LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTIKAS - STATISTIKAS NODAĻA

PROGNOZĒŠANA AR KLASTERIZĀCIJAS METODI

KURSA DARBS

Autors: **Anete Rubine**

Stud. apl. ar08242

Darba vadītājs: doc. Dr.math. Jānis Valeinis

RĪGA 2012

Saturs

Ievads

Laika rindu analizēšana un nākotnes vērtību prognozēšana ir viena no nozīmīgākajām problēmēm, ar kuru saskaras datu analītiķi daudzās nozarēs, sākot no finansēm un ekonomikas lēdz ražošanas darbības vadīšanai vai telekomunikācijām. *Prognozēšana* ir kāda nākotnes rezultāta paredzēšana, un tā problēmas tiek klasificētas kā:

• īslaicīgās prognozēšanas problēmas ietver īsa laika perioda nākotnes vērtību prognozēšanu (dienas, nedēļas, mēneši);

• vidēji ilgās prognozēšanas problēmas ietver laika periodu prognozēšanu no viena līdz diviem gadiem;

• ilglaicīgās prognozēšanas problēmas ietver laika periodu prognozēšanu, kas var pārsniegt divus gadus.

Laikrindu dati var tikt definēti pēc interesējošā mainīgā novērojumu hronoloģiskas secības. Prognozēšana ir svarīga ne tikai valsts iestādēm un lieliem uzņēmumiem, bet arī sabiedrībai, jo precīzākas prognozes tiek veiktas, jo cilvēkam ir lielāka iespēja saplānot savu personīgo bužetu. Mums ir svarīgi laicīgi zināt par gāzes vai elektroenerģijas cenu pieaugumu. Savukārt uzņēmumiem ir svarīgi zināt pieprasījumu.

Mans mērķis ir veikt elektroenerģijas cenu un pieprasījuma prognozēšanu. Mūsdienās ir daudzas metodes kā to veikt, populārākā no tādām ir prognozēšana izmantojot ARMA modeļus. Savukār es šajā darbā aplūkošu PSF algoritmu (*Pattern Sequence - based Forecasting*), kas ir interesants ar to, ka spēj prognozēt arī datu "izlecēju" vērtības. PSF algoritmam ir sava īpatība, tas nestrādā ar ar reāliem datiem, bet gan ar to apzīmētajiem datiem, kurus mēs iegūstam klasterizācijas ceļā.

Vispirms iepazīsimies ar darba teorētisko pusi, kurā tiks apskatīta klasterizācija un PSF algoritms. Pēc tam tiks veikta praktiskā daļa programmā R, kurā tiks prognozētas elektroenerģijas cenas un pieprasījums.

1. Klasterizācija

Klasterizācija ir process, kura laikā laikrindas vērtības tiek aizstātas ar klasteriem-apzīmētām vērtībām. Pirmais solis, kas jāveic, ielādējam reālos datus programmā R, un varam sākt klasterizācijas procesu. Sākumā aprēķinam *Silhouette indeksu* pēc formulas:

$$silh(i) = \frac{a(i) - b(i)}{\max\{a(i), b(i)\}},$$

kur vidējais attālums objektam i, $(i \in A)$, starp pārējiem objektiem, kas atrodas kopā A, apzīmē $a(i)$, un vidējais attālums objektam i starp visiem pārējiem objektiem, kas atrodas klasterī $C \neq A$ tiek apzīmēts ar $d(i, C)$. Visiem klasteriem, kam $C \neq A$, $d(i, C)$ tiek aprēķinātas vērtības un mazākā tiek izvēlēta sekojošā veidā:

$$b(i) = \min_{C \neq A} d(i, C), i \in A.$$

Vērtība $b(i)$ parāda atšķirību objektam i no tā tuvākā kaimiņa klastera. $Silh(i)$ vērtība var būt no -1 līdz $+1$, kur $+1$ un -1 parāda vai objekts i pieder vai nepieder attiecīgajam klasterim. Ja silhouette indeksa i vērtība pieder klasterim A un tā vērtība ir tuva nullei, tas nozīmē, ka objekts i var piederēt tuvākajam A kaimiņa klasterim. Ja klasters A ir kopa tikai ar vienu elementu, tad silhouette indekss objektam i nav definēta, šādā gadījumā tā vērtība ir vienāda ar nulli. Mērķa funkcija ir vidējā $silh(i)$ vērtība un vislabākā klasterizācija ir sasniegta, kad $silh(i)$ tiek maksimizēta. Programmā R, lai šo indeksu aprēkinātu, tiek lietota *manhattan* metode, kas nodrošina to, ka distance, starp klastera centru un datu punktiem, tiek aprēķināta kā koordinašu attālumu absolūto vērtību summa.

$$d[jk] = sum(abs(x[ij] - x[ik])).$$

Tālāk tiek veikta datu normalizācija pēc formulas:

$$x_j \leftarrow \frac{x_j}{\frac{1}{N}\sum_{i=1}^{N} x_i},$$

kur x_j ir cena/pieprasījums j-tajā dienas stundā un N ir vienāds ar 24, jo katra vērtība parāda vienas stundas izmaiņas. Tagad varam veikt klasterizāciju. Šajā darbā tiek izmantots K-vērtības algoritms, kas ir optimāls cenas/pieprasījuma datu kopu klasificēšanai.

K-vērtības algoritms ir viens no vienkāršākajiem algoritmiem, kas atrisina klasterizācijas problēmu. Procedūra ir vienkāršs un ērts veids, lai klasificētu attiecīgo datu kopumu, izmantojot noteiktu skaitu klasteru (pieņemsim k klasterus), kas ir fiksēts lielums. Galvenā ideja ir definēt k lielumu, kas ir minimālais attālums līdz klastera vidum (katram klasterim tie ir atšķirīgi). Šī attāluma atrašanai tika izmantota silhouette funkcija. K-vērtības algoritma mērķis ir minimizēt mērķa funkciju, kas šajā gadījumā ir kvadrātisko kļūdu funkcija. Mērķa funkcija ir:

$$J = \sum_{j=1}^{k} \sum_{i=i}^{n} ||x_i^{(j)} - c_j||^2,$$

kur $||x_i^{(j)} - c_j||^2$ ir izvēlēto attālumu mērs starp datu punktu $x_j^{(i)}$ un klastera centru c_j, attālumu rādītājs n datu punktiem līdz attiecīgajiem klasteru centriem. Tādā veidā mēs apzīmējam datus ar attiecīgo k indeksu - klasterizācija. [1]

2. PSF algoritms

PSF algoritms tiek uzdots šādi:

PSF()

$ES_d \leftarrow \{\}$

$\widehat{X}(d) \leftarrow 0$

katrai dienai $d \in T$

$S_W^{d-1} \leftarrow [L_{d-W}, L_{d-W+1}, ..., L_{d-2}, L_{d-1}]$

katram j**, kurš** $X(j) \in D$

$S_W^j \leftarrow [L_{j-W+1}, L_{j-W+2}, ..., L_{j-1}, L_j]$

ja $(S_W^j = S_W^{d-1})$

$ES_d \leftarrow ES_d \cup j$

katram $j \in ES_d$

$\widehat{X}(d) \leftarrow \widehat{X}(d) + X(j+1)$

$\widetilde{X}(d) \leftarrow \widetilde{X}(d)/size(ES_d)$

$D \leftarrow D \triangleright \widehat{X}(d)$

$[L_1, L_2, ..., L_{d-1}, L_d] \leftarrow$klasterizācija$(D, K)$

$d \leftarrow d + 1$

atgriež $\widehat{X}(d)$ visām T dienām.

Algoritmā **D** ir datu kopa, **K** klasteru skaits, $[L_1, L_2, ..., L_{d-1}, L_{d-1}]$ apzīmētu datu kopa, **W** loga garums un **T** izmēģinājuma datu kopa (manā gadījumā $D = T$). Lai algoritms tiktu veiksmīgi pielietots, jāzin laikrindas vēsturiskās vērtības līdz dienai $d-1$, prognozēšanas mērķis ir prognozēt diens d 24 stundu cenu/pieprasījuma vērtības.
Pieņemsim, ka $X(i) \in R^{24}$ ir vektors, kas sastāv no 24 stundu elektroenerģijas cenu/pieprasījuma vērtībām kādai konkrētai dienai i

$$X(i) = [x_1, x_2, ..., x_{24}].$$

Pieņemsim, ka $L_i \in \{1, ..., K\}$ apzīmējumi cenām/pieprasījumam dienai i, kas tiek iegūti ar klasterizācijas metodi, kur K ir klasteru skaits. Pieņemsim, ka S_W^i ir apzīmēto cenu/pieprasījumu rezultāts W secīgām dienām, kas ir atpakaļ ejošs, sākot ar dienu i. Pieņemsim, ka $X(i) \in R^{24}$ ir vektors, kas sastāv no 24 stundu elektroenerģijas ce-

nu/pieprasījuma vērtībām kādai konkrētai dienai i

$$S_W^i = [L_{i-W+1}, L_{i-W+2}, ..., L_{i-1}, L_i],$$

kur loga garums W ir parametrs, kurš tiek aprēķināts sekojoši, tiek izmantoti klasterizācijas ceļā apzīmētie dati. Atrast vērtību W, nozīmē minimizēt funkciju

$$\sum_{d \in TS} ||\widehat{X}(d) - X(d)||,$$

kur $\widehat{X}(d)$ ir prognozētās vērtības dienai d, atsaucoties uz PSF algoritmu $X(d)$ ir reālās vērtības un TS ir datu kopa. Praktiski loga garuma vērtība W tiek aprēķināta, izmantojot mēnešu skaidtu (n=3), kur katrs n ir konkrēts mēnesis. Prognozes kļūdas tiek rēķinātas katrā mēnesī mainot loga garumu W. Ikmēneša kļūdas tiek aprakstītas $e_m\{W = j\}$, kur $j = 1, ..., W_{max}$, kur $W_{max} = 1$, e_m- mēnesis. Pēc tam tiek aprēķinātas kļūdas vidējās vērtības katram apskatītajam loga garumam

$$e_j = \frac{1}{n} \sum_{i=1}^{n} e_m\{W = j\},$$

kur $n = 3$ un mēneši={janvāris, februāris, marts}, e_m-mēnesis. Tiek izvēlēts tas loga garums W, kuram ir mazākā kļūda

$$W = arg\min\{e_j\}, j = 1, ..., W_{max}.$$

PSF algoritms cenas/pieprasījuma prognozēšanai dienai d vispirms meklē apzīmētos datus datu kopā, kas pilnīgi vienādi ar S_W^{d-1}, ja vienādā apakškopa ES_d ir definēta kā

$$ES_d = \{j, \text{ tādu ka} S_W^j = S_W^{d-1}\}.$$

Gadījumā, kad šāda vienādība netiek atrasta, algoritms meklē apakškopu, kas ir pilnīgi vienāda ar S_{W-1}^{d-1}. Tādā veidā loga garums, kas sastāv no apzīmētajiem datiem, samazinās par vienu vienību. Šī stratēģija garantē, ka tiks atrasta vismaz viena vienādība, kad W būs pilnīgi vienāds ar viens.

Saskaņā ar PSF algoritmu, 24 stundu prognoze cenas/pieprasījuma laika rindu vērtībām

dienai d tiek prognozētas ar vidējo vērtību dienām, kas seko pēc ES_d,

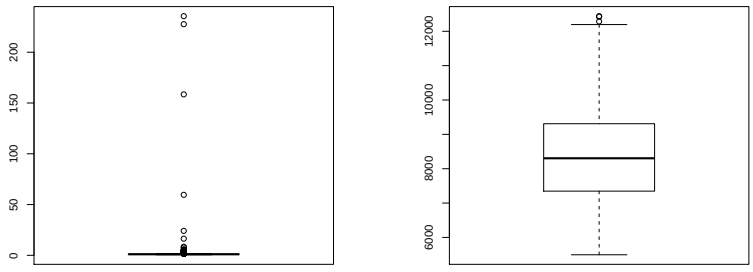
$$\widehat{X}(d) = \frac{\sum_{j \in ES_d} X(j+1)}{size(ES_d)},$$

kur $size(ES_d)$ ir elementu skaits, kas pieder kopai ES_d.

Gadījumā, kad jāveic vidēja vai gara laika perioda prognoze, visas darbības tiek attiecinātas uz visu datu kopu, un klasterizācijas process tiek atkārtots ar paplašinātām datu kopām, līdz prognozēšana ir izpildīta. [2]

3. Elektroenerģijas cenas un pieprasījuma prognozēšana

Savam darbam izmantošu 2005. gada pirmo trīs mēnešu (janvāris, februāris, marts) elektroenerģijas cenu un pieprasījuma datus no Austrālijas Nacionālā elektroenerģijas pārstāvja (ANEM). Lai pēc tam varētu pārbaudīt kursa darba "PROGNOZĒŠANA AR KLASTERIZĀCIJAS METODI"precizitāti, dati ir pieejami ikvienam Austrālijas Nacionālā elektroenerģijas pārstāvja mājas lapā `http://www.aemo.com.au/`.
Elektroenerģijas cenu un pieprasījuma laikrindas ir interesantas ar to, ka tajās ir novērojami dati "izlecēji". Tie var rasties, piemēram, dēļ neparadzētiem laikapstākļiem un dabas stihijām, kas ietekmē gan cenas paugstināšanos vai pazemināšanos, kā arī pieprasījumu. Darbā apskatīto datu vizuālā interpretācija.

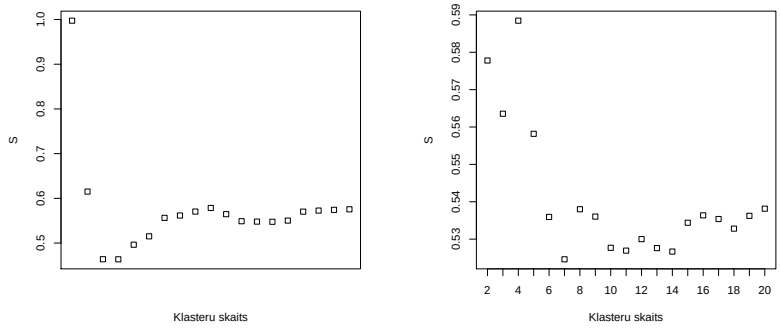


1. att.: Kreisā pusē elektroenerģijas cenas dati. Labajā pusē elektroenerģijas pieprasījuma dati .

3.1. Silhouette indekss

Aprēķināšu Silhouette indeksu :

$$silh(i) = \frac{a(i) - b(i)}{\max\{a(i), b(i)\}}.$$



2. att.: Kreisajā pusē elektroenerģijas cenas Silhouette indekss no 1 līdz 20. Labajā pusē elektroenerģijas pieprasījuma Silhouette indekss no 1 līdz 20.

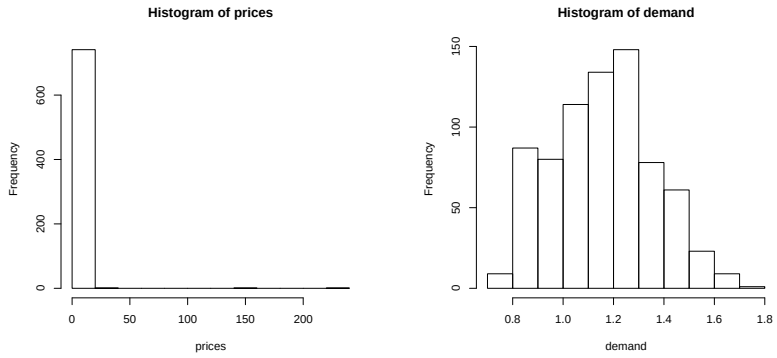
Silhouette indeks norāda, cik klasteros dati būs jādala, lai veiksmīgāk varētu veikt datu apzīmēšanu. Jāizvēlas tas klasteru skaits, kuram silhouette indekss ir vismazākais. Manā gadījumā elektroenerģijas cenas klasteru skaits būs 5, un elektroenerģijas pieprasījuma klasteru skaits būs 7.

3.2. Datu normalizācija

Pirms klasterizācijas dati vēl ir jānormalizē. Varam pieņemt, ka tendence cenas izmaiņām gada garumā ir tāda pati kā iepriekšējo gadu inflācijai, tas ir, oriģinālā tendence tiek nogludināta ar sākotnējiem datiem. Transformācija, ko pielietosim:

$$x_j \leftarrow \frac{x_j}{\frac{1}{N}\sum_{i=1}^{N} x_i},$$

kur x_j ir cena/pieprasījums j-tajā dienas stundā un N ir vienāds ar 24, jo katra vērtība parāda vienas stundas izmaiņas.



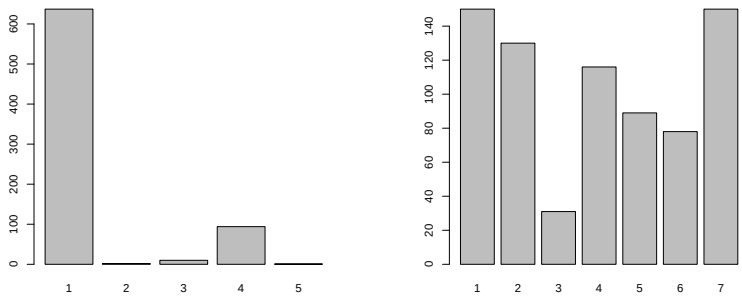
3. att.: Kreisā pusē normalizēti elektroenerģijas cenas dati. Labajā pusē normalizēti elektroenerģijas pieprasījuma dati .

3.3. Klasterizācija

Tagad esam nonākuši pie vienas no svarīgākajām daļām - klasterizācijas. Datu kopa, kas sastāv no ik stundas cenu/pieprasījumu datiem, klasterizācijas problēma ir sadalīt datus K klasteros tā, lai cenas/pieprasījuma dati sastāvētu no K klasteriem. Kā rezultātā, datubāzes izmēri tiek krasi samazināti no 24 dimensijām uz vienu dimensiju. K-vērtības algoritma mērķa funkcija ir:

$$J = \sum_{j=1}^{k} \sum_{i=i}^{n} ||x_i^{(j)} - c_j||^2,$$

kur $||x_i^{(j)} - c_j||^2$ ir izvēlēto attālumu mērs starp datu punktu $x_j^{(i)}$ un klastera centru c_j, attālumu rādītājs n datu punktiem līdz attiecīgajiem klasteru centriem. Tādā veidā mēs apzīmējam datus ar attiecīgo k indeksu - klasterizācija.



4. att.: Kreisajā pusē elektroenerģijas cenas pēc klasterizācijas. Labajā pusē elektroenerģijas pieprasījuma pēc klasterizācijas.

3.4. Prognozēšana

Tagad veikšu prognozi elektroenerģijas cenai un pieprasījumam, izmantojot PSF algoritmu, bet par loga garumu izvēlēšos fiksētus lielumus $W = 1, ..., 7$. Prognoze tiek veikta 2005.04.01 $1:00:00$, kuras reālā vērtība ir:

Reālās vērtības 2005.04.01 1:00:00	
cena	pieprasījums
20.62	7819.36

Prognozētās vērtības:

Eklektroenerģijas cena			Elektroenerģijas pieprasījums		
W	prognoze	kļūda %	W	prognoze	kļūda %
1	33.1979	38	1	7921.190	1
2	40.8081	49	2	7897.746	1
3	24.5614	16	3	9340.513	16
4	19.7878	4	4	9813.223	20
5	20.0067	3	5	9116.573	14
6	N/V	-	6	N/V	-
7	N/V	-	7	N/V	-

Redzam, ka elektroenerģijas cenas prognozēšanai optimālais loga garums $W = 5$, bet elektroenerģijas pieprasījuma prognozēšanai optimālais loga garums $W = 2$.

Secinājumi

Mērķis ir sasniegts, elektroenerģijas cenas un pieprasījums ir prognozēts. Elektroenerģijas cenai 2005.04.01 1 : 00 : 00 bija jābūt 20.62, pēc prognozēšanas tuvākā vērtība ir 20.0067. Elektroenerģijas pieprasījums 2005.04.01 1 : 00 : 00 bija jābūt 7819.36, pēc prognozēšanas tuvākā vērtība ir 7897.746. Iegūtos rezultātus var saukt par precīziem, jo kļūda nav liela, cenas prognozei tā ir tikai 3% zem patiesās vērtības, un attiecīgi pieprasījuma prognozes kļūda ir zem 1%. Kļūda ir skaidrojama ar to, ka dati ir mazā apjomā, lai precīzi paredzētu tekošās stundas vērtību, kā arī ar to, ka loga garums W netika aprēķināts, bet gan pieņemts kā fiksēts lielums.

1. Programmas kods

```
DATU IEGŪŠANA
 library(clusterSim)
j=1:744
x[j]<-scan(file="janprices.txt")
#x[j]<-data

n=31
m=24
matrix(x[j], n , m)

DATU NORMALIZĒŠANA
i=1:24
x[j]<-x[j]/(sum(x[i])/24)

n=31
m=24
matrix(x[j],n,m)

M<-matrix(x[j],n,m)

c(t(M))
data<-c(t(M))
hist(data)

SILHOUETTE INDEX
library(clusterSim)

data<-scan(file="jandemand.txt")



md <- dist(data, method="manhattan")
```

```
# nc - number_of_clusters
min_nc=2
max_nc=20
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
cl2 <- pam(md, nc, diss=TRUE)
res[nc-min_nc+1, 2] <- S <- index.S(md,cl2$cluster)
clusters <- rbind(clusters, cl2$cluster)
}
print(paste("max S for",(min_nc:max_nc)[which.max(res[,2])],"clusters=",max(res[,2]
print("clustering for max S")
print(clusters[which.max(res[,2]),])
write.table(res,file="S_res.csv",sep=";",dec=",",row.names=TRUE,col.names=FALSE)
plot(res,type="p",pch=0,xlab="Number of clusters",ylab="S",xaxt="n")
axis(1, c(min_nc:max_nc))

K-MEAN ALGORITMS DATU AIZVIETOŠANAI
kres<-kmeans(data,5) #(data,7)->jandemand
plot(data)
kmeansRes<-factor(kres$cluster) #labeled data

PSF ALGORITMS
data<-scan(file="jandemand.txt")
#funkcija no diviem mainigajiem, pati virkne, un loga garums
logi <- function(virkne, garums)
{
#mainigo deklaracijas pirms tie tiek izmantoti
mekleta_virkne <- c(); pirmslogu_vertibas<-c(); j=1;
pirmslogu_indeksi<-c();
```

```
#pirmais cikls, nem pedejos "garums" elementus no "virkne"
#saglaba tos "mekleta_virkne"
for (i in 1:garums)
{
 mekleta_virkne[garums+1-i] <- virkne[length(virkne)+1-i]
}
#otrs cikls, iet cauri visai virknei, meklejot
#virknes daļas, kas sakrīt
for (i in 1:(length(virkne)-garums-1))
{
 test <- identical(as.integer(virkne[i:(i+garums-1)]), mekleta_virkne)
 #salidzinam divas virknes
 if (test == TRUE) #ja sakrit
 {
 #ieglabajam "pirmsloga" skaitli vektora
 #pirmslogu_vertibas[j]<-virkne[(i+garums)]; j = j+1
 pirmslogu_indeksi[j]<-(i+garums); j = j+1
 #padzenam ciklu uz prieksu par atrasto virkni,
 #jo logi nevar parklaties
 i = i+garums
 }
}
#funkcija izmet ara visus "pirmslogu skaitlus"
return (pirmslogu_indeksi)
}

#darbibas piemers
#izmet ara visus pirmslogu indeksu, ja meklejam virknes ar garumu 2
indeksi<- logi(kmeansRes, 2)
iistie<-data[indeksi] #"istas" vertibas
mean(iistie)
```

Bibliogrāfija

[1] F. Martinez - Alvarez, A. Troncoso, J.C. Riquelme, J.S. Aguilar (2010). Energy time series forecasting based on pattern sequence similarity, IEEE Transactions on Knowledge and Data Engineering, in press.

[2] http://www.icors11.uva.es/book_of_abstracts.pdf

[3] R.A. Maronna, R.D. Martin, V.J. Yohai (2007). Robust Statistics: Theory and Methods. Wiley.

[4] P.J. Rousseeuw, M.Hubert (2011). Robust statistics for outlier detection, Data Mining and Knowledge Discovery, 1(1), 73-79.

[5] S.Gelper, R. Fried and C. Croux (2010). Robust forecasting with exponential and Holt-Winters smoothing, Journal of Forecasting, 29, 285-300.

[6] P. Geleano, D. Pena, R.S. Tsay (2006). Outlier detection in multivariate time series by projection pursuit, Journal of the American statistical Association, 101(474), 645-669.

[7] http://www.aemo.com.au/

KURSA darbs "PROGNOZĒŠANA AR KLASTERIZĀCIJAS METODI" izstrādāts LU Fizikas un Matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Anete Rubine

_____ _____

(paraksts) (datums)

Rekomendēju darbu aizstāvēšanai.

Vadītājs: doc. Dr.math. Jānis Valeinis

_____ _____

(paraksts) (datums)

Recenzents:

_____ _____

(paraksts) (datums)

Darbs iesniegts Matemātikas - statistikas nodaļā _____

(datums)

_____

(darbu pieņēma)

Darbs aizstāvēts kursa gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums_____

(datums)

Komisijas sekretārs/-e: _____ _____

(Vārds, Uzvārds) (paraksts)