećem grafiku se vidi da se polinom 10. stepena najbolje uklopio u podatke.

**Popunjavanje splajnom**

Umesto da tražimo jedan polinom koji će opisati sve vrednosti atributa, možemo da tražimo polinom nad delovima vrednosti; između svake dve tačke formiramo poseban interpolacioni polinom. Ovakva interpolacija zove se interpolacija splajnom. Rezultat interpolacije splajnom nije jedan polinom nego skup polinoma.  
Koristimo funkciju interpolate() i definišemo parametar method. Ovaj parametar govori koja interpolaciona tehnika se koristi.

#### Linearni splajn

Napomena: nije isto što i linearna regresija. Pretpostavlja linearan odnos između dve susedne vrednosti i aproksimira nepoznatu vrednost (nalazi se između dve poznate vrednosti: …, , …). Između svake dve tačke formiramo polinom prvog stepena. Primer: imamo 4 podatka (a,b,c,d) = (2, NaN, NaN, 5). Znamo a i d, a pitamo se vrednosti za b i c.

a, d = 2, 5

a,b,c,d = interpolated = np.linspace(a,d, num=4) # 2, 3, 4, 5

Za linearni splajn koristimo ugrađenu metodu interpolate(method='linear', limit\_direction='both'). Drugi parametar govori da se radi i ekstrapolacija pored interpolacije, kako bi dobili i početne i krajnje tačke. Obrati pažnju koliko se linearni splajn bolje uklopio u podatke za razliku od interpolacionog polinoma.

#### Ostali splajnovi

Linerani splajn je veoma gruba interpolacija. Prelaz iz jedne u drugu tačku je retko linearan. Zato se često između dve tačke traži polinom većeg stepena.

Na taj način prelazi između tačaka nisu 'grubi' kao kod linearnog splajna.

U praksi se često koriste splajnovi 1, 2, i 3. stepena (retko je potreban veći stepen).

Napomena: Ekstrapolacija splajnom nekad nije dobra ideja. Pogledaj desni kraj splajna i uoči koliko se ne uklapa u podatke. To je zato što ekstrapoliramo poslednjih par vrednosti u podacima (fali poslednjih par vrednosti). U ovakvoj situaciji bilo bi bolje iskoristiti neku od prethodnih tehnika, umesto da ekstrapoliramo.

**Trend u podacima**

Podaci u koloni lotsize(m^2) su sortirani u opadajućem redosledu. Šta da kolona nije bila sortirana? Šta bi bilo da smo izmešali redove u podeli na trening i test skup i zatim primenili interpolaciju? Demonstriramo na primeru:

Sa prvog grafika vidimo da je rezultat interpolacije (nad izmešanim podacima) pronašao odgovarajući splajn.  
Sa drugog grafika vidimo da rezultat interpolacije ne prati trend u podacima za površinu placa.  
Možemo zaključiti da interpolacija nad izmešanim podacima ne daje dobre rezultate. Interpolaciju treba iskoristiti pre mešanja podataka.

Savet: kada se primenjuje interpolacija nad nekom kolonom osigurati se da postoji trend u podacima. U koliko trend ne postoji, onda interpolacija uglavnom nije dobar izbor.