KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS PROGRAMA

AIVARAS GLUODAS

Prognozuojančios duomenų gavybos metodų lyginamoji analizė

Magistro darbas

Darbo vadovas
prof. dr. Vacius Jusas

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS PROGRAMA

AIVARAS GLUODAS

Prognozuojančios duomenų gavybos metodų lyginamoji analizė

Magistro darbas

Vadovas:.....
prof. dr. Vacius Jusas
2015-05-25

Recenzentas: .....
dr. Kęstutis Paulikas
2015-05-25

Autorius: .....
IFM-3/1 gr. studentas
Aivaras Gluodas
2015-05-25

KAUNAS, 2015

AUTORIŲ GARANTINIS RAŠTAS
DĖL PATEIKIAMO KŪRINIO

2015-05-25d.

Kaunas

Autorius, Aivaras Gluodas
(vardas, pavardė)

_____,

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas (toliau vadinama – Kūrinys) „Prognozuojančios duomenų gavybos metodų lyginamoji analizė“
(kūrinio pavadinimas)

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

1) jį sukūrė ir parašė Kūrinyje įvardytas autorius;
2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
7) Autorius už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autorius

Aivaras Gluodas _____ _____
(vardas, pavardė) (parašas)

Santrauka

Darbo analitinėje dalyje apžvelgiamos, prognozuojančios duomenų gavybos metodikos. Palyginimui alikti pasirenkamos trys daugialypės regresijos metodikos: mažiausių kvadratų metodas, polinominė regresija ir mažiausių absoliutinių nuokrypių metodas. Siekiant patyrinėti regresinių modelių kūrimo principus, projektinėje dalyje sukuriamas programinis įrankis, realizuojantis minėtus metodus.

Tyrimui atlikti pasirenkamas duomenų rinkinys su statistika apie miškų gaisrus. Duomenyse - informacija apie gaisro išdegintą plotą ir įvarūs meteorologiniai išmatavimai gaisro dieną. Naudojant programinį įrankį sukuriami trys modeliai, bandantys paaiškinti išdeginto ploto priklausomybę nuo oro sąlygų. Modelių teikiami rezultatai palyginami su tikrom stebėjimų reikšmėm.

Artimiausi realiems duomenims rezultatai gauti taikant mažiausių kvadratų metodą, jo modelio liekanų vidurkis 31% mažesnis už polinominės regresijos ir 9% už mažiausių abs. nuokrypių metodą. Algoritmų greičio tyrimas nustatė, kad trumpiausiai trunka polinominės regresijos algoritmas, jis 15% greitesnis nei mažiausių kvadratų ir 3,8 kartus greitesnis už mažiausių absoliutinių nuokrypių algoritmą. Remiantis sukurtais modeliais, prognozuojant kitas, tos pačios populiacijos reikšmes tiksliausias buvo polinominės regresijos metodas, jo paklaidos 2% mažesnės už mažiausių kvadratų ir 6% už mažiausių absoliutinių nuokrypių metodus.

Summary

Analysis chapter of this paper presents an overview of predictive data mining techniques. For comparative analysis three multiple regression techniques are chosen: least squares regression, polynomial regression and least absolute deviations regression. In order to research regression techniques, software tool is designed, that fits regression curves based on user inputs.

For experiment dataset is chosen holding data on forest fires and various meteorological measurements. Using previously mentioned software tool, three models are fitted, trying to explain burned area of forest varience in regard to meteorological data. Model estimated values are then compared to real observations.

Closest to real data came the results of least squares regression, it's mean squared error is 31% lower than that of polynomial regression and 9% lower than least absolute regression MSE. Algorithm speed test showed that polynomial regression algorithm is the fastest, it's 15% faster than least squares and 3,8 times faster than least absolute deviations algorithm. Data is divided into 2 parts. First part is used as training data to fit models with all methods. Then other values of the same population are predicted using models. Polynomial regresssion model came closest to real observations, least squares was 2% worse and least absolute deviations was 6% worse.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Terminų ir santrumpų žodynėlis

DG (*angl. Data mining*) - duomenų gavyba

MLR (*angl. multiple lienar regression*) - daugialypė tiesinė regresija

PCR (*angl. principal component regression*) - esminių faktorių regresija

RR (*angl. ridge regression*) - gūbrinė regresija

PLS (*angl. partial least squares*) - mažiausių kvadratų metodas

NLPLS (*angl. nonlinear partial squares*) - netiesinis mažiausių kvadratų metodas

MSE (*angl. Mean square error*) – vidutinė kvadratinė paklaida

MAE (*angl. Mean absoliute error*) – vidutinė absoliuti paklaida

1. Įvadas

Duomenų gavyba (DG) - mokymo algoritmų ir statistinių metodų taikymas įvairioms gyvenimo situacijoms. Pritaikius DG gavybos metodus iš turimų duomenų gaunama informacija yra įdarbinama priimant argumentuotus sprendimus. Duomenų gavybos metodus galima pritaikyti daugeliui gyvenimo sričių, kaip finansai, bankininkystė, pardavimai, komercija, kompiuterių tinklai, medicina, populiacijos tyrinėjimas, migracijos tyrimai, moksliniai tyrimai ir t.t. Visos šios sritys vienais ar kitais būdais saugo tam tikrą informaciją, tačiau neturi tinkamų įrankių panaudoti tą informaciją ruošiantis ateičiai.

Pastaraisiais metais ypatingai ištobulėjusios informacijos rinkimo technologijos privedė prie labai greito duomenų generavimo. Toks greitas duomenų bazių augimas reikalauja efektyvių metodų ir įrankių, leidžiančių transformuoti turimus duomenis į naudingą informaciją.

Dėl milžiniško informacijos kiekio gaunamo kiekvieną dieną, svarbu išsiaiškinti kokią duomenų gavybos techniką naudoti duotai duomenų bazei. Duomenų rinkiniai dažnai nėra tikslūs, pilni ar turi pasikartojančią-perteklinę informaciją. Tyrinėjant duomenis būtų patogu turėti įrankį, kuris galėtų parinkti metodą, tinkamą tiriamam duomenų rinkiniui. Šiuolaikiniai duomenų gavybos įrankiai veikia tik su struktūrizuotomis duomenų bazėmis, tačiau didžioji dalis duomenų nėra struktūrizuoti. Kol kas nėra sukurta gero įrankio kuris galėtu veikti su nekorektiškais duomenimis ar parinktų algoritmą, labiausiai tinkamą turimai duomenų bazei.

Prognozavimas – tikriausiai labiausiai išvystyta duomenų gavybos sritis, tuo pačiu jis padeda plačiausiam gyvenimo sričių ratui. Duomenų gavyboje metodo pasirinkimas analizuojant duomenų rinkinį priklauso nuo analitiko. Daugumoje atvejų švaistoma daug laiko bandant kiekvieną prognozavimo techniką, bandant surasti labiausiai tinkamą. Atsiradus modifikuotoms prognozavimo technikoms, analitikas turi žinoti kuris įrankis labiausiai tinkamas turimiems duomenims.

Šiame darbe bus palyginami keli DG prognozavimo metodai, taip patikrinant metodų prognozavimo galimybes. Taip pat bus bandoma nustatyti kiekvieno metodo stipriąsias ir silpnąsias vietas. Tyrimas leis pasirinkti geriausią metodą sprendžiant nesudėtingus duomenų gavybos uždavinius. Tai leis sumažinti neproduktyvų laiką bei gauti geriausią įmanomą prognozę.

2. Analitinė dalis

2.1. Duomenų gavyba

Duomenų gavyba - praeityje surinktų duomenų tyrinėjimas, bandant surasti pasikartojančius dėsnius ar ryšius tarp kintamųjų. Gauti rezultatai pritaikomi kitiems tos pačios populiacijos duomenų poaibiams. Duomenų gavybos metodikos paremtos trimis mokslo sritimis: statistikos, dirbtinio intelekto ir mašinų mokymo [1].

Duomenų gavyba pagal sprendžiamas užduotis skirstoma į dvi dideles sritis: prognozuojančiąją ir aprašomąją. Dažniausiai duomenų gavybos metodams keliama užduotis yra prognozavimas, dėl to ši sritis labiausiai ištobulinta ir turi daugiausiai pritaikymų gyvenimo situacijose. Duomenų gavybos procesas turi tris pagrindinius žingsnius [2].

DG pradedama duomenų rinkimu ir talpinimu į duomenų saugyklą. Pats duomenų rinkimas ir talpinimas yra plati tema, kuriai priklauso dominančių duomenų savybių atradimas, duomenų valymas ir apsaugojimas. Kitas žingsnis – duomenų atranka ir redukcija. Gavyba ar prognozavimui skirto modelio kūrimas yra paskutinis iš pagrindinių žingsnis. Galiausiai gauti rezultatai interpretuojami ir naudojami priimant sprendimus.

Šiais laikais duomenų gavyba jau yra taikoma daugelyje sričių ir vis dar ieškoma daugiau pritaikymų. Tai vyksta todėl, nes duomenų gavybos metodai, naudojant tik sukauptus duomenis, gali atverti galimybes, kurių nebuvo galima net tikėtis. Keli duomenų gavybos pritaikymai pateikti *1 lentelė.* Duomenų gavybos naudojimo sritys

1 lentelė. *Duomenų gavybos naudojimo sritys*

Pritaikymas	**Įvestis**	**Išvestis**
Verslas	Pirkimų istorija, kreditinių kortelių informacija	Kokie pirkiniai dažnai perkami kartu
Tikslinė auditorija	Vartotojų pateikti filmų ar kitų prekių įvertinimai	Rekomenduojami filmai ar kiti produktai
Paieška internete	Vartotojo užklausa	Dokumentai susiję su vartotojo užklausa
Medicininė diagnozė	Paciento istorija, fiziniai, demografiniai duomenys	Diagnozė
Klimato tyrimai	Matavimai gauti iš palydovų	Galimi įvykiai, laiko eilutės
Procesų gavyba	Darbo eiga, įvykiais paremti duomenys	Nesutapimai

2.2. Duomenų paruošimas

Neapdoroti duomenys ne visada tinkami analizei, tai ypač teisinga prognozuojančiajai duomenų gavybai. Duomenys turi būti paruošti ar transformuoti prieš taikant duomenų gavybos metodus. Šis žingsnis labai svarbus, nes skirtingi metodai elgiasi skirtingai priklausomai nuo duomenų paruošimo. Egzistuoja daugelis duomenų paruošimo metodų, skirtų pasiekti skirtingus tikslus [5].

- Slenkantis vidurkis. Pasirenkamas taškas eilutėje, tarkime 5-asis narys. Pradedant nuo šio taško skaičiuojamas vidurkis šiam ir prieš jį neesantiems keturiems taškams, taško reikšmė pakeičiama vidurkio reikšme. Veiksmai kartojami likusiems eilutės taškams. Šis metodas padeda sumažinti eilutės variaciją [5].

- Normalizacija. Dauguma modelių yra tinkami taikyti normalizuotoms duomenų bazėms. Išmatuoti įrašai yra suvedami į intervalą [-1, 1]. Iš stulpeliuose esančių įrašų atimamas stulpelio vidurkis, gauti rezultatai padalinami iš stulpelio standartinio nuokrypio. Atlikus šiuos veiksmus sumažinama duomenų rinkinio dispersija. Stulpelio vidurkiai lygūs nuliui o dispersijos lygios vienetui, kiekvienas įrašas turi vienodą galimybę patekti į modelį [5].

- Trūkstamų įrašų valdymas. Trūkstami įrašai – dar viena neapdorotų duomenų problema. Dažniausiai tokia reikšmė egzistuoja tačiau ji buvo praleista renkant duomenis. Tokius įrašus reikia sutvarkyti prieš taikant duomenų gavybos metodus. Dauguma duomenų gavybos modelių sunkiai susidoroja su tokiais įrašais. Kai kurie modeliai tokias reikšmes praleidžia, kiti automatiškai suranda joms pakaitalus. Automatiškai randant pakaitalus atsiranda šališko įrašo tikimybė. Šiai problemai spręsti yra naudojami įverčiai. Vienas būdas užpildyti tuščius įrašus yra vietoje tuščių laukų įrašyti egzistuojančių duomenų vidurkį. Šis būdas nepakeičia viso duomenų rinkinio vidurkio [5].

- Nepriklausomų faktorių skaičiaus mažinimas. Kai duomenų rinkiniui priklauso daugiau nepriklausomų faktorių negu galima įtraukti į modelį, būtina dalį jų pašalinti. Tai daroma atrenkant faktorius, kurie geriausiai apibūdina tiriamą dydį. Tam atlikti yra daug metodų, šiame darbe aptarsime esminių faktorių analizę ir koreliacijos koeficientų analizę [5].

2.3. Koreliacijos koeficientų analizė

Koreliacijos koeficientų analizę vertina priklausomybę tarp dviejų atsitiktinių kintamųjų. Koreliacijos koeficientas lygus kovariacijai padalintai iš atsitiktinių dydžių standartinių nuokrypių sandaugos ir gali įgyti reikšmę, priklausančią intervalui [-1; 1].

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

Neigiamas koreliacijos koeficientas reiškia netiesiogiai proporcingą ryšį: vienam atsitiktiniam dydžiui augant kitas – mažėja, teigiamas reiškia tiesiogiai proporcingą ryšį [6].

***2 lentelė**. Koreliacijos koeficiento reikšmės*

Labai stipri	Stipri	Vidutinė	Silpna	Labai silpna	Nėra ryšio	Labai silpna	Silpna	Vidutinė	Stipri	Labai stipri
-1	nuo -1 iki -0,7	nuo -0,7 iki -0,5	nuo -0,5 iki -0,2	nuo -0,2 iki 0	0	nuo 0 iki 0,2	nuo 0,2 iki 0,5	nuo 0,5 iki 0,7	nuo 0,7 iki 1	+1

Jei koreliacijos koeficientas artimas 1 ar -1, tarp atsitiktinių dydžių egzistuoja labai stiprus, artimas tiesiniui ryšys. Jei vienas iš dydžių yra pastovus koreliacijos koeficientas bus artimas 0. Norint tiksliai nustatyti koreliacijos koeficientą duomenų rinkiniui, kuris gali įgyti labai plataus intervalo reikšmes, duomenis reikia normalizuoti arba suvesti į mažesnį intervalą [6].

Jeigu turime kelių kintamųjų vektorių $x = (x_1, x_2, \dots, x_n)$, galime apskaičiuoti koreliacijas tarp visų kintamųjų porų. Šių koreliacijų visuma sudaro koreliacijų matricą. Galima patikrinti, ar koreliacijos tarp kintamųjų yra statistiškai reikšmingos. Koreliacijas galime skaičiuoti visiems kintamiesiems pasinaudodami matriciniais operatoriais. Tarkime turime daugiamatę nepriklausomų kintamųjų stebėjimų matricą [6]:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{1,N} & \cdots & x_{n,N} \end{bmatrix} \tag{2.2}$$

čia *n* yra požymių (kintamųjų) skaičius, *N* – įrašų (objektų, stebėjimų) skaičius.

Pirmiausia matricą X centruojame $\tilde{X} = X - \bar{X}$, čia $\bar{X}$ yra matrica, kur kiekvieno stulpelio elementai yra vienodi ir lygūs matricos *X* atitinkamo stulpelio aritmetiniam vidurkiui. Po to randame stebėjimų matricos *n*-matę kovariacijų matricą.

$$Q \equiv \begin{bmatrix} Q_{1,1} & \cdots & Q_{1,n} \\ \vdots & \ddots & \vdots \\ Q_{1,N} & \cdots & Q_{n,N} \end{bmatrix} \equiv \frac{1}{N-n-1}\ \tilde{X}^T \tilde{X} \tag{2.3}$$

Koreliacijų matrica randama pagal formulę:

$$K = \sigma^{-0.5} Q \sigma^{-0.5} \tag{2.4}$$

Čia, $\sigma = \begin{bmatrix} Q_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{n,N} \end{bmatrix}$ $\sigma^{-0.5} = \begin{matrix} \frac{1}{\sqrt{Q_{1,1}}} & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \frac{1}{\sqrt{Q_{n,N}}} \end{matrix}$

Koreliacijos matrica yra simetriška, visi jos pagrindinės įstrižainės elementai lygūs vienetui.

2.4. Esminių faktorių analizė

Tarkime turime *k* kintamųjų $X_1, X_2, .. X_n$daugelio kintamųjų tarpusavio priklausomybę vadinsime priklausomybės struktūra, kurią vertinsime stebimų kintamųjų koreliacijomis arba kovariacijomis bei dispersijomis. Taikant pagrindinių komponenčių metodą yra randamos stebimų kintamųjų $X_1, X_2, .. X_n$tiesines tarpusavyje nekoreliuojančios reikšmės $Y_1, Y_2, .. Y_n$ [6].

$$Y_i = \sum_{j=i}^{k} \propto_{k,j} X_j \tag{2.5}$$

Y_i vadinamos pagrindinėmis komponentėmis. Jos tenkina sąlygas:

(2.6)

$$cov(X_i, Y_i) = 0;\ i, j = 1, \dots, k\ i \neq j$$

$$DY_1 \geq DY_2 \geq D3 \geq \cdots DY_n \tag{2.7}$$

$$\sum_{i=1}^{k} \mathrm{DY_i} = \sum_{i=1}^{k} \mathrm{DX_i} \tag{2.8}$$

Pagrindinės komponentės išsaugo tiek informacijos apie kintamuosius, kiek ir pradiniai duomenys. Tuomet bendraisiais faktoriais laikysime sunormuotas pagrindines komponentes. Tarkime $Y_1 = \propto_{1,1} X_1 + \cdots + \propto_{1,k} X_k$, tada surandame $\propto_{1,1} + \cdots + \propto_{1,k}$, maksimizuojančias Y_i reikšmių dispersiją [6]:

$$\mathrm{DY_i} = \sum_{\mathrm{i}=1}^{k} \sum_{\mathrm{j}=1}^{\mathrm{k}} \propto_{1,i} \propto_{1,j} \sigma^2{}_{i,j} \tag{2.9}$$

Čia $\sum_{i=1}^{k} \propto^2{}_{1,\mathrm{i}} = 1,$ o $\sigma^2{}_{i,j} = cov(X_i, X_i).$

Šio uždavinio sprendinys $\propto = (\propto_{1,1} + \cdots + \propto_{1,k})$ yra pradinių kovariacijų matricos *S* tikrinis vektorius ir atitinka maksimalią matricos *S* tikrine reikšmė, kuri žymima $\mathrm{DY_1}$. $\mathrm{Y_1}$ reikšmė vadinama kintamųjų $X_1, X_2, \dots X_k$ pirmąja pagrindine komponente, kuri paaiškina $\frac{100\mathrm{DY_1}}{D}$ procentų bendrosios dispersijos. Kuo daugiau pagrindinė komponentė paaiškina bendrosios kintamųjų dispersijos, tuo daugiau joje išlieka informacijos apie kintamųjų elgesį. Sakoma, kad kvadratinė matrica *V* turi tikrinę reikšmę λ, atitinkančią tikrinį vektorių $\vec{\propto} \neq 0$, jeigu $V \vec{\propto} = \lambda \vec{\propto}$. Pradinius kintamuosius galima išreikšti pagrindinėmis komponentėmis tokiu būdu [6]:

$$X_i = \sum_{\mathrm{j}=1}^{k} \propto_{i,j} \mathrm{Y_j}\ \mathrm{i} = 1, \dots, k \tag{2.10}$$

Norėdami išskirti bendruosius faktorius iš pradžių apskaičiuojame k pradinių komponenčių įverčius:

$$\hat{Y}_i = \sum_{\mathrm{j}=1}^{k} \propto_{i,j} \dots \mathrm{Y_j}\ \mathrm{i} = 1, \dots, k \tag{2.11}$$

2.5. Daugialypė tiesinė regresija

Egzistuoja daug prognozuojančiųjų duomenų gavybos metodų ir jie yra įvairūs, paremti regresija, neurotinklais, sprendimų medžiais ir t.t. Šiame darbe aptarsime ir palyginsime tik regresijos modelius.

Tarkime, kad y yra priklausomas kintamasis, kurio i – tąją reikšmę $y_i$norime prognozuoti esant fiksuotoms nepriklausomų kintamųjų reikšmėms $x_{1,i}, \dots, x_{n,i}$, . Taigi, yra duota stebėjimų matrica:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{1,N} & \cdots & x_{n,N} \end{bmatrix}$$

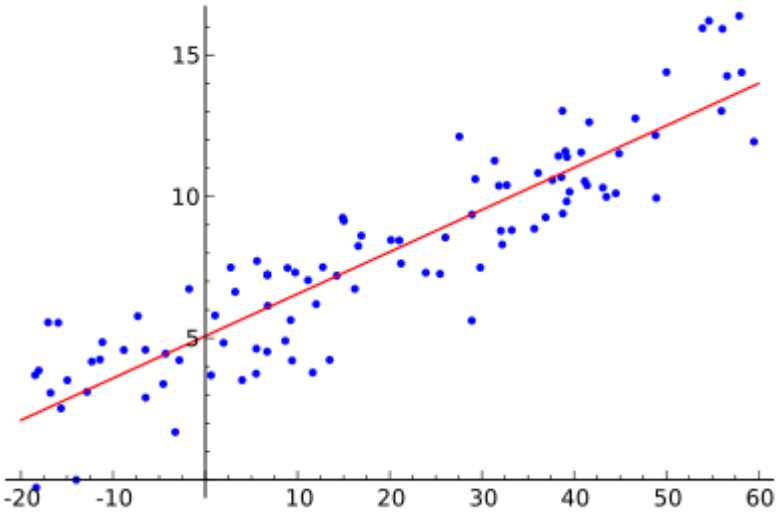
Priklausomo kintamojo stebėjimų vektorius:

$$Y = (y_1, y_2, \dots, y_N)$$

Tiesinės regresijos modelis yra:

$$Y_i = b_1 x_{1,i} + b_2 x_{2,i} + \cdots + b_n x_{n,i} + e_i \tag{2.12}$$

Koeficientai b_i randami mažiausių kvadratų metodu.

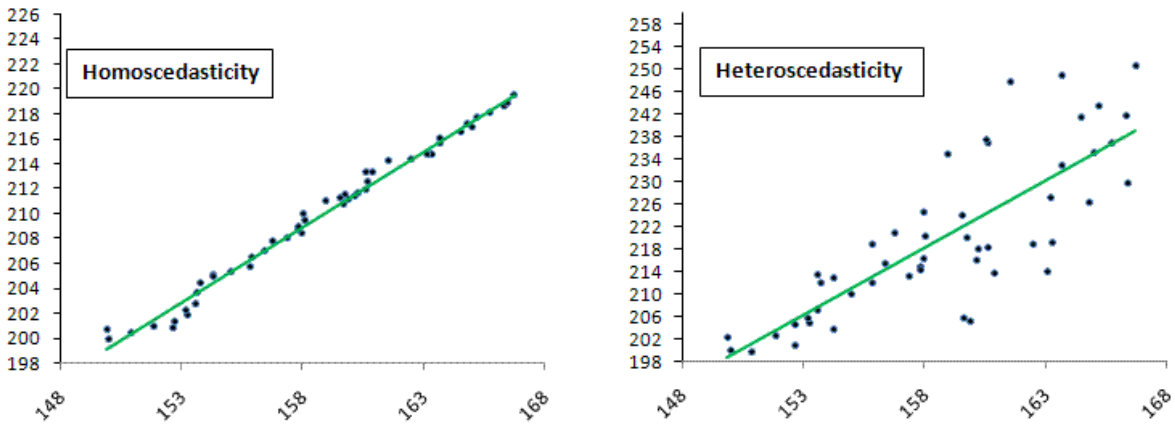


1 pav. *Turimi duomenys ir regresijos modelis*

Regresijos modelyje pabrėžiama y priklausomybė nuo x, bet y gali priklausyti ir nuo kitų neįvardintų kintamųjų. Pabrėžtina tai, kad rinkdamiesi regresijos modelį, tik pasirenkame priklausomybės tipą su nežinomais koeficientais *a* ir *b*. Tikrindami, ar modelis tinka, kartu randame ir šių koeficientų įverčius. Modelio lygtis parodo, kodėl esant tai pačiai reikšmei x_i galima gauti skirtingas y realizacijas y_i [4].

- Laikoma, kad egzistuoja tiesinis ryšys tarp įvesties ir išvesties kintamųjų
- Paklaidos e_i yra atitiktinai pasiskirstę pagal normalųjį skirstinį su vidurkiu 0
- Paklaidos e_i nepriklausomos
- Paklaidos e_i dispersijos lygios nežinomam skaičiui σ^2
- Jokio nepriklausomo kintamojo negalima išreikšti tiesiškai per likusius, tokiu atveju kintamieji nebėra nepriklausomi

Dispersijų lygybės prielaida dar vadinama homoskedastiškumo reikalavimu, pateikta *2 pav***.** Homoskedastiniai ir heteroskedastiniai duomenysTai reikalavimas, kad su kiekvienu fiksuotu y_i galimų *y* reikšmių sklaida būtų vienoda. Pažymėtina tai, kad patys *y* reikšmių vidurkiai kinta tiesiškai, o vienoda tik reikšmių sklaida apie vidurkius. homoskedastiškumo reikalavimas netenkinamas, sakoma, kad duomenys heteroskedastiški - didėjant *x,* sklaida didėja. Modelis, sudarytas labai heteroskedastiškiems duomenims nėra patikimas [4].



2 pav. *Homoskedastiniai ir heteroskedastiniai duomenys*

Multikolinearumas dar viena problema dėl kurios modelis gali būti nekorektiškas. Ši problema pasireiškia kai faktoriai x_i tarpusavyje priklausomi. Problemos galima išvengti šalinant problematiškus faktorius iš galutinio modelio [4].

Galutinis kiekvieno prognozuojančio duomenų gavybos metodo tikslas kuo labiau sumažinti modelio sudėtingumą bei paklaidas. Todėl geras metodas turi sumažinti nepriklausomų faktorių skaičių bei paklaidas.

2.6. Mažiausių kvadratų metodas (*angl. Partial Least Squares Regression*)

Mažiausių kvadratų metodas - dažniausiai naudojamas regresijos modelis. Buvo atrastas dviejų nesusijusių mokslininkų, beveik tuo pačiu metu: Carl Friedrich, Vokietijoje apie 1795 metus ir Adrien Marie Legendre, Prancūzijoje 1805 metais. Anksčiausias metodo panaudojimas, išspausdintas knygoje 1805 metais, bandant prognozuoti kometų trajektorijas [3].

Daugialypės regresijos atveju, modelį sudarys vienas priklausomas kintamasis ir *n* nepriklausomų kintamųjų.

$$Y_i = b_0 + b_1 x_{1,i} + b_2 x_{2,i} + \cdots + b_n x_{n,i} + e_i \tag{2.13}$$

Čia b_i modelio koeficientai, su kuriais modelis turi mažiausią liekanų kvadratų sumą. Modeliavimo metu minimizuojama suma: $\sum_{i=0}^{n} \mathrm{e_i}^2$ kur

$$\mathrm{e_i} = y_i - b_0 - \sum_{i=0}^{n} \mathrm{b_i x_{i,j}} \tag{2.14}$$

Algebrinės formulės mažiausių kvadratų metodui – labai didelės, jas sunku užrašyti kiekvienam įrašui, todėl skaičiavimams atlikti naudojama matricų anotacija [3].

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{11} & \dots & x_{1m} \\ 1 & x_{21} & \dots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \dots & x_{nm} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \tag{2.15}$$

Čia n – įrašų skaičius, m – nepriklausomų kintamųjų skaičius.

Tada modelio išraiška matricų anotacijoje:

$$y = Xb + e \tag{2.16}$$

Formulė rasti mažiausių kvadratų regresijos koeficientų įverčius *b*:

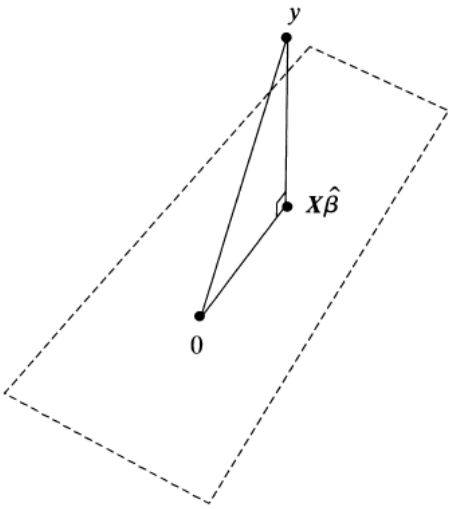
$$\hat{b} = (X'X)^{-1}X^{-1}y \tag{2.17}$$

2.6.1. Geometrinė metodo interpretacija

Naudodami formulę (2.17) gauname modelio koeficientų įverčius $\hat{b}$, jais remiantis sukuriamas regresijos modelis iš to išplaukia prognozuojamos priklausomojo kintamojo y reikšmės $\hat{y} = X\hat{b}$. Tada modelio liekanų vektorius $\hat{e} = y - \hat{y}$. $\hat{e}$ - vektorius n-tosios eilės erdvėje, jo ilgis:

$$|\hat{e}| = \sqrt{\sum \hat{e}_i{}^2} \tag{2.18}$$

Taigi liekanų kvadratų minimizavimas yra tas pats, kaip vektoriaus $\hat{e}$ ilgio minimizavimas, kuris yra lygus atstumui tarp y ir $\hat{y}$, pavaizduota *3 pav*. Geometrinė mažiausių kvadratų metodo interpretacija, n=3 atveju**.**



***3 pav**. Geometrinė mažiausių kvadratų metodo interpretacija, n=3 atveju*

Iš visų kandidatų *b,* atstumas tarp *y* ir Xb minimizuojamas pagal atitinkamą $\hat{b}$. Visą aibę Xb galima įsivaizduoti, kaip *n-1* eilės hiperplokštumą. Mes bandome surasti šios plokštumos tašką artimiausią taškui *y*. Akivaizdu, kad artimiausias Xb plokštumos taškas gaunamas iš *y* išvedus tiesę, statmeną tiriamai plokštumai. Tada $\hat{b}$ aprašo tiesė einanti per tašką *y* ir statmenai kertanti plokštumą Xb, t.y vektorius $X\hat{b} - y$. Tai galima išreikšti [3]:

$$X'\left(X\hat{b} - y\right) = 0 \tag{2.19}$$

Ši formulė yra ekvivalenti formulei *(2.17)*.

2.6.2. Rezultatų interpretavimas

Pirmas tyrimas, kuriant regresijos modelį dažniausiai atliekamas, įsitikinti, ar nepriklausomi faktoriai iš tikrųjų turi naudingos informacijos, bandant paaiškinti priklausomo faktoriaus elgesį. Modelio korektiškumas gali būti nusakomas įvertinant liekanų vektoriaus $\hat{e}$ dydžius. Kuo mažesnės liekanos, tuo geriau modelis aprašo duomenis. Mažiausių kvadratų metodo atveju, bendrą liekanų dydžių įvertį aprašo liekanų kvadratų suma *SSR*. Šis įvertis naudojamas palyginti pilną modelį su visais nepriklausomais faktoriais ir sumažintą modelį, iš kurio pašalinti 1 ar daugiau nepriklausomų faktorių [3].

$$F = \frac{SSR_{sumažintas} - SSR_{pilnas}}{4\hat{\sigma}^2} \tag{2.20}$$

Čia $\hat{\sigma}^2$ – liekanų dispersijos σ^2 įvertis

$$\hat{\sigma}^2 = \frac{\sum \hat{e}_i{}^2}{n} \tag{2.21}$$

Determinacijos koeficientas nusako kaip gerai nepriklausomi faktoriai, aprašo stebimąjį dydį. Jis išreiškiamas taip:

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(\hat{y}_i - \bar{y})^2} \tag{2.22}$$

Stebimas faktoriaus įrašas y_i nuo vidurkio skiriasi dydžiu $y_i - \bar{y}$. Nuokrypis gali būti padalintas į dvi dalis: $y_i - \bar{y} = (y_i - \hat{y}_i) + (\hat{y}_i - \bar{y})$. Antrą dalį nusako ryšys:

$$\hat{y}_i = \widehat{b_0} + \widehat{b_1} x_{i,1} + \cdots + \widehat{b_m} x_{i,m} \tag{2.23}$$

Sumą $\sum(y_i - \bar{y})^2$ galima laikyti pilna stebimo faktoriaus variacija, o sumą $\sum(\overline{y_i} - \bar{y})^2$ – variacijos dalis kurią paaiškina nepriklausomi faktoriai. Taigi R^2 yra santykis tarp paaiškintos ir pilnos variacijų. R^2 kinta tarp 0 ir 1, kuo jo reikšmė artimesnė 1, tuo labiau modelis tinkamas tiriamam duomenų rinkiniui[3].

2.7. Mažiausių absoliutinių nuokrypių metodas (*angl. Least Absolute Deviations Regression*)

Mažiausių absoliučių nukrypimų metodą pirmą kartą pristatė Roger Joseph Boscovitch, beveik 50 metų prieš mažiausių kvadratų metodą, 1757 metais. Jis sukūrė metodą norėdamas atnaujinti netikslius matavimus. Metodas buvo taikomas modeliuoti Žemės formos išmatavimus. Po 30 metų metodą pasiskolino Pierre Simon Laplace. Šis metodas buvo naudojamas retai, kadangi mažiausių kvadratų metodas buvo labiau mėgiamas.

Mažiausių kvadratų metodas buvo populiaresnis dėl, palyginus lengvesnių skaičiavimų ir savo patobulinimų, kuriuos į gyvendino Gausas ir Laplasas. Šiomis dienomis skaičiavimų sudėtingumas nebėra toks ribojantis faktorius, dėl to mažiausių absoliutinių nukrypimų metodas yra vėl naudojamas.

Bendruoju atveju tiesinės regresijos modelis:

(2.24)

$$Y = \alpha + \beta X + e$$

Mažiausių kvadratų metodu įverčiai $\hat{\alpha}$ ir $\hat{\beta}$ parenkami taip, kad liekanų kvadratų suma $\sum \hat{e}_i^{\,2}$ būtų kaip įmanoma maža. Mažiausių absoliutinių nukrypimų metode įverčiai $\hat{\alpha}$ ir $\hat{\beta}$ parenkami taip, kad liekanų absoliutinių reikšmių suma $\sum|\hat{e}_i|$ būtų kaip įmanoma mažesnė. T.y įverčiai $\hat{\alpha}$ ir $\hat{\beta}$ parenkami taip, kad minimizuotų išraišką:

$$\sum|y_i - (a + bx_i)|$$

Skirtumas $y_i - (a + bx_i)$, vadinamas taško (x_i, y_i) nuokrypiu nuo kreivės $\hat{Y} = a + bX$. Mažiausių absoliutinių nukrypimų idėja nėra sudėtingesnė už mažiausių kvadratų metodo idėją, iš tikrųjų ji dar paprastesnė, tačiau patys liekanų skaičiavimai daug sudėtingesni. Mažiausių absoliutinių nuokrypių metodas neturi analitinių formulių liekanų skaičiavimui, todėl modelio kūrimas, šiuo atveju yra iteracinis procesas. Kitame skyriuje pateikiamas liekanų įverčių skaičiavimo algoritmas[3].

2.7.1. Algoritmas

Mūsų tikslas – rasti kreivę, kuri geriausiai modeliuoja turimus duomenis, atsižvelgiant į tai, kad faktoriai, kuriuos įtraukiam į modelį turėtų kuo mažesnę absoliutinių nuokrypių sumą. Pagrindinė algoritmo procedūra – bet kuriam taškui (x_0, y_0) iš visų jį kertančių kreivių išrinkti geriausią. Ši procedūra atliekama turint galvoje, kad mažiausių absoliutinių nuokrypių modelio regresijos kreivė eina per du duomenų taškus. Taigi algoritmas pradedamas turint vieną tašką, tarkim (x_1, y_1), ir jam randama geriausia kreivė, kertanti šį tašką. Ši kreivė taip pat kerta kitą duomenų tašką (x_2, y_2). Toliau randame geriausią kreivę kertančią šį tašką, taip veiksmai kartojami visiems duomenų įrašams. Tęsiant skaičiavimus, gaunamos vis geresnės kreivės. Galiausiai naujai gauta kreivė sutaps su praeitos iteracijos kreive. Įvykus šiai sąlygai iteracijos procesas stabdomas ir gauta kreivė laikoma mažiausių absoliutinių nuokrypių regresijos kreive.

Geriausios kreivės kertančios tašką duotą tašką (x_0, y_0) procedūra – kiekvienam duomenų įrašui skaičiuojamas kreivės kertančios taškus (x_0, y_0) ir (x_i, y_i) nuolydis: $(y_i - y_0)/(x_i - x_0)$. Jei $x_0 = x_i$, nuolydžiui i, gauname ne-apibrėžtumą, tačiau tokius taškus galima ignoruoti. Duomenys perindeksuojami, taip, kad $\frac{y_1-y_0}{x_1-x_0} \leq \frac{y_2-y_0}{x_2-x_0} \leq \cdots \leq \frac{y_n-y_0}{x_n-x_0}$, tada $T = \sum|x_i - x_0|$. Randame indeksą k, tenkinatį sąlygas:

$$\begin{gathered} |x_1 - x_0| + \cdots + |x_{k-1} - x_0| < \frac{T}{2} \\ |x_1 - x_0| + \cdots + |x_{k-1} - x_0| + |x_k - x_0| > \frac{T}{2} \end{gathered} \tag{2.25}$$

Geriausia kreivė, kertanti tašką (x_0, y_0) išreiškiama: $\hat{Y} = \alpha + \beta X$, čia:

$$\begin{aligned} \beta &= \frac{y_k - y_0}{x_k - x_0} \\ \alpha &= y_0 - \beta x_0 \end{aligned} \tag{2.26}$$

Egzistuoja kitas būdas rasti geriausią kreivę, yra paprastesnis iš realizavimo pusės tačiau, atliekama daugiau skaičiavimų. Žinoma, kad regresijos tiesė kerta du taškus. Žinant tai geriausią kreivę įmanoma rasti tarp visų įmanomų kreivių kertančių visas įmanomas duomenų irašų poras (x_0, y_0) ir (x_i, y_i). Galima apskaičiuoti absoliutinių nukrypimų sumas kiekvienai kreivei, ir parinkti tą, kurios suma mažiausia. Šio būdo naudojimo korektiškumas priklauso nuo duomenų įrašų skaičiaus n[3].

2.7.2. Daugialypės regresijos modelis

Bendrą modelį galima pritaikyti daugialypės regresijos atvejui. Tai padarysime sukurdami modelį, duomenų rinkiniui, kuriame saugoma statistika apie kilusius gaisrus, pateikta *3 lentelė.* Kilusių gaisrų statistika

3 lentelė. *Kilusių gaisrų statistika*

	FIRE	AGE	THEFT	INCOME
1	6,2	0,604	29	11,744
2	9,5	0,765	44	9,323
3	10,5	0,735	36	9,948
4	7,7	0,669	37	10,656
5	8,6	0,814	53	9,73
6	34,1	0,526	68	8,231
7	6,9	0,785	18	11,104
8	7,3	0,901	31	10,694
9	15,1	0,898	25	9,631
10	29,1	0,827	34	7,995
11	2,2	0,402	14	13,722
12	5,7	0,279	11	16,25
13	2	0,077	11	13,686
14	2,5	0,638	22	12,405
15	3	0,512	17	12,198
16	5,4	0,851	27	11,6
17	2,2	0,444	9	12,765
18	7,2	0,842	29	11,084
19	15,1	0,898	30	10,51
20	16,5	0,727	40	9,784
21	18,4	0,729	32	7,342
22	36,2	0,631	41	6,565
23	18,5	0,783	22	8,014
24	23,3	0,79	29	8,177
25	12,2	0,48	46	8,212
26	5,6	0,715	23	11,23
27	21,8	0,731	4	8,33
28	21,6	0,65	31	5,583
29	9	0,754	39	8,564
30	3,6	0,208	15	12,102
31	5	0,618	32	11,876
32	28,6	0,781	27	9,742
33	17,4	0,686	32	7,52
34	11,3	0,734	34	7,388
35	3,4	0,02	17	13,842
36	11,9	0,57	46	11,04
37	10,5	0,559	42	10,332
38	10,7	0,675	43	10,908
39	10,8	0,58	34	11,156
40	4,8	0,152	19	13,323
41	10,4	0,408	25	12,96
42	15,6	0,578	28	11,26
43	7	0,114	3	10,08
44	7,1	0,492	23	11,428
45	4,9	0,466	27	13,731

*3 lentelė***.** Kilusių gaisrų statistika pateikti duomenys apie labiausiai apgyvendintas Čikagos miesto vietoves 1975 metais. Kiekviena vietovė turi keturias ją aprašančias charakteristikas:

- FIRE – gaisrų skaičius 1000-čiui gyvenamųjų namų
- AGE – santykis, kiek iš gyvenamųjų namų šiame regione yra senesni nei 25 metų
- THEFT – vagysčių skaičius 1000 gyventojų
- INCOME – vidutinis gyventojo atlyginimas tūkstančiais

Bus sudaromas modelis su priklausomas faktoriumi – FIRE ir nepriklausomais faktoriais: AGE, THEFT ir INCOME.

Šiuo atveju regresijos modelis atrodys taip:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + e$$

Čia $Y = \log(fire)$, $X_1 = AGE$, $X_2 = THEFT$, $X_3 = INCOME$, mažiausių absoliutinių nuokrypių metodu sudarysime modelį, kuris aprašys gaisrų priklausomybę nuo trijų faktorių[3].

2.7.3. Regresijos koeficientų įverčiai

Įverčiai koeficientams b_0, b_1, b_2, b_3 parenkami taip, kad liekanų suma $\sum|\hat{e}_i|$ būtų kaip įmanoma mažesnė. T.y parenkami koeficientų b_0, b_1, b_2, b_3 įverčiai $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$, minimizuojantys:

$$\sum|y_i - (b_0 + b_1 x_{i1} + b_1 x_{i12} + b_1 x_{i3})| \tag{2.27}$$

Tada:

$$b = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \; ir \; x = \begin{bmatrix} 1 \\ x_{i1} \\ x_{i2} \\ x_{i3} \end{bmatrix}$$

Tada formulė *(2.27)* užsirašo:

$$\sum|y_i - b'x_i| \tag{2.28}$$

Norėdami surasti vektorių b, minimizuojantį sąlygą *(2.28)*, naudosime algoritmą pateiktą skyriuje 2.7.1. Pradedant nuo vektoriaus b ir taikome algoritmą tol kol gauname geriausią vektorių $\hat{\beta}$.

Kiekviename iteracijos žingsnyje ieškosime krypties vektoriaus d ir reikšmės t, turint juos, gauname iteracijos rezultatą: $b^* = b + td$.

Vektoriaus įvertis kryptimi d: randama t rekšmė, kuri minimizuoja išraišką:

$$\sum |y_i - (b + td)' x_i| \tag{2.28}$$

Tegu: $z_i = y_i - b' x_i$ ir $w_i = d' x_i$, tada turime rasti reikšmę t, kuri minimizuotų išraišką:

$$\sum |z_i - t x_i| \tag{2.29}$$

Reikšmė t randama, pagal *(2.25)* sąlygas, šiuo atveju:

$$|w_1| + \cdots + |w_{k-1}| < \frac{T}{2}$$
$$|w_1| + \cdots + |w_{k-1}| + |w_k| > \frac{T}{2}$$
$$T = \sum |w_i|$$

Kiekvienoje algoritmo iteracijoje įvertinami 4-i krypčių vektoriai: d_1, d_2, d_3, d_4. Jie aprašo 8-ias įmanomas kryptis, kadangi kiekvienas vektorius d_i rodys ir į priešingą kryptį $-d_i$. Iš šių 8-ių krypčių, mus domina ta, labiausiai mažėjanti artėjant prie $t = 0$. Norėdami nustatyti, mažėjimo greitį, ieškome išvestinės taške $t = 0$. Radus išvestines kiekvienai iš 8-ių krypčių, pasirenkame turinčią labiausiai neigiamą reikšmę. Kai visų krypčių išvestinės teigiamos iteracinis procesas stabdomas.

Paprastos regresijos atveju, mažiausių absoliutinių nuokrypių regresijos kreivė kerta du duomenų taškus. Panašiai ir daugialypėje regresijoje su p nepriklausomų faktorių, o regresijos kreive įtakojama $p+1$ duomenų taškų. Mūsų atveju $p = 3$, $p + 1 = 4$. Taigi pradedant algoritmą parenkami 4 pradiniai taškai su indeksais $i = 1, 2, 3, 4$, ir randame pradinį b įvertį iš $Ab = c$, čia:

$$A = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix} \quad ir \;\; c = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Iš to išplaukia: $b = A^{-1}c$, tada pradiniai krypčių vektoriai d_1, d_2, d_3, d_4 lygūs 4-iems A^{-1} stulpeliams[3].

Taigi, taikydami daugialypės regresijos algoritmą kilusių gaisrų duomenims, pasirenkame 4-is pradinius duomenų įrašus su indeksais $i = 1, 2, 3, 4$, turime *A* ir *c*.

$$A = \begin{bmatrix} 1 & 0.604 & 29 & 11.744 \\ 1 & 0.765 & 44 & 9.323 \\ 1 & 0.735 & 36 & 9.948 \\ 1 & 0.669 & 37 & 10.656 \end{bmatrix} \quad ir \quad c = \begin{bmatrix} 1.825 \\ 2.251 \\ 2.351 \\ 2.041 \end{bmatrix}$$

Tada pradiniai įverčiai:

$$b = A^{-1}c = \begin{bmatrix} 47.93 \\ -23.26 \\ -0.1161 \\ -2.443 \end{bmatrix}$$

Krypčių matrica:

$$A^{-1} = \begin{bmatrix} -284.9 & -308.3 & 157.8 & 436.4 \\ 164.5 & 176.6 & -79.71 & -261.3 \\ 0.5233 & 0.6744 & -0.4656 & -0.7321 \\ 14.59 & 15.51 & -8.185 & -21.91 \end{bmatrix}$$

Dabar turime rasti įverčių vektorių, geresnį už *b*. Norėdami tai padaryti, skaičiuojame kiekvieno A^{-1} stulpelio d_i išvestines. Pirmas žingsnis – rasti $z_i = y_i - b'x_i$ ir $w_i = d'_1 x_i$. Čia *A* matricos i-toji eilutė yra x_i . Gaunami rezultatai:

4 lentelė. *Išvestinės tarpiniai skaičiavimai*

	z	w	sign(z/w)	|w|
1	0	1	*	1
2	0	0	*	0
3	0	0	*	0
4	0	0	*	0
5	3,08	18,71	+	18,71
6	-4,16	-42,68	+	42,68
7	1,48	15,67	+	15,67
8	4,74	35,57	+	35,57
9	2,1	16,43	+	16,43
...				
41	-1,54	-15,58	+	15,58
42	-0,98	-10,86	+	10,68
43	-18,36	-117,48	+	117,48
44	-3,94	-25,17	+	25,17
45	1,18	6,25	+	6,25

Kadangi mes pasirinkome pradinius duomenis su indeksais $i = 1, 2, 3, 4$, pirmos 4-ios *z* reikšmės lygios 0. Gavę $z_i \; ir \; w_i$ reikšmes nustatome z_i/w_i ženklą, jei w_i = 0, gauname ne-

apibrėžtumą, tokie duomenų įrašai neįtakoja išvestinės reikšmės, kadangi $|w_i| = 0$. Dabar sumuojame visus $|w_i|$, kuriems ženklas $sign(z_i/w_i)$ neigiamas arba lygus 0 ir atimame $|w_i|$, kurių ženklas teigiamas. Gauname išvestinę kryptimi $d_1 = -1221$.

Atlikus skaičiavimus kryptimi d_1, išvestinė kryptimi -d_1 lygi $d_1 = -(d_1 - 1) + 1 = 1223$. Išvestinių priešingomis kryptimis suma lygi 2. Gauname išvestines visiems A^{-1} stulpeliams:

$$d = \begin{bmatrix} -1221 & 1223 \\ -1323 & 1325 \\ 654 & -652 \\ 1903 & -1901 \end{bmatrix}$$

Labiausiai neigiama išvestinė kryptimi $-d_4 = -1901$. Todėl ieškome geresnio vektoriaus. Tegu $z_i = y_i - b'x_i$ ir $w_i = d'_4 x_i$. Reikšmė t – geriausios kreivės nuolydžio koeficientas, kuri kerta tašką $(0, 0)$ ir aprašo duomenis (w_i, z_i). Pritaikę algoritmą pateiktą (2.8.1) skyriuje, gauname, kad geriausia kreivė eina ir per duomenų tašką (w_{13}, z_{13}). Todėl 4-ta eilutės duomenys keičiami duomenimis iš 13-os eilutės:

$$A = \begin{bmatrix} 1 & 0.604 & 29 & 11.744 \\ 1 & 0.765 & 44 & 9.323 \\ 1 & 0.735 & 36 & 9.948 \\ 1 & 0.077 & 11 & 13.686 \end{bmatrix} \quad ir \;\; c = \begin{bmatrix} 1.825 \\ 2.251 \\ 2.351 \\ 0.693 \end{bmatrix}$$

Kiekviename žingsnyje 4 duomenų įrašai aprašo regresijos kreivę. Šie taškai vadinami regresijos pagrindu. Kiekvienoje iteracijoje vienas taškas iš regresijos pagrindo keičiamas tašku iš likusių duomenų. Iteracijos procesą pradėjome taškais su indeksais $i = 1, 2, 3, 4$. Pirmame žingsnyje 4-as taškas buvo pakeistas 13-u, antroje iteracijoje 3-ias pakeistas 27-u, tada 2 pakeistas 24-u, 1 pakeistas 21-u ir t.t. Galiausiai išvestinės visomis kryptimis yra teigiamos ir iteracijos procesas stabdomas. Gauname, kad regresijos kreivę geriausiai aprašo duomenų įrašai 35, 18, 37, 43:

$$\hat{\beta} = \begin{bmatrix} 1 & 0.02 & 17 & 13.842 \\ 1 & 0.842 & 29 & 11.084 \\ 1 & 0.559 & 42 & 10.332 \\ 1 & 0.114 & 3 & 10.08 \end{bmatrix}^{-1} \begin{bmatrix} 1.224 \\ 1.974 \\ 2.351 \\ 1.946 \end{bmatrix} = \begin{bmatrix} 4.362 \\ -0.09098 \\ 0.01299 \\ -0.2425 \end{bmatrix}$$

Gauname geriausią regresijos kreivę:

$$\hat{Y} = 4.362 - 0.09098X_1 + 0.01299X_2 - 0.2425X_3$$

2.8. Polinominė regresija (*angl. Polynomial Regression*)

Polinominė regresija – tiesinės regresijos forma, kurioje ryšys tarp priklausomo faktoriaus *y* ir nepriklausomų nepriklausomų faktorių $x_1, x_2, .. x_n$ nusakomas *m*-tosios eilės polinomu. Polinominės regresijos modelis aprašo netiesinę priklausomybę tarp x_i reikšmės ir ją atitinkančios y_i reikšmės.

Bendruoju atveju m-os eilės polinomu išreikštas modelis atrodo taip:

$$Y = \alpha_0 + \alpha_1 X + \alpha_2 X^2 + \cdots + \alpha_m X^m + e \tag{2.30}$$

Kaip ir mažiausių kvadratų atveju, modelį galime išreikšti matricų anotacija. (2.30) formulė atrodys taip:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & {x_1}^2 & \dots & {x_1}^m \\ 1 & x_2 & {x_2}^2 & \dots & {x_2}^m \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & {x_n}^2 & \dots & {x_n}^m \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \tag{2.31}$$

Regresijos koeficientai randami naudojant mažiausių kvadratų metodą:

$$\hat{a} = (X'X)^{-1} X^{-1} y \tag{2.32}$$

2.8.1. Daugialypės regresijos modelis

Kai kurie žingsniai naudoti kuriant paprastus tiesinius ir daugialypius modelius galioja ir daugialypės polinominės regresijos modeliui. Kurdami modelį mes keliame hipotezę, kad priklausomo faktoriaus *y* elgesys gali būti paaiškinamas sudėtine nepriklausomų faktorių $x_1, x_2, .. x_n$ įtaka. Taip pat visi x_i yra nepriklausomi tarpusavyje.

Bendru atveju pilna antros eilės polinominės regresijos modelio su dviem nepriklausomais faktoriais x_1, x_2 išraiška atrodo taip[8]:

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 {x_1}^2 + \alpha_4 {x_2}^2 + \alpha_5 x_1 x_2 \tag{2.33}$$

Pilnas antros eilės polinominis modelis sudarytas tiesinių faktorių x_1, x_2, antros eilės faktorių ${x_1}^2, {x_2}^2$ ir sąveikos faktorius $x_1 x_2$.

Modelį galima išplėsti iki antros eilės polinominio modelio su trim kintamaisiais. Trijų kintamųjų modelis turėtų tiesinius faktorius x_1, x_2, x_3, antros eilės faktorius ${x_1}^2, {x_2}^2, {x_3}^2$, dvigubos

ir trigubos saveikos faktorius: $x_1x_2,\ x_1x_3,\ x_2x_3\ ir\ x_1x_2x_3$. Panašiai modelį galima išplėsti pridedant daugiau kintamųjų ar kuriant aukštesnio laipsnio modelį.

Kaip ir mažiausių kvadratų metode, paklaidas rasim minimizuodami paklaidų kvadratų sumą SSE[8].

$$SSE = \sum e_i^2 = \sum (y_i - \alpha_0 - \alpha_1 x_{1i} - \alpha_2 x_{2i} - \alpha_3 {x_{1i}}^2 - \alpha_4 {x_{2i}}^2 - \alpha_5 x_{1i} x_{2i}) \quad (2.34)$$

Norėdami minimizuoti šią kvadratinę funkciją, randame funkcijos dalines išvestines kiekvieno nežinomojo $b_i,\ i = 0,1..5$ atžvilgiu. Dalines išvestines prilyginame nuliui ir sprendžiame gautą tiesinių lygčių sistemą.

$$\begin{cases} \frac{\partial SSE}{\partial \alpha_0} = 0 \\ \frac{\partial SSE}{\partial \alpha_1} = 0 \\ \dots \\ \frac{\partial SSE}{\partial \alpha_5} = 0 \end{cases} \quad (2.35)$$

Gauname $k = 6$ lygčių sistemą su 6-iais nežinomaisiais, perėjus prie matricų anotacijos gauname:

$$A = \begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} & \sum {x^2}_{2i} & \sum {x^2}_{2i} & \sum x_{1i}x_{2i} \\ \sum x_{1i} & \sum {x^2}_{1i} & \sum x_{1i}x_{2i} & \sum {x^3}_{2i} & \sum x_{1i}{x^2}_{2i} & \sum {x^2}_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum {x^2}_{2i} & \sum {x^2}_{1i}x_{2i} & \sum {x^3}_{2i} & \sum x_{1i}{x^2}_{2i} \\ \sum {x^2}_{1i} & \sum {x^3}_{1i} & \sum {x^2}_{1i}x_{2i} & \sum {x^4}_{2i} & \sum {x^2}_{1i}{x^2}_{2i} & \sum {x^3}_{1i}x_{2i} \\ \sum {x^2}_{2i} & \sum x_{1i}{x^2}_{2i} & \sum {x^3}_{2i} & \sum {x^2}_{1i}{x^2}_{2i} & \sum {x^4}_{2i} & \sum x_{1i}{x^3}_{2i} \\ \sum x_{1i}x_{2i} & \sum {x^2}_{1i}x_{2i} & \sum x_{1i}{x^2}_{2i} & \sum {x^3}_{1i}x_{2i} & \sum x_{1i}{x^3}_{2i} & \sum {x^2}_{1i}{x^2}_{2i} \end{bmatrix} \quad (2.35)$$

Nežinomųjų vektorius *a* ir dešinioji lygties pusė *c:*

$$a = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad \text{ir} \quad c = \begin{bmatrix} \sum y_i \\ \sum y_i x_{1i} \\ \sum y_i x_{2i} \\ \sum y_i {x^2}_{1i} \\ \sum y_i {x^2}_{2i} \\ \sum y_i x_{1i} x_{2i} \end{bmatrix} \quad (2.36)$$

Lygčių sistema *Aa = c* sprendžiama atliekant šias matricų operacijas:

$$Aa = c \ \rightarrow A^{-1}Aa = \ A^{-1}c \ \rightarrow Ia = A^{-1}c \ \rightarrow a = \ A^{-1}c \quad (2.37)$$

2.9. Kriterijai modelių palyginimui

- Vidutinė kvadratinė paklaida (MSE) . Tikriausiai pats reikšmingiausiais kriterijus lyginant įvairių duomenų gavybos metodų prognozavimo galimybes. Kriterijus įvertina liekanų kvadratų vidurkį. Šis kriterijus atskleidžia, kaip gerai modelis prognozuoja reikšmes, kaip pateikiami nauji duomenų rinkiniai. Aukšta MSE reikšmė praneša apie blogą modelį. MSE suteikia tiek pat informacijos kaip ir R^2 determinacijos koeficientas [7].

- Regresijos koeficientų svoriai. Gavus galutinį modelį, iš jo koeficientų svorių galima spręsti ar modelis yra geras. Jei duomenyse yra nereikalingų įrašų – regresijos koeficientų svoriai kyla, galimas multikolinearumas. Nors tokiu atveju MSE gali būti mažas, modelio patikimumas nėra didelis, kas gali privest prie nepastovumo [7].

- Kintamųjų kiekis modelyje. Gera prognozuojantysis duomenų gavybos technika įvertina kuo daugiau prieinamos informacijos. Ji sukuria modelį, kuris ištiria kuo didesnį informacijos kiekį su kaip įmanoma mažu MSE. Tačiau pridedant daugiau kintamųjų, auga tikimybė į modelį įtraukti perteklinę informaciją. Idealiu atveju pasirenkami geriausi kintamieji, kurie suteiks pakankamai informacijos modeliuojant [7].

- Efektyvumo koeficientas. Naudojamas įvairiose mokslo srityse vertinant modelių veikimą.

$$E = 1 - \frac{\sum_{i=1}^{n}(O_i - X_i)^2}{\sum_{i=1}^{n}(O_i - \bar{X})^2} \qquad (2.13)$$

  Kinta intervale [-1; 1], čia -1ar 1 reiškia blogą modelį, E artėjant prie nulio prognozuojamos reikšmės artėja prie tikrųjų [7].

- Vidutinė absoliuti paklaida (MAE) . Tai visų liekanų suma. Šis kriterijus pranašesnis už MSE kai duomenyse yra kritinių reikšmių, nes MSE atveju skaičiuojant kvadratus tos reikšmės yra dar labiau išpučiamos, taip gaunant netikslumus prognozuojant [7].

2.10. Analizės išvados

- Išnagrinėjus įvairias prognozuojančios duomenų gavybos metodikas, palyginimui pasirinkti trys regresinės analizės būdai, kadangi jie pateikia lengvai suprantamus ir interpretuojamus modelius, kurie gali būti taikomi daugelyje gyvenimo sričių.

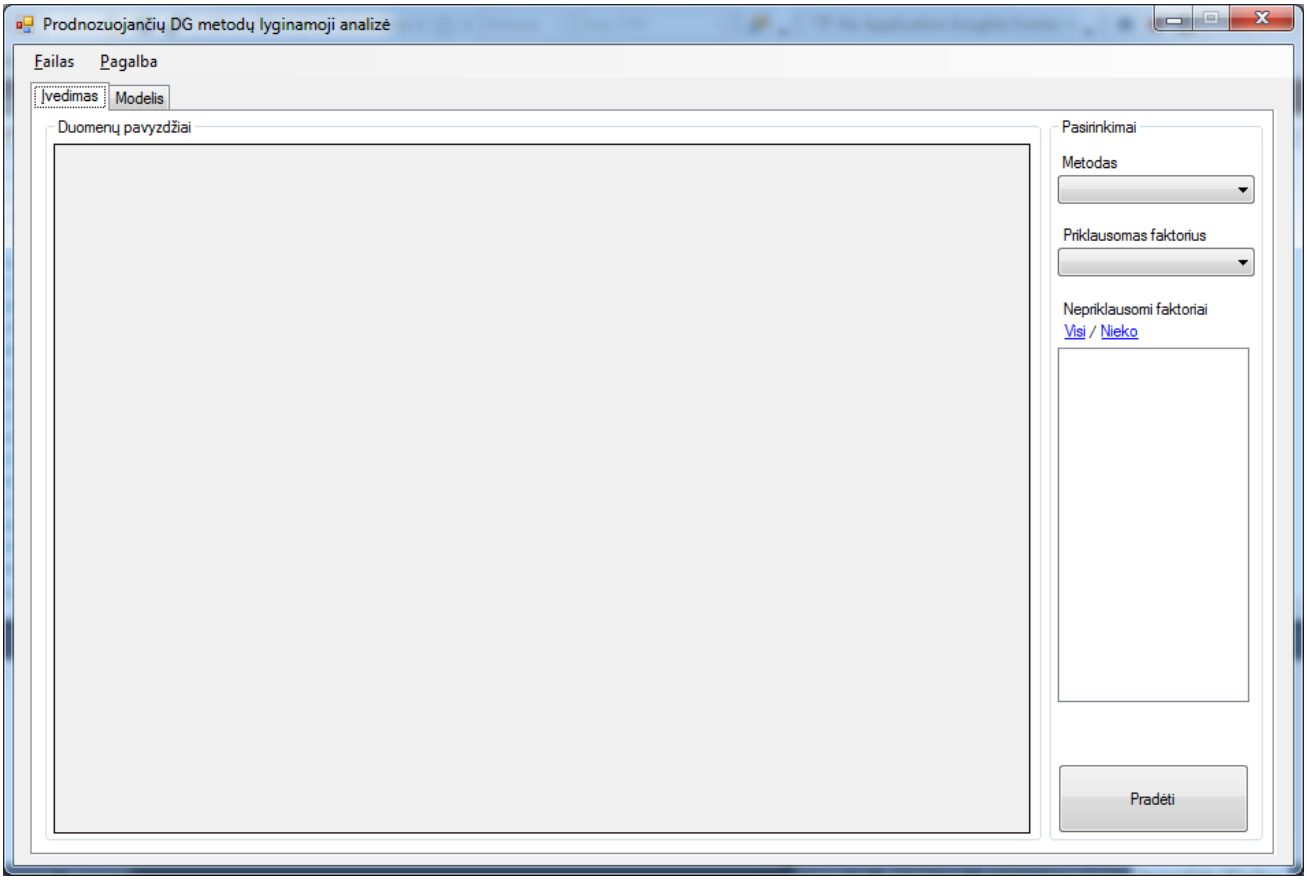
- Mažiausių kvadratų metodas, ieško regresijos kreivės minimizuojant liekanų kvadratų vidurkį. Šis matodas praktikoje yra plačiai naudojamas, dėl savo sąlyginio realizavimo ir skaičiavimų paprastumo.

- Polinominė regresija, tiriamą ryšį tarp tiriamo dydžio ir *m* priklausomų faktorių aprašo *m+1* eilės polinomu. Dideliam nepriklausomų faktorių kiekiui šio metodo modelio išraiška gali tapti ganėtinai sudėtinga, taip pat šis metodas jautresinis statistinėm anomalijom.

- Mažiausių absoliutinių nuokrypių metodo idėja paprasčiausia iš trijų, tačiau nėra anlizinių formulių modelio koeficientams rasti, todėl modelis kuriamas iteraciniu procesu, parenkant geriasius *m+1* duomenų įrašų, jais remiantis gaunama regresinė kreivė.

3. Projektinė dalis

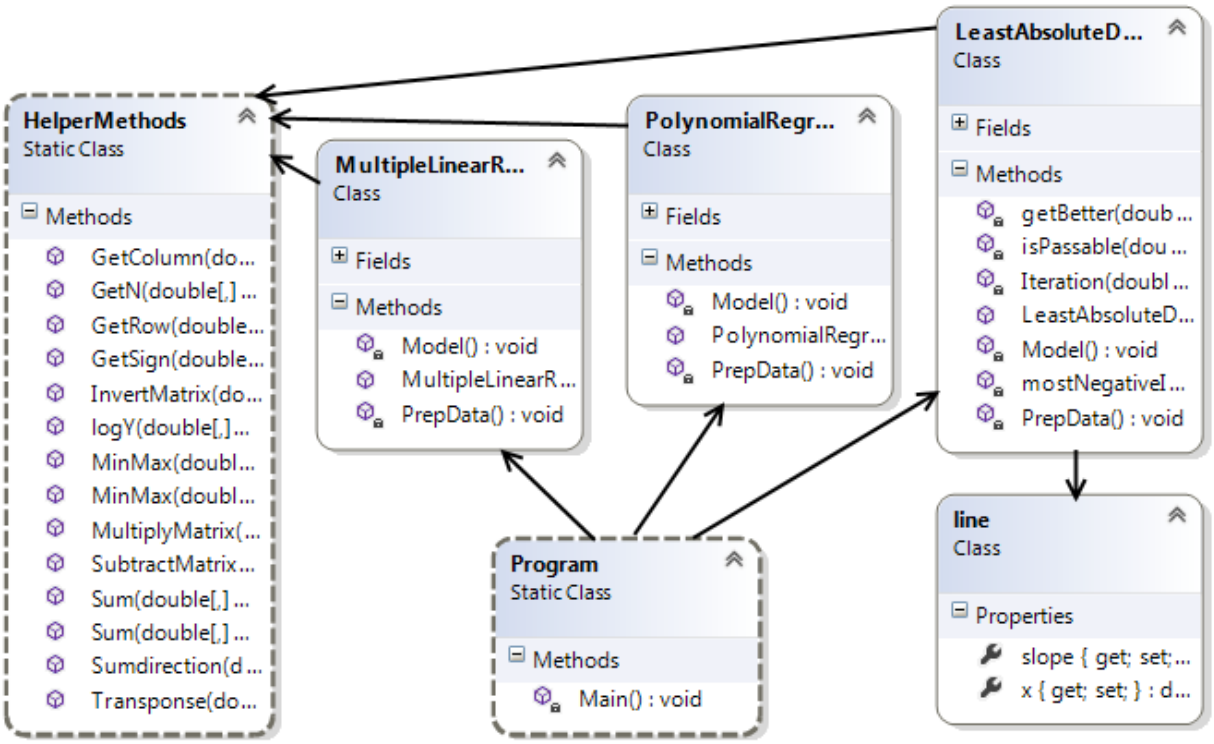
Tyrimui atlikti sukurtas įrankis. Programa kurta C# kalba, Visual Studio 2013 aplinkoje, naudojant .Net Framework 4.5.1 versiją, tinkama naudoti Windows platformoje. Paruošta programos Release versija, programos paleidžiamasis failas: Prognozavimo_metodų_lyginamoji_analizė.exe. Programa iš pateiktų duomenų sukuria tris modelius, priklausomai nuo vartotojo įvestų parametrų.



4 pav. *Programos pradinis langas*

3.1. Architektūra

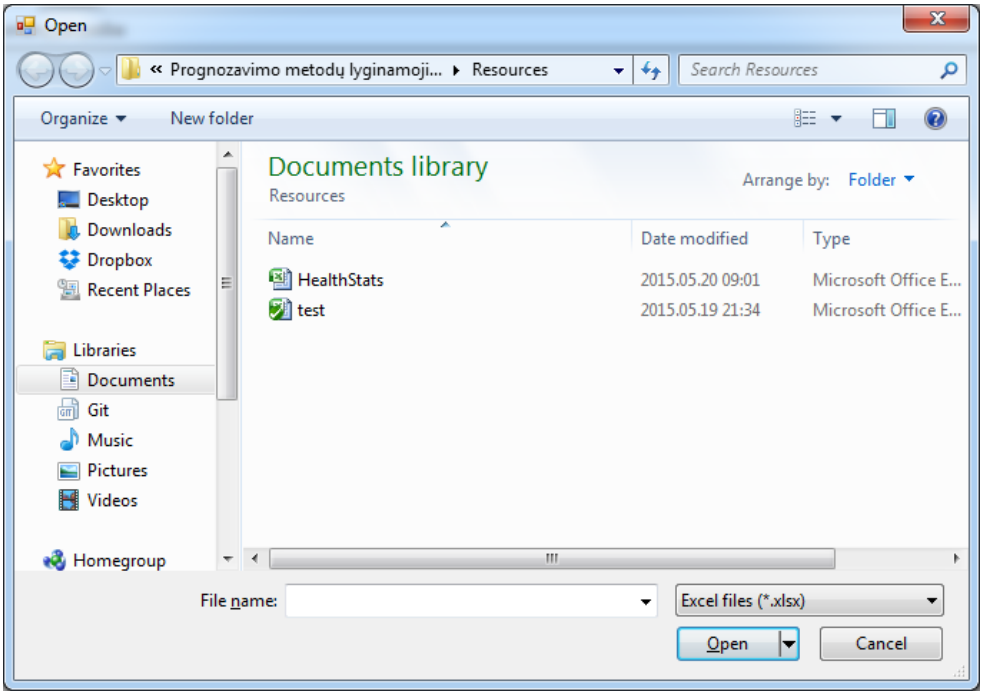
Kadangi visų trijų modelių kūrimui buvo naudojama tiesinė algebra, beveik visi kintamieji išreikšti matricomis, taip palengvinami veiksmai, kaip matricų transponavimas, daugyba, t.t. Kiekvienas modelis turi savo klasę: MultipleLinearRegression, PolynomialRegression, LeastAbsoluteDeviationsregRession, visos jos naudoja HelperMethods klasę, kurioje realizuoti įvairūs matricinius veiksmus atliekantys metodai, pateikta *5 pav***.** Programos klasių diagrama. Pagrindinė klasė kuria klasių objektus, priklausomai nuo vartotojo užklausų.



5 pav. *Programos klasių diagrama*

3.2. Vartotojo gidas

Atidarius programą pradiniame lange menu juostoje, Failas →Atidaryti skiltyje nurodomas duomenų failas *pav.6*. Priimami Microsoft Office Excel formatų .xls ir .xlsx failai. Pirmoje failo eilutėje turėtų būti stulpelių pavadinimai, kitose eilutėse - duomenys.

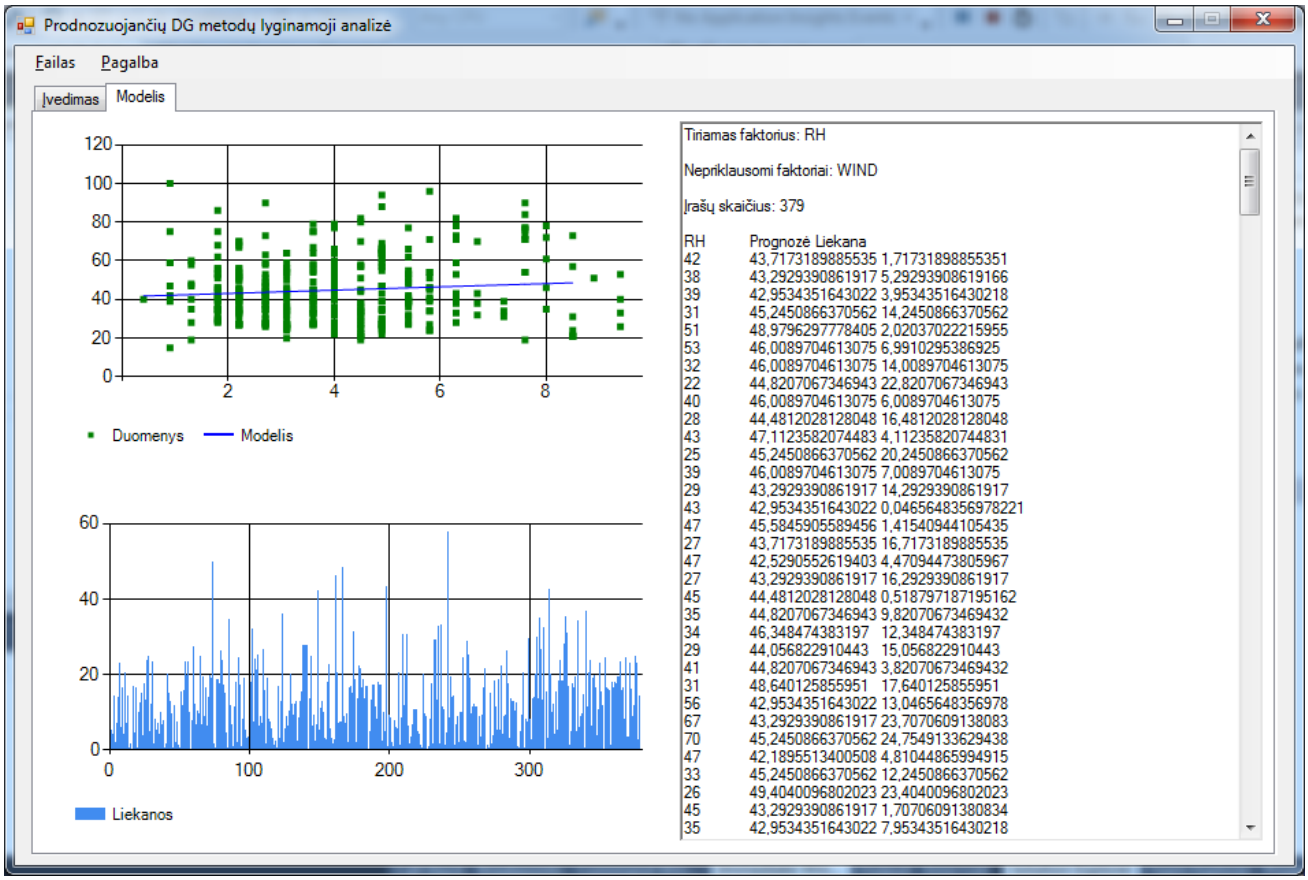


6 pav. *Duomenų įvedimas*

Pasirinkus duomenų failą, programos langas turėtų užsipildyti duomenimis. Belieka pasirinkti modeliavimo parametrus:

- Metodas – modeliavimo metodas visada bus trys galimi variantai, pasirenkamas vienas iš jų.
- Priklausomas faktorius – įvedus duomenis sąrašas užpildomas duomenų stulpelių pavadinimais. Pasirenkamas vienas priklausomas faktorius, kurio reikšmę prognozuosime.
- Nepriklausomi faktoriai – pasirinkus priklausomą faktorių sąrašas užpildomas likusiais faktoriais. Pasirenkamas vienas ar daugiau nepriklausomų faktorių.

Pasirinkus visus paminėtus parametrus, spragtelim mygtuką „Pradėti“, atlikus skaičiavimus programa peradresuos į rezultatų langą *7 pav.* Rezultatų langas



7 pav. *Rezultatų langas*

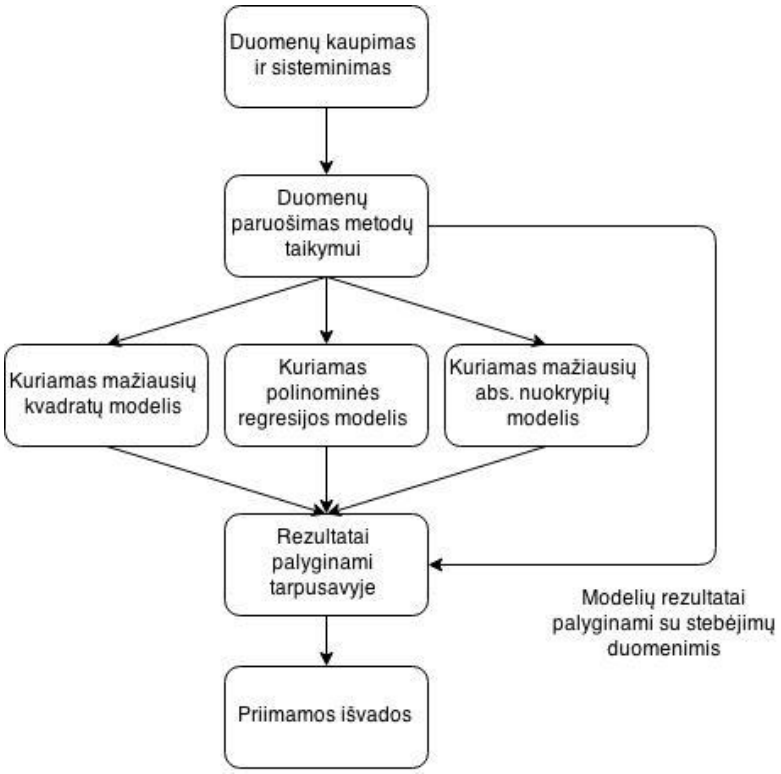
3.3. Projektinės dalies išvados

- Sukurta įrankis, gana lankstus. Iš pasirinkto duomenų failo nuskaičius duomenis programos lange galima pasirinkti tiriamą dydį bei faktorius, kurių įtaką bandysime nagrinėti. Realizuota nemažai pagalbinių metodų, kurie palengvintų darbą bandant plėsti programos funkcionalumą.

- Programa tinkama sudarinėti nesudėtingus modelius, nusakančius priklausomybes tarp duomenų įrašų, bei prognozuoti tos pačios duomenų populiacijos artimos ateities įrašus. Nepaisant to programa neatlieka jokios išankstinės duomenų analizės ar metodų taikymo korektiškumo pateiktiems duomenims, tuo turėtų pasirūpinti vartotojas, atlikdamas tyrimą ar duomenys tinkami regresinei analizei.

4. Tiriamoji dalis

Tyrimas pradedamas pasirenkant duomenų rinkinį kuriam taikysime sukurtus modelius. Duomenys paruošiami modelių taikymui, remiantis technikomis aprašytomis skyriuose 2.2, 2.3 ir 2.4. Pertvarkius duomenis ir paruošus duomenis kuriami regresiniai modeliai. Gavus rezultatus įvertinama, kuriam modeliui geriausiai sekėsi aprašyti turimus duomenis, bandoma nustatyti kitus pranašumus ar silpnasias vietas.



8 *pav.* *Tyrimo procesas*

Eksperimentas bus atliekamas su duomenų rinkiniu, kuriame saugoma statistika apie miškuose kilusius gaisrus. Duomenų rinkinyje saugoma informacija apie išdegusį miško plotą ir įvairūs meteorologiniai išmatavimai gaisro dieną. Iš viso duomenų rinkinio parinkti trys tarpusavyje nepriklausomi faktoriai: TEMP, RH ir WIND, įtakojantys priklausomą faktorių AREA, duomenų pavyzdys *5 lentelė***.** Tyrimo duomenų pavyzdys*je*. Duomenų rinkinyje viso - 379 įrašai, pilnas duomenų rinkinys pateiktas 7.2 priede. AREA stulpelio įrašai iškreipti link 0, todėl jam pritaikyta *log(n+1)* transformacija. Stulpelių paaiškinimai:

- TEMP – temperatūra gaisro metu, C°
- RH – santykinė oro drėgmė, procentais
- WIND – vėjo greitis, *km/h*
- AREA – gaisro metu sudegęs miško plotas

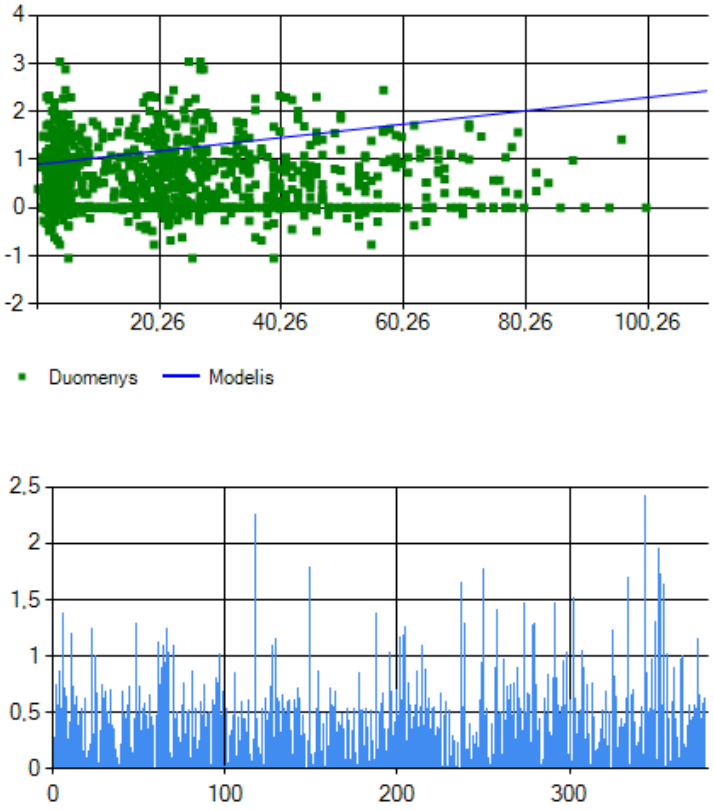
5 lentelė. *Tyrimo duomenų pavyzdys*

TEMP	RH	WIND	log(AREA)
18	42	2,7	-0,443697499
21,7	38	2,2	-0,366531544
21,9	39	1,8	-0,327902142
23,3	31	4,5	-0,259637311
21,2	51	8,9	-0,214670165
16,6	53	5,4	-0,148741651
23,8	32	5,4	-0,113509275
27,4	22	4	-0,045757491
13,2	40	5,4	-0,022276395
24,2	28	3,6	-0,017728767

4.1. Rezultatai

Keliame hipotezę, kad išdegęs miško plotas gali priklausyti nuo temperatūros, oro drėgmės ir vėjo greičio. Naudojantis įrankiu aprašytu 3 skyriuje, gaisrų statistikos duomenų rinkiniui sukuriami trys modeliai. Toliau pateikiami rezultatai.

Įvykdžius mažiausių kvadratų tyrimą, gauname modelį su liekanomis, pateiktą *9 pav*. Mažiausių kvadratų duomenys, modelis ir liekanos, tikrų stebėjimų ir modelio teikiamų rezultatų palyginimą *6 lentelė*. Mažiausių kvadratų duomenys, prognozuojama reikšmė ir liekana ir stebėjimų – įverčių palyginimą, pateikta *10 pav***.** Stebėjimai ir mažiausių kvadratų modelio prognozės



9 pav. *Mažiausių kvadratų duomenys, modelis ir liekanos*

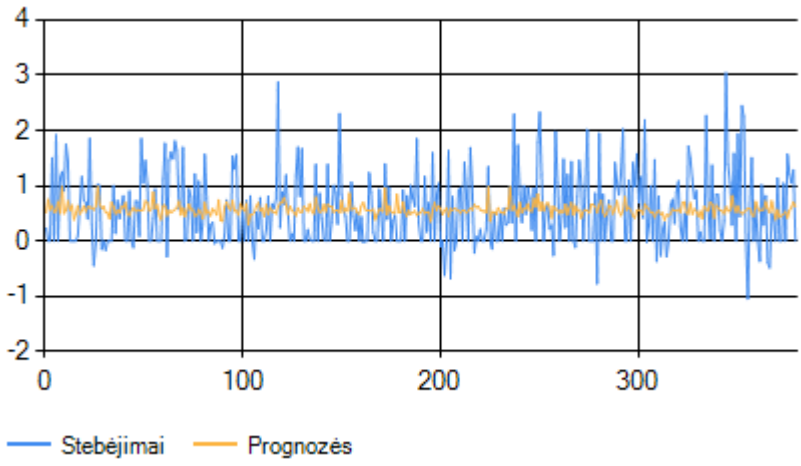
Gaunamas modelis:

$$\log(AREA) = 1.0303130528559259 - 0.029344051543351259T \cdot TEMP - 0.00205531732325554 \cdot \mathrm{RH} - 0.01908705224681351 \cdot \mathrm{WIND}$$

6 lentelė. *Mažiausių kvadratų duomenys, prognozuojama reikšmė ir liekana*

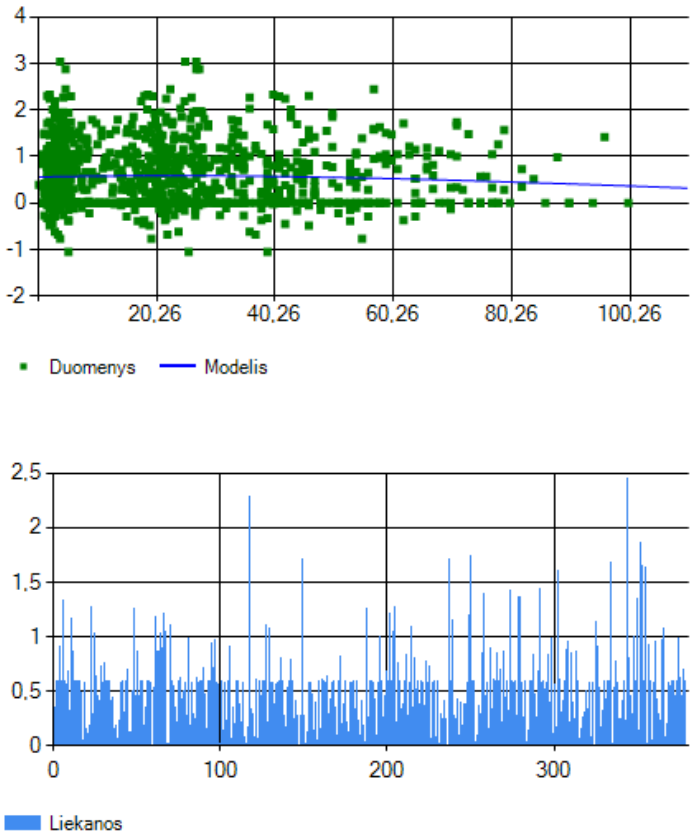
log(AREA)	Prognozė	Liekana
0,235528447	0,515709544	0,280181097
0	0,752326592	0,752326592
0	0,566391378	0,566391378
1,501333179	0,642294582	0,859038597
0	0,527809994	0,527809994
1,917768002	0,535250029	1,382517974
0	0,71479586	0,71479586
1,145817714	0,499041108	0,646776606
1,25163822	0,98204688	0,26959134
0,904174368	0,497165241	0,407009127
1,748498127	0,546662095	1,201836031
1,457276186	0,723671205	0,733604981
0	0,561127586	0,561127586
0	0,646370165	0,646370165
0	0,385259989	0,385259989
0	0,489224536	0,489224536
0,100370545	0,640739196	0,540368651
0,643452676	0,487281144	0,156171533
1,163459552	0,543520437	0,619939115
0,730782276	0,638045518	0,092736758
0,694605199	0,531054369	0,16355083
0,399673721	0,615998372	0,216324651
1,85308953	0,609480285	1,243609245

0,301029996	0,603524881	0,302494886
-0,443697499	0,562072163	1,005769662
-0,045757491	0,627644848	0,673402338
1,030599722	0,98204688	0,048552842
1,000434077	0,668909844	0,331524233
-0,148741651	0,590059997	0,738801648
0	0,627473891	0,627473891
-0,167491087	0,518032686	0,685523774
0	0,502963094	0,502963094
0	0,399175488	0,399175488
0	0,70016947	0,70016947
1,000867722	0,623835636	0,377032085
0,139879086	0,493212489	0,353333403
0,731588765	0,560957335	0,17063143
0,40654018	0,504214755	0,097674574
0,656098202	0,622104024	0,033994178
0,818225894	0,602514217	0,215711676
0,004321374	0,69339363	0,689072256
0	0,491317342	0,491317342
0,903089987	0,457507588	0,445582399
0	0,557102883	0,557102883
-0,119186408	0,60707969	0,726266097
0,718501689	0,534965574	0,183536115
0,71432976	0,57416519	0,14016457
0,089905111	0,540416253	0,450511141
1,849787824	0,557159628	1,292628197
1,049992857	0,561852599	0,488140258



10 pav. *Stebėjimai ir mažiausių kvadratų modelio prognozės*

Įvykdžius polinominės regresijos tyrimą, gauname modelį su liekanomis, pateiktą *11 pav.* Polinominės regresijos duomenys, modelis ir liekanostikrų stebėjimų ir modelio teikiamų rezultatų palyginimą *7 lentelė***.** Polinominės regresijos duomenys, prognozuojama reikšmė ir liekana**6 lentelė.** Mažiausių kvadratų duomenys, prognozuojama reikšmė ir liekana ir stebėjimų – įverčių palyginimą, pateikta *12 pav.* Stebėjimai ir polinominės regresijos modelio prognozės



11 pav. *Polinominės regresijos duomenys, modelis ir liekanos*

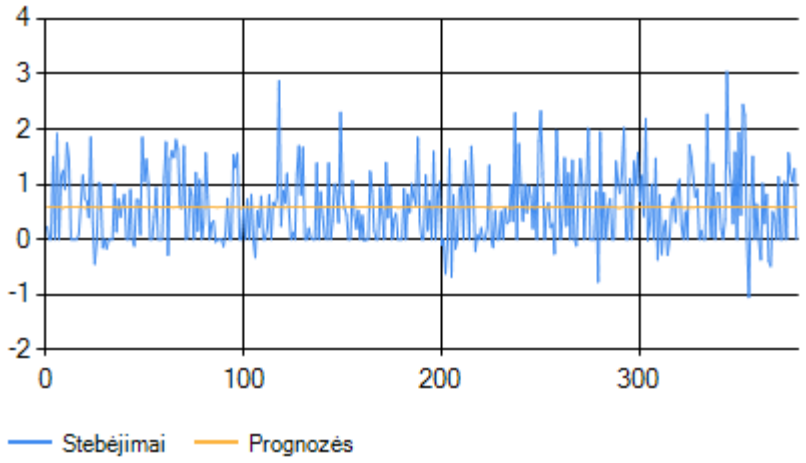
Gaunamas modelis:

$$\log(AREA) = 0.34742272326290213 - 0.0049958582324322372 \cdot TEMP + 0.00007061823176911067 9 \cdot \mathrm{RH}^2 + 0.000000495016216038701 23 \cdot \mathrm{WIND}^3$$

7 lentelė. *Polinominės regresijos duomenys, prognozuojama reikšmė ir liekana*

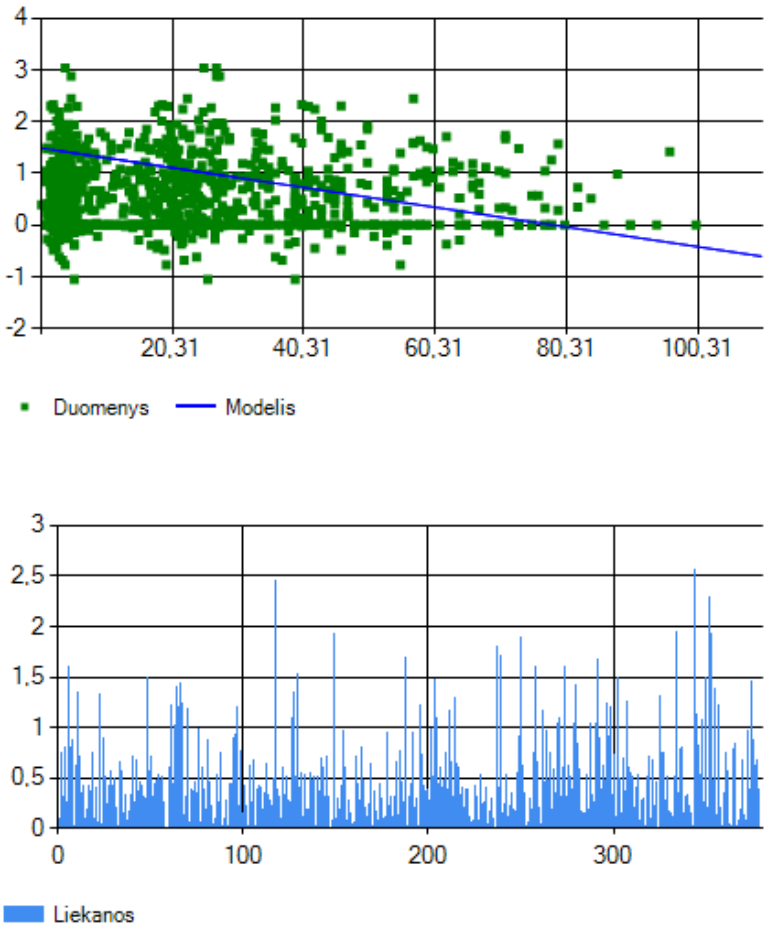
log(AREA)	Prognozė	Liekana
0,235528447	0,588504185	0,352975738
0	0,588139187	0,588139187
0	0,587069149	0,587069149
1,501333179	0,588399346	0,912933833
0	0,588180495	0,588180495
1,917768002	0,588527777	1,329240225
0	0,587248566	0,587248566
1,145817714	0,586196332	0,559621383
1,25163822	0,572100999	0,679537221
0,904174368	0,588626884	0,315547484
1,748498127	0,588746598	1,159751529
1,457276186	0,587414926	0,86986126
0	0,587872036	0,587872036
0	0,588626884	0,588626884
0	0,588663739	0,588663739
0	0,58824466	0,58824466
0,100370545	0,588693238	0,488322693
0,643452676	0,588309694	0,055142983
1,163459552	0,58878452	0,574675032
0,730782276	0,573066026	0,15771625
0,694605199	0,588762122	0,105843077
0,399673721	0,588703441	0,189029719

1,85308953	0,587072482	1,266017048
0,301029996	0,588589256	0,28755926
-0,443697499	0,587899612	1,031597111
-0,045757491	0,587399087	0,633156577
1,030599722	0,572100999	0,458498723
1,000434077	0,58847775	0,411956327
-0,148741651	0,587190953	0,735932605
0	0,585079444	0,585079444
-0,167491087	0,588756943	0,75624803
0	0,588370514	0,588370514
0	0,587571591	0,587571591
0	0,588679007	0,588679007
1,000867722	0,588690135	0,412177586
0,139879086	0,588660514	0,448781427
0,731588765	0,572100999	0,159487766
0,40654018	0,588679007	0,182138826
0,656098202	0,588611713	0,067486489
0,818225894	0,588568945	0,229656948
0,004321374	0,587305098	0,582983725
0	0,588626884	0,588626884
0,903089987	0,588280019	0,314809968
0	0,588072146	0,588072146
-0,119186408	0,587694452	0,70688086
0,718501689	0,58564782	0,132853869
0,71432976	0,588644199	0,12568556
0,089905111	0,588214651	0,49830954
1,849787824	0,588781757	1,261006067
1,049992857	0,587468223	0,462524634



12 pav. *Stebėjimai ir polinominės regresijos modelio prognozės*

Įvykdžius pmažiausių absoliutinių nuokrypių tyrimą, gauname modelį su liekanomis, pateiktą *13 pav*. Mažiausių abs.nuokrypių regresijos duomenys, modelis ir liekanos tikrų stebėjimų ir modelio teikiamų rezultatų palyginimą *8 lentelė*. Mažiausių absoliutinių nuokrypių duomenys, prognozuojama reikšmė ir liekanair stebėjimų – įverčių palyginimą, pateikta *14 pav*. Stebėjimai ir mažiausių abs. nuokrypių modelio prognozės



13 pav. *Mažiausių abs.nuokrypių regresijos duomenys, modelis ir liekanos*

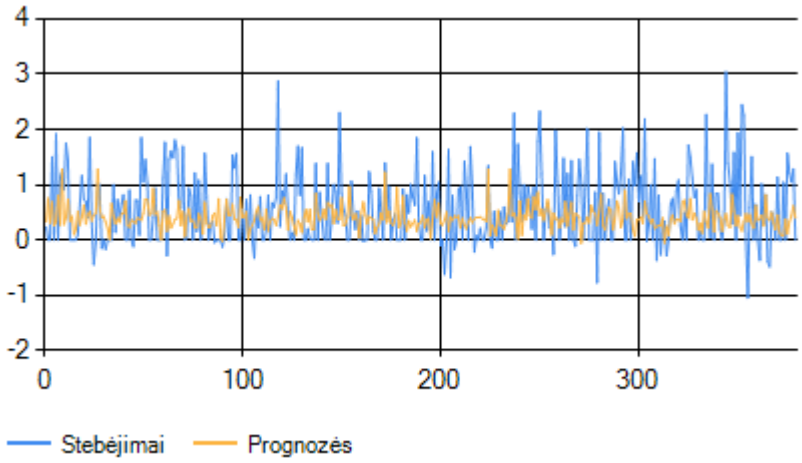
Gaunamas modelis:

$$\log(AREA) = 1.4976615878300743 - 0.02992729044635949 \cdot TEMP - 0.013609638186957217 \cdot \mathrm{RH} + 0.02445362934444207 \cdot \mathrm{WIND}$$

8 lentelė. *Mažiausių absoliutinių nuokrypių duomenys, prognozuojama reikšmė ir liekana*

log(AREA)	Prognozė	Liekana
0,235528447	0,331473041	0,095944595
0	0,756213719	0,756213719
0	0,307011089	0,307011089
1,501333179	0,690278746	0,811054433
0	0,257732666	0,257732666
1,917768002	0,319441116	1,598326886
0	0,801489325	0,801489325
1,145817714	0,272557176	0,873260539
1,25163822	1,282049499	0,030411279
0,904174368	0,278202165	0,625972203
1,748498127	0,400404194	1,348093933
1,457276186	0,741260203	0,716015983

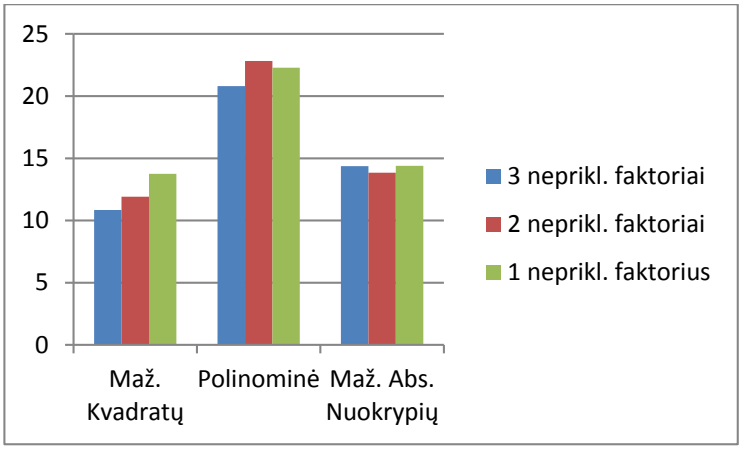
0	0,346496126	0,346496126
0	0,429072412	0,429072412
0	0,100199018	0,100199018
0	0,180841738	0,180841738
0,100370545	0,518659394	0,418288849
0,643452676	0,270764316	0,372688361
1,163459552	0,417557303	0,745902248
0,730782276	0,634667261	0,096115015
0,694605199	0,287090526	0,407514673
0,399673721	0,458286263	0,058612542
1,85308953	0,5194541	1,33363543
0,301029996	0,345257311	0,044227315
-0,443697499	0,453390355	0,897087854
-0,045757491	0,476056307	0,521813797
1,030599722	1,282049499	0,251449777
1,000434077	0,572079535	0,428354543
-0,148741651	0,411607341	0,560348992
0	0,427359249	0,427359249
-0,167491087	0,324418291	0,491909379
0	0,209699737	0,209699737
0	0,00118974	0,00118974
0	0,657748505	0,657748505
1,000867722	0,436472886	0,564394835
0,139879086	0,28801581	0,148136723
0,731588765	0,398063585	0,33352518
0,40654018	0,319530942	0,087009239
0,656098202	0,479184486	0,176913716
0,818225894	0,481767179	0,336458715
0,004321374	0,711627756	0,707306383
0	0,264592527	0,264592527
0,903089987	0,231733517	0,67135647
0	0,362842594	0,362842594
-0,119186408	0,332501818	0,451688225
0,718501689	0,300522084	0,417979605
0,71432976	0,397087804	0,317241955
0,089905111	0,383852707	0,293947595
1,849787824	0,354854002	1,494933822
1,049992857	0,48170774	0,568285117



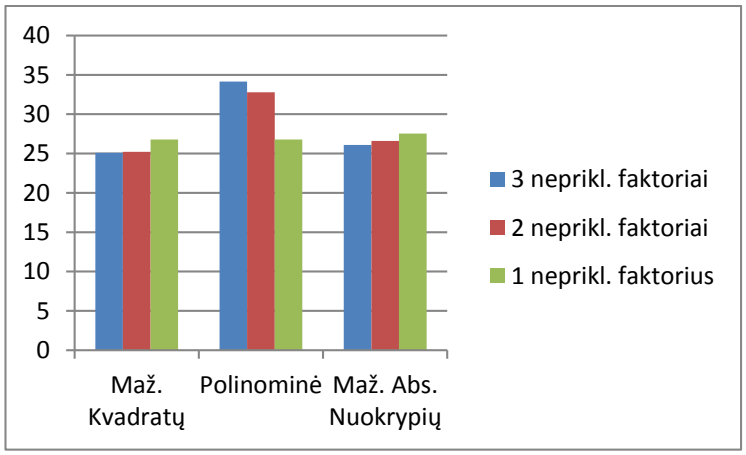
14 pav. *Stebėjimai ir mažiausių abs. nuokrypių modelio prognozės*

4.2. Palyginimas

Pirmiausia palyginame modelio tikslumą. Tai padarysime įvertinami paklaidų kvadratų vidurkį (MSE), pateikta **15 pav.** Paklaidų kvadratų vidurkiai pagal metodą ir faktorių skaičių, 50 duomenų įrašųir *16 pav.* Paklaidų kvadratų vidurkiai pagal metodą ir faktorių skaičių, 379 duomenų įrašai

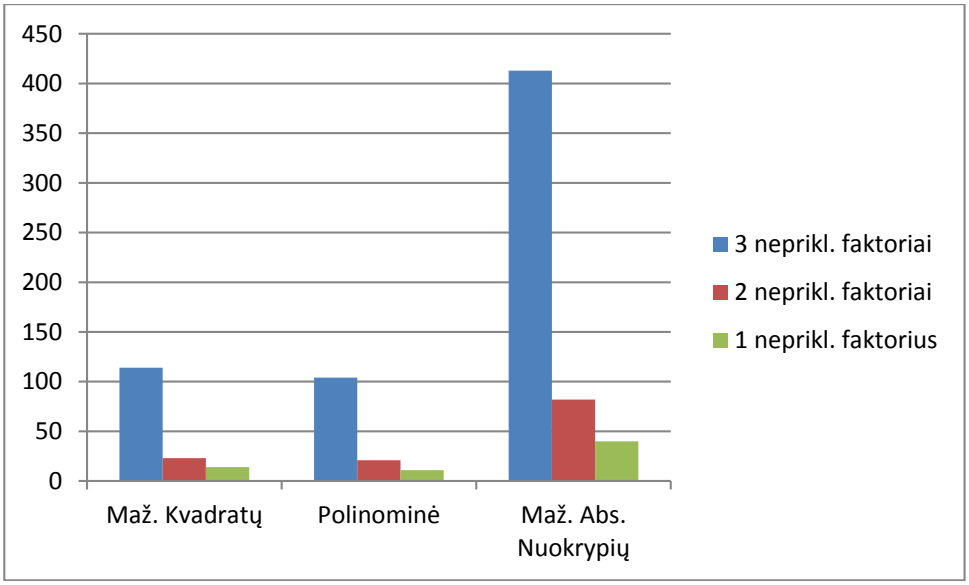


15 pav. *Paklaidų kvadratų vidurkiai pagal metodą ir faktorių skaičių, 50 duomenų įrašų*



16 pav. *Paklaidų kvadratų vidurkiai pagal metodą ir faktorių skaičių, 379 duomenų įrašai*

Toliau atliekamas metodų greičio tyrimas, rezultatai *17 pav.* Skaičiavimų laikas pagal metodą ir faktorių skaičių



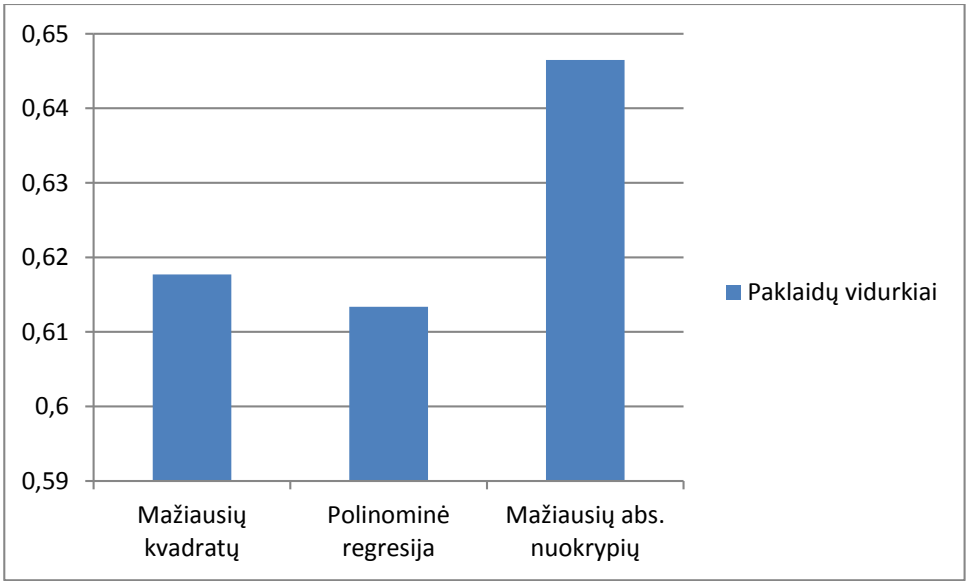
17 pav. *Skaičiavimų laikas pagal metodą ir faktorių skaičių*

Atskyrus 50 mokomųjų duomenų įrašų, sukuriami trys modeliai. Jais remiantis prognozuojamos kitos 50, tos pačios populiacijos reikšmių. Mokomieji duomenys ir prognozių rezultatai pateikti *9 lentelė.* Pronozavimo rezultatai.palyginimas *18 pav*. Prognozių paklaidų vidurkiai

9 lentelė. *Pronozavimo rezultatai.*

TEMP	RH	WIND	log(AREA)	Mažiausių kv.	Paklaida	Polinominė	Paklaida	Maž. Abs.	Paklaida
22,2	45	3,6	0	0,217672441	0,2176724	0,37953968	0,3795396	0,28232530	0,2823253
32,6	26	3,1	0,4424797	-0,03891114	0,4813909	0,23231041	0,2101693	-01129969	0,5554766
11,8	31	4,5	0	0,534446673	0,5344466	0,35638082	0,3563808	0,34655751	0,3465575
8,8	35	3,1	0,0413926	0,640979431	0,5995867	0,38998125	0,3485885	0,45172231	0,4103296
11	46	5,8	1,4369573	0,502278986	0,9346783	0,44199304	0,9949642	0,50733578	0,9296215
17,8	56	1,8	0,2900346	0,358534471	0,0684998	0,47995810	0,1899235	0,48686072	0,1968261
17,6	46	3,1	0,8475726	0,360143287	0,4874293	0,40893854	0,4386341	0,38600364	0,4615690
25,9	41	3,6	0	0,11732072	0,1173207	0,33676233	0,3367623	0,16855931	0,1685593
21	32	3,1	0	0,289147954	0,2891479	0,31483751	0,3148375	0,17835719	0,1783571
14,2	58	4	0	0,418070907	0,4180709	0,51407294	0,5140729	0,57022820	0,5702282
15,1	64	4	1,1458177	0,379329357	0,7664883	0,56126922	0,5845484	0,61219157	0,5336261
21	42	2,2	0,8633228	0,285773128	0,5775497	0,36708553	0,4962373	0,28167175	0,5816511
30,2	22	4,9	0	0,005379159	0,0053791	0,23078526	0,2307852	-0,1119585	0,1119576
20,7	46	2,7	1,4797192	0,276811548	1,2029076	0,39344638	1,0862728	0,32568750	1,1540317
24,2	27	3,1	0	0,205523576	0,2055235	0,27801839	0,2780183	0,06458024	0,0645802
28,3	26	3,1	1,8068580	0,087268282	1,7195897	0,25379260	1,5530654	-0,0272107	1,8340645
17,3	80	4,5	0	0,27234384	0,2723438	0,71299616	0,7129961	0,72616817	0,7261681
19,3	39	3,6	0,1931246	0,315102094	0,1219775	0,35843608	0,1653114	0,28026446	0,0871398
21,4	44	2,7	-0,1674911	0,260381347	0,4278724	0,37723799	0,5447290	0,29174845	0,4592395
11,8	88	4,9	0,9872192	0,409658764	0,5775604	0,83539742	0,1518218	0,91426001	0,0729592
20,7	37	2,2	1,2355284	0,30485293	0,9306755	0,34069008	0,8948383	0,23772421	0,9978042
16,8	43	3,1	0,7656685	0,38978448	0,3758840	0,39408016	0,3715883	0,37200490	0,3936636
18,9	64	4,9	0	0,250643614	0,2506436	0,54231151	0,5423115	0,53292806	0,5329280
13,8	24	5,8	0	0,465332623	0,4653326	0,31925256	0,3192525	0,23176742	0,2317674
18,9	41	3,1	1,0145205	0,332272606	0,6822479	0,37172499	0,6427955	0,31013409	0,7043864
23,4	40	6,3	0	0,141201125	0,1412011	0,34363258	0,3436325	0,19810470	0,1981047
21,1	71	7,6	0,3364597	0,120164438	0,2162953	0,59821392	0,2617541	0,54859519	0,2121354
32,4	27	2,2	0	-0,0179193	0,0179193	0,23704287	0,2370428	-0,095571	0,0955713
19,4	19	1,3	1,5013331	0,397174256	1,1041589	0,27599734	1,2253358	0,08735107	1,4139821
30,6	28	3,6	0,3159703	0,006122802	0,3098475	0,24993725	0,0660330	-0,055037	0,3710075
21,3	44	4,5	1,0856472	0,228959058	0,8566882	0,37777294	0,7078743	0,28684613	0,7988011
16,1	44	4	1,6934631	0,391091652	1,3023714	0,40373798	1,2897251	0,39250864	1,3009544

17,7	25	3,1	2,1899953	0,400370545	1,7896247	0,30314717	1,8868481	0,17429026	2,0157050
25,9	41	3,6	0	0,11732072	0,1173207	0,33676233	0,3367623	0,16855931	0,1685593
24,1	50	4	0	0,144007336	0,1440073	0,4035998	0,4035998	0,29281806	0,2928180
18,9	34	7,2	1,5360531	0,268402914	1,2676502	0,33482044	1,2012327	0,22451714	1,3115360
18	42	2,7	-0,4436975	0,364261756	0,8079592	0,38207757	0,8257750	0,33960957	0,7833070
21,5	28	4,5	1,1942367	0,255975325	0,9382614	0,29542157	0,8988151	0,12307054	1,0711662
16,8	47	4,9	1,1017470	0,347206517	0,7545405	0,41954621	0,6822008	0,40505378	0,6966932
29,2	30	4,9	0,2900346	0,018280672	0,2717539	0,26515831	0,0248763	-	0,3021483
20,9	66	4,9	1,1858253	0,187844876	0,9979804	0,55068054	0,6351448	0,51299870	0,6728266
12,2	78	6,3	0	0,391752444	0,3917524	0,71623835	0,7162383	0,80104898	0,8010489
24,8	28	1,8	1,1550322	0,210674995	0,9443572	0,27889302	0,8761392	0,06757756	1,0874546
30,8	30	4,9	0,9339931	-0,02866981	0,9626629	0,25716493	0,6768282	-0,044035	0,9780289
22,2	48	1,3	0	0,255406709	0,2554067	0,39922016	0,3992201	0,32109846	0,3210984
15,2	31	8,5	0,2878017	0,358328688	0,0705269	0,33965380	0,0518520	0,26339556	0,0244061
24,2	27	3,1	0,8182258	0,205523576	0,6127023	0,27801839	0,5402075	0,06458024	0,7536456
17,4	43	6,7	0,0293837	0,303464661	0,2740808	0,39121678	0,3618330	0,34623924	0,3168554
30,2	25	4,5	0,4393326	0,006848028	0,4324846	0,24072930	0,1986033	-0,080468	0,5197978
22,1	37	3,6	-0,67778	0,237049385	0,9148300	0,33371371	1,0114944	0,20442776	0,8822084
25,1	27	4	3,0377610	0,161935582	2,8758254	0,27353905	2,764222	0,04317538	2,9945856



18 pav. *Prognozių paklaidų vidurkiai*

5. Išvados

Palyginus modelių teikiamus rezultatus, buvo nustatyta, kad mažiausių kvadratų modelis duomenų rinkinį aprašė tiksliausiai. Palyginus su kitais metodais paklaidų kvadratų vidurkis 31% mažesnis už polinominės regresijos ir vidutiniškai 9% už mažiausių absoliutinių nuokrypių.

Atlikus algoritmų skaičiavimų laiko palyginimą, nustatyta, kad polinominės regresijos modeliavimo laikas trumpiausias, 15% greitesnis nei mažiausių kvadratų ir 3,8 karto greitesnis už mažiausių absoliutinių nuokrypių. Pastarajame metode didžiausią laiko dali užima geriausių taškų atranka, priklausomai nuo duomenų rinkinio ar parinktų pradinių taškų, skaičiavimų laikas gali drastiškai kisti.

Įvertinus prognozavimo rezultatus, nustatyta, kad nors ir nežymiai, tiksliausiai prognozuoja polinominės regresijos modelis, paklaidos 2% mažesnės nei mažiausių kvadratų ir 6% nei mažiausių absoliutinių nuokrypių.

Norint sudaryti realų modelį, rekomenduojama išbandyti į modelį įtraukti skirtingus parametrus, patikrinti ar netrūksta duomenų įrašų, įsitikinti, kad duomenyse nėra nelogiškų įrašų.

Tyrimo metu pastebėta, kad taikant regresinius modelius duomenų rinkiniams ne visada gaunami tenkinantys rezultatai, todėl prieš taikant duomenų gavybos metodikas duomenis reikia paruošti apdorojimui.

6. Literatūros sąrašas

1. *Jiawei Han, Micheline Kamber, Jian Pei* (2011) *„Data Mining (Third Edition)“*

2. *Joshua ZHexue Huang, Longbing Cao Jaideep Srivastava* (1996), *„Advances in Knowledge Discovery and Data Mining.“*

3. *David Birkes ir Yadolah Dodge* (2011) *„Alternative Methods Of Regression“*

4. *Daniel T. Larose* (2006) *„Data Mining Methods and Models“*

5. *Simonoff, Jeffrey S*. (1998) „*Smoothing Methods in Statistics“*, 2nd edition

6. *Leonidas Sakalauskas* (2009) „*Duomenų gavyba“*, Paskaitų konspektas

7. *Victoria Garment, Ways to Test the Accuracy of Your Predictive Models.* [žiūrėta 2015 05 20], prieiga per internetą*: http://www.utdallas.edu/~herve/Abdi-PLS-pretty.pdf*

8. *Ph.D. Gary R. Waissi personal page.* [žiūrėta 2015 05 20], prieiga per internetą*: http://www.public.asu.edu/~gwaissi/*

7. Priedai

7.1. Programos kodas

MultipleLinearRegression.cs:

```
class MultipleLinearRegression
    {
        public MultipleLinearRegression(string resp, string[] pred, DataTable data, int rows, int cols)
        {
            Response = resp;
            Predictors = pred;
            Data = data;

            n = rows;
            m = cols;

            Y = new double[n, 1];
            Ypred = new double[n, 1];
            e = new double[n, 1];
            b = new double[m, 1];
            X = new double[n, m + 1];

            PrepData();
            Model();
        }

        public string Response;
        public string[] Predictors;
        private DataTable Data;

        private int n;
        private int m;

        public double[,] Y, X, Ypred, e, b;
        public double s2, r2;

        private void Model()
        {

            b =
HelperMethods.MultiplyMatrix(HelperMethods.MultiplyMatrix(HelperMethods.InvertMatrix(HelperMethods.MultiplyMat
rix(HelperMethods.Transponse(X), X)), HelperMethods.Transponse(X)), Y);

            Ypred = HelperMethods.MultiplyMatrix(X, b);

            e = HelperMethods.SubtractMatrixAbs(Y, Ypred);

            r2 = Math.Pow((n * HelperMethods.Sum(Y, Ypred) - HelperMethods.Sum(Y) * HelperMethods.Sum(Ypred))
/
                (Math.Sqrt(n * HelperMethods.Sum(Y, Y) - Math.Pow(HelperMethods.Sum(Y), 2)) * Math.Sqrt(n *
HelperMethods.Sum(Ypred, Ypred) - Math.Pow(HelperMethods.Sum(Ypred), 2))), 2);

            s2 = HelperMethods.Sum(e, e);
        }

        private void PrepData()
        {
            for (int i = 0; i < n; i++)
            {
                Y[i, 0] = Double.Parse(Data.Rows[i][Response].ToString());
                X[i, 0] = 1;

                for (int j = 0; j < m; j++)
                    X[i, j + 1] = Double.Parse(Data.Rows[i][Predictors[j]].ToString());
            }
        }
        }
```

PolynomialRegression.cs:

```
class PolynomialRegression
    {
        public PolynomialRegression(string resp, string[] pred, DataTable data, int rows, int cols)
        {
            Response = resp;
            Predictors = pred;
            Data = data;

            n = rows;
            m = cols;

            Y = new double[n*m, 1];
            Ypred = new double[n, 1];
            e = new double[n*m, 1];
            b = new double[3, 1];
            X = new double[n*m, 3];

            PrepData();
            Model();
        }

        public string Response;
        public string[] Predictors;
        private DataTable Data;

        private int n;
        private int m;

        public double[,] Y, X, Ypred, e, b;
        public double s2, r2;

        private void PrepData()
        {
            int j = 0, k = 0;
            for (int i = 0; i < n * m; i++)
            {
                if (i % n == 0 && i != 0)
                    k++;

                j = i % n;

                Y[i, 0] = Double.Parse(Data.Rows[j][Response].ToString());

                X[i, 0] = 1;
                X[i, 1] = Double.Parse(Data.Rows[j][Predictors[k]].ToString());
                X[i, 2] = Math.Pow(X[i, 1], 2);
            }
        }

        private void Model()
        {
            b =
HelperMethods.MultiplyMatrix(HelperMethods.MultiplyMatrix(HelperMethods.InvertMatrix(HelperMethods.MultiplyMat
rix(HelperMethods.Transponse(X), X)), HelperMethods.Transponse(X)), Y);

            Ypred = HelperMethods.MultiplyMatrix(X, b);

            e = HelperMethods.SubtractMatrixAbs(Y, Ypred);

            s2 = HelperMethods.Sum(e, e);
        }
            }
```

LeastAbsoluteDeviations.cs:

```
class LeastAbsoluteDeviationsRegression
    {
        public LeastAbsoluteDeviationsRegression(string resp, string[] pred, DataTable data, int rows, int
cols)
        {
            Response = resp;
            Predictors = pred;
```

```
            Data = data;

            n = rows;
            m = cols;

            Y = new double[n, 1];
            Ylog = new double[n, 1];
            Ypred = new double[n, 1];

            W = new double[n, 1];
            Z = new double[n, 1];

            e = new double[n, 1];
            b = new double[m, 1];
            X = new double[n, m + 1];

            PrepData();
            Model();
        }

        public string Response;
        public string[] Predictors;
        private DataTable Data;

        private int n;
        private int m;

        public double[,] Y, X, Z, W, Ypred, Ylog, e, b;
        public double s2, r2;

        private void PrepData()
        {
            for (int i = 0; i < n; i++)
            {
                Y[i, 0] = Double.Parse(Data.Rows[i][Response].ToString());
                X[i, 0] = 1;

                for (int j = 0; j < m; j++)
                    X[i, j + 1] = Double.Parse(Data.Rows[i][Predictors[j]].ToString());
            }

            Ylog = HelperMethods.logY(Y);
        }

        private void Model()
        {
            double[,] iterationResults;
            int[] indexes = new int[] { 0, 1, 2, 4 };

            do
            {
                iterationResults = Iteration(HelperMethods.GetN(X, indexes), HelperMethods.GetN(Ylog,
indexes));
                indexes[mostNegativeIndex(iterationResults)] = getBetter(W, Z);
            }
            while (!isPassable(iterationResults));

            double sum;


            for (int i = 0; i < Ypred.GetLength(0); i++)
            {
                sum = b[0,0];

                for (int j = 1; j < b.GetLength(0); j++)
                    sum += b[j, 0] * X[i, j];

                Ypred[i, 0] = sum;
            }

            Ypred = HelperMethods.MultiplyMatrix(X, HelperMethods.Transponse(b));

            e = HelperMethods.SubtractMatrixAbs(Y, Ypred);
            // end
        }

        private double[,] Iteration(double[,] X4, double[,] Y4)
        {
```

```
            double[,] directionSums = new double[4,2];

            int[] signs = new int[n];

            double[,] directionMatrix = HelperMethods.InvertMatrix(X4);

            b = HelperMethods.MultiplyMatrix(HelperMethods.InvertMatrix(X4), Y4);

            for (int j = 0; j < 4; j++)
            {
                W[j, 0] = 1;

                for (int i = 4; i < n; i++)
                {
                    Z[i, 0] = Ylog[i, 0] - HelperMethods.MultiplyMatrix(HelperMethods.Transponse(b),
HelperMethods.Transponse(HelperMethods.GetRow(X, i)))[0, 0];
                    W[i, 0] = HelperMethods.MultiplyMatrix(HelperMethods.GetColumn(directionMatrix, j),
HelperMethods.Transponse(HelperMethods.GetRow(X, i)))[0, 0];
                    signs[i] = HelperMethods.GetSign(Z[i, 0], W[i, 0]);
                }

                directionSums[j, 0] = HelperMethods.Sumdirection(W, signs);
                directionSums[j, 1] = -(directionSums[j, 0] - 1) + 1;

                W[j, 0] = 0;
            }

            return directionSums;
        }

        private bool isPassable(double[,] a)
        {
            for (int i = 0; i < a.GetLength(0); i++)
                if (Math.Abs(a[i, 0]) > 15)
                    return false;

            return true;
        }

        private int getBetter(double[,] W, double[,] Z)
        {
            line[] slopes = new line[W.GetLength(0)];

            for (int i = 0; i < W.GetLength(0); i++)
                if (W[i, 0] != 0)
                    slopes[i] = new line { slope = (Z[i, 0]) / (W[i, 0]), x = W[i, 0] };
                else
                    slopes[i] = new line { slope = 0, x = 0 };

            slopes = slopes.OrderBy(x => x.slope).ToArray();

            double sum = 0;

            for (int i = 0; i < slopes.Length; i++)
                sum += Math.Abs(slopes[i].x);

            int bestIndex = 0;
            double temp = 0;

            while (temp < sum / 2)
            {
                temp += Math.Abs(slopes[bestIndex].x);
                bestIndex++;
            }

            return bestIndex;
        }

        private int mostNegativeIndex(double[,] a)
        {
            int index = 0;
            double min = a[0,0];

            for (int i = 0; i < a.GetLength(0); i++)
                if (a[i, 0] < min || a[i, 1] < min)
                {
                    min = Math.Min(a[i, 0], a[i, 1]);
```

```
                index = i;
            }

        return index;
    }
}

public class line{

    public double slope { get; set; }
    public double x { get; set; }
        }
```

HelperMethods.cs

```
public static class HelperMethods
    {
        public static double[,] Transponse(double[,] matrix)
        {
            int rows = matrix.GetLength(0);
            int cols = matrix.GetLength(1);

            double[,] temp = new double[cols, rows];

            for (int i = 0; i < rows; i++)
                for (int j = 0; j < cols; j++)
                    temp[j, i] = matrix[i, j];

            return temp;
        }

        public static double[,] MultiplyMatrix(double[,] a, double[,] b)
        {
            double[,] c = new double[a.GetLength(0), b.GetLength(1)];

            if (a.GetLength(1) == b.GetLength(0))
            {
                for (int i = 0; i < c.GetLength(0); i++)
                {
                    for (int j = 0; j < c.GetLength(1); j++)
                    {
                        c[i, j] = 0;
                        for (int k = 0; k < a.GetLength(1); k++)
                            c[i, j] = c[i, j] + a[i, k] * b[k, j];
                    }
                }
            }

            return c;
        }

        public static double[,] InvertMatrix(double[,] a)
        {
            Matrix<double> A = DenseMatrix.OfArray(a);
            return A.Inverse().ToArray();
        }

        public static double[,] SubtractMatrixAbs(double[,] a, double[,] b)
        {
            double[,] c = new double[a.GetLength(0), a.GetLength(1)];

            for (int i = 0; i < a.GetLength(0); i++)
            {
                c[i, 0] = Math.Abs(a[i, 0] - b[i, 0]);
            }
            return c;
        }

        public static double Sum(double[,] a)
        {
            double sum = 0;
            for (int i = 0; i < a.GetLength(0); i++)
            {
                sum += a[i, 0];
            }
```

```
    return sum;
}

public static double Sum(double[,] a, double[,] b)
{
    double sum = 0;
    for (int i = 0; i < a.GetLength(0); i++)
    {
        sum += a[i, 0] * b[i, 0];
    }

    return sum;
}

public static double[,] logY(double[,] a)
{
    double[,] temp = new double[a.GetLength(0), 1];

    for (int i = 0; i < a.GetLength(0); i++)
    {
        temp[i, 0] = Math.Log(a[i, 0]);
    }

    return temp;
}

public static double[] MinMax(double[,] X)
{
    double min = X[0, 1];
    double max = X[0, 1];

    for (int i = 0; i < X.GetLength(0); i++)
        for (int j = 1; j < X.GetLength(1); j++)
        {
            if (X[i, j] < min)
                min = X[i, j];
            if (X[i, j] > max)
                max = X[i, j];
        }

    return new double[] { min, max };
}

public static double[] MinMax(double[,] X, int index)
{
    double min = X[0, 1];
    double max = X[0, 1];

    for (int i = 0; i < X.GetLength(0); i++)
        {
            if (X[i, index] < min)
                min = X[i, index];
            if (X[i, index] > max)
                max = X[i, index];
        }

    return new double[] { min, max };
}

public static double[,] GetN(double[,] a, int[] index){

    double[,] temp = new double[index.Length, a.GetLength(1)];

    for (int i = 0; i < index.Length; i++)
    {
        for (int j = 0; j < a.GetLength(1); j++)
        {
            temp[i, j] = a[index[i], j];
        }
    }

    return temp;
}

public static double[,] GetRow(double[,] a, int index)
{
    double[,] temp = new double[1, a.GetLength(1)];
```

```
        for (int j = 0; j < a.GetLength(1); j++)
        {
            temp[0, j] = a[index, j];
        }

        return temp;
    }

    public static double[,] GetColumn(double[,] a, int index)
    {
        double[,] temp = new double[1, a.GetLength(0)];

        for (int j = 0; j < a.GetLength(0); j++)
        {
            temp[0, j] = a[j, index];
        }

        return temp;
    }

    public static int GetSign(double a, double b)
    {
        if (b == 0)
            return 0;
        else
            if (a / b > 0)
                return 1;
            else
                return -1;
    }

    public static double Sumdirection(double[,] a, int[] signs)
    {
        double sum = 0;

        for (int i = 0; i < signs.Length; i++)
        {
            if(signs[i]>0)
                sum -= Math.Abs(a[i, 0]);
            else
                sum += Math.Abs(a[i, 0]);
        }

        return sum;
    }
}
```

7.2. Tyrimo duomenys

Gaisrų statistika**:**

TEM	RH	WIN	log(AREA)
18	42	2,7	-0
22	38	2,2	-0
22	39	1,8	-0
23	31	4,5	-0
21	51	8,9	-0
17	53	5,4	-0
24	32	5,4	-0
27	22	4	-0
13	40	5,4	-0
24	28	3,6	-0
17	43	6,7	0
24	25	4,5	0
23	39	5,4	0,1
25	29	2,2	0,1
25	43	1,8	0,2
20	47	4,9	0,2
30	27	2,7	0,2
16	47	1,3	0,2
29	27	2,2	0,2
18	45	3,6	0,2
21	35	4	0,2
19	34	5,8	0,2
16	29	3,1	0,2
20	41	4	0,3
15	31	8,5	0,3
18	56	1,8	0,3
18	67	2,2	0,3
5,3	70	4,5	0,3
17	47	0,9	0,4
23	33	4,5	0,4
15	26	9,4	0,4
21	45	2,2	0,4
22	35	1,8	0,4
17	50	4	0,4
20	39	5,4	0,4
18	39	2,2	0,5
14	53	1,8	0,5
20	39	4,9	0,7
5,8	54	5,8	0,7
19	44	2,7	0,7
18	45	2,2	0,7
14	66	5,4	0,7
24	32	6,7	0,7
19	32	4	0,7
12	53	2,2	0,8
17	45	4,5	0,8
21	34	4,9	0,8
18	46	3,1	0,8
12	39	5,8	0,9
21	42	2,2	0,9
13	42	0,9	0,9
12	60	4	0,9
12	33	4	0,9
24	28	2,7	0,9
25	22	4,5	0,9
24	25	4	1
25	22	4,5	1
24	36	5,4	1
5,8	54	5,8	1
22	15	0,9	1
14	59	6,3	1,1
23	38	3,6	1,1
22	33	2,2	1,1
12	54	3,6	1,1
8,8	68	2,2	1,1
20	37	2,7	1,1
15	64	4	1,1
22	34	1,8	1,2
23	31	7,2	1,2
21	37	2,2	1,2
20	33	6,3	1,3
23	26	4,9	1,4
18	25	3,1	1,4
5,1	96	5,8	1,4
20	47	4,9	1,4
11	46	5,8	1,4
17	27	4,9	1,5
17	27	4,9	1,5
17	60	1,3	1,5
12	73	6,3	1,5
19	19	1,3	1,5
15	27	3,1	1,5
16	59	3,1	1,5
19	49	3,6	1,6
11	46	5,8	1,6
13	79	3,6	1,6
15	57	4,5	1,6
23	39	4,9	1,7
16	44	4	1,7
20	34	4,5	1,8
28	26	3,1	1,8
16	43	4	1,9
26	21	4,5	1,9
28	27	3,1	2
19	43	2,7	2
24	36	3,1	2
18	25	3,1	2,2
20	41	5,8	2,3
18	46	1,8	2,3
19	40	2,2	2,3
25	27	4	3
13	75	1,8	0
15	51	2,7	0
17	20	3,1	0
15	66	4	1
22	73	7,6	0
22	54	7,6	0,5
27	38	6,3	-0
26	39	5,4	-1
21	70	2,2	-0
29	28	2,7	0
22	40	0,4	0,4
27	25	3,1	-0
24	36	3,1	-1
22	37	3,6	-1
21	38	2,7	0,2
19	41	3,1	1
22	46	4	0
24	41	2,2	0,9
21	44	2,7	-0
21	59	0,9	0
24	40	1,8	0,1
28	32	4	0,9
11	84	7,6	0,5
21	42	3,1	0,6
19	39	3,6	0,2
22	53	3,1	0,8
22	54	7,6	-0
19	55	4	-1
24	24	3,1	0
21	32	3,1	0
19	53	2,7	0,6
22	56	3,1	-0
20	58	4,5	1
20	47	4	0,5
4,8	57	8,5	1
5,1	61	8	1
5,1	61	4,9	0,7
4,6	21	8,5	1,3
4,6	21	8,5	1
4,6	21	8,5	1,3
4,6	21	8,5	1
2,2	59	4,9	1
5,1	24	8,5	1,4
4,2	51	4	0
8,8	35	3,1	0
7,5	46	8	1,4
23	40	6,3	0
13	90	7,6	0
22	49	2,7	0
24	32	1,8	0
24	30	1,8	0
19	53	1,8	0
25	39	0,9	0,9
23	40	1,3	0,4
27	28	5,4	1,9
17	67	3,6	0,8
22	48	1,3	0
14	46	1,8	-0
15	45	2,2	0
20	43	4,9	0
11	90	2,7	0
21	25	4,9	0
19	39	5,4	0,5
19	38	4,5	0
19	38	4,5	0
11	94	4,9	0
19	52	2,2	0
17	53	5,4	-0
24	35	3,6	0,7
16	45	1,8	0
25	27	2,2	0
25	27	2,7	0
25	28	1,8	1,2
12	78	6,3	0
24	27	4,9	0
20	41	1,8	0,2
19	30	2,7	0
19	24	5,8	0
19	24	4,9	0,6
22	27	2,2	0
22	28	6,3	0,6
19	34	7,2	1,5
17	28	4	0,9
17	28	4	0
13	39	2,7	0,3
14	56	1,8	0,6
24	27	3,1	0
24	27	3,1	0
21	32	2,2	0
20	35	1,8	0
24	27	4	0,5
24	27	3,1	0,8
22	28	4,5	1,2
17	41	2,2	1
18	54	3,1	0,3
18	51	5,4	0
9,8	86	1,8	0
19	44	2,2	0
23	34	2,2	1,7
23	35	2,2	0,9
20	41	1,8	0,2
19	44	2,2	0,6
16	51	2,2	0
21	37	1,8	0
16	51	4,5	0,3
12	66	4,9	0,8
17	43	3,1	0,8
21	35	2,2	1,5
10	75	3,6	0
17	57	4,5	0
13	64	3,6	0,2
10	75	3,6	0,6
15	53	6,3	0,9
21	43	3,6	0,3
20	47	2,7	0,2
19	50	2,2	0,8
21	35	4,9	1,1
21	35	4,9	0,1
16	55	3,6	0
20	39	2,7	0
21	39	2,2	0,9
18	42	2,2	0
17	45	4	0,6
15	64	3,1	-0
16	53	2,2	0,5
21	35	2,7	0,8
20	45	3,1	1,3
16	38	5,4	0,2
16	27	3,6	0
17	47	4,9	1,1
14	77	7,6	0
14	77	7,6	1
14	58	4	0
10	75	0,9	0
20	42	2,7	0
10	78	4	1,3
15	57	4,9	1,6
21	54	2,2	0

22	42	2,2	2,2
8,7	51	5,8	0
5,2	10	0,9	0
19	39	7,2	0,9
16	63	2,7	1,2
28	29	1,8	0,8
21	58	2,7	1,6
21	44	4,5	1,1
21	50	2,2	1,2
21	55	5,4	1,4
12	48	5,4	0
23	34	3,1	1,5
23	34	3,1	0
7,5	71	6,3	1
21	46	2,7	1,5
22	43	4	1,8
15	19	7,6	0
5,3	68	1,8	0
10	62	1,8	1,7
20	55	4,9	0,6
24	33	3,6	0,6
26	32	3,1	0
28	34	4,5	0
28	34	4,5	0,9
23	46	4	0,7
25	36	4	0
21	41	3,6	0
22	34	2,2	0,8
28	27	2,2	0
17	67	4,9	0,6
14	46	4	0
20	44	3,1	0,9
23	31	5,4	0
15	42	2,7	0
8,2	53	9,4	0,7
23	27	4,5	0,2
26	33	3,6	0
24	50	4	0
28	27	4,9	2,9
26	39	3,1	0,8
14	24	5,8	0
25	42	5,4	0,4
25	36	4	0,5
26	36	4,5	2,3
31	19	4,5	0
29	27	3,6	0,8
22	48	4	-0
27	31	3,6	0,7
20	56	2,2	0
20	56	2,2	0
28	33	2,2	0,4
26	34	5,8	0
25	44	4	0,5
19	45	3,6	0
23	40	4	0,8
24	38	6,7	0
21	66	4,9	1,2
22	45	3,6	0
24	51	1,8	0
27	35	1,3	-0
14	73	2,7	0
24	53	4	0,8
19	46	2,2	-0
16	58	3,6	0
26	29	1,8	0,1
11	64	3,1	0,5
15	78	8	0
16	58	3,6	1
17	80	4,5	0
19	70	2,2	0
8,9	35	8	0
11	77	4	0
19	61	4,9	0
23	49	5,4	0,8
12	88	4,9	1
18	65	4	0
17	54	3,1	0
17	56	3,1	0
18	48	2,7	0
17	59	2,7	0
20	50	4	1,9
19	64	4,9	0,5
16	72	8	0,3
19	64	4,9	0
19	64	4,9	0
15	76	7,6	0,6
4,6	82	6,3	0,7
5,1	77	5,4	0,3
4,6	59	0,9	0,8
10	45	5,8	0,5
11	41	5,4	0,7
13	27	3,6	0,8
14	33	9,4	1,8
18	27	5,8	0
18	40	4	1,6
14	79	4	0,3
25	50	3,1	1,8
26	35	2,7	1
23	40	9,4	0,5
27	28	1,3	0,2
26	45	4	0,9
18	82	4,5	0,3
23	57	4,9	2,4
30	25	4,5	0,4
30	22	4,9	0
23	40	5,8	0,1
31	27	5,4	0
33	25	4	1,4
31	28	3,6	0,3
24	43	6,3	0,3
26	34	3,6	1,2
19	71	7,6	1,7
21	58	1,3	0
29	33	4	0
32	21	4,5	0
32	27	2,2	0
28	29	4,5	1,6
31	30	4,9	0,9
24	42	2,2	0
33	26	3,1	0,4
32	27	2,2	1,2
33	26	2,7	1,6
27	63	4,9	1
22	65	4,9	0
22	65	4,9	0
21	69	4,9	0
29	30	4,9	0,3
29	29	4,9	1,7
27	35	1,8	0,8
19	73	8,5	0
26	41	3,6	0
26	41	3,6	0
21	71	7,6	0,3
18	62	5,4	-0
28	35	2,7	0
28	32	2,7	0,8
22	71	5,8	1,7
21	70	6,7	1
26	42	4	0
12	31	4,5	0