به نام خدا

پروژه درس داده کاوی

عنوان:

یافتن بهترین مدل برای پیشگویی رده ی تعلق گرفتن یا نگرفتن یارانه به یک خانوار

استاد راهنما:

دکتر محمدرضا فقیهی حبیب آبادی

دانشجو:

سهراب فریدی 97422188

دانشکده: **علوم ریاضی**

رشته: **ریاضی کاربردی**

گرایش: **علوم داده ها**

پاییز 1398

مقدمه:

داده های من جمع آوری شده بوسیله ی مرکز آمار کشوری است و مربوط به طرح آمارگیری هزینه و درآمد خانوار های شهری در سال 1397 است. این داده شامل 72 متغییر چون درآمد، هزینه، ... از خانوار ها در شهرها ی مختلف است که در پروژه ی من 2747 خانوار که شامل استان های مرکزی: 590 خانوار، همدان: 774 خانوار، قم: 539 خانوار، قزوین: 420 خانوار و البرز با 424 خانوار است، مورد بررسی قرار خواهد گرفت.

بنا، بر آن است که تمام مراحل را در چهار فصل باز گو کنیم.

فصل اول:

1:درک مهم ترین هدف داده کاوی:

با توجه به قانون جدیدی که در کشور وضع شده است یارانه به سه دهک اول جامعه تعلق نخواهد گرفت. ما نیز قصد داریم متغییر برآمدی رسته ای تعریف کنیم که بیانگر تعلق گرفتن یا نگرفتن یارانه به خانوارها است، به عبارت دیگر: بودن یا نبودن خانوار در 3 دهک برتر اقتصادی .

2: بدست آوردن مجموعه داده های مورد استفاده در تحلیل:

این مرحله از قبل انجام شده بود و داده ها کاملا آماده در دسترس قرار دارند.

که البته می دانیم تمام این داده ها از پرسشنامه های جمع آوری شده بوسیله ی مرکز آمار کشوری بدست آمده اند:

پرسشنامه طرح هزینه و درآمد خانوارهای شهری شامل بخشهای زیر است:

- خصوصیات اجتماعی اعضای خانوار
- مشخصات محل سکونت و تسهیلات و لوازم عمده زندگی
- هزینه های خوراکی و غیرخوراکی خانوار
- درآمدهای خانوار

حال چند مفهوم و متغییر مورد استفاده را با توجه به پرسش نامه، تعریف میشود:

خانوار

خانوار از یک یاچند نفر تشکیل میشود که با هم در یک مکان زندگی میکنند و با یکدیگر هم خرج هستند . که در داده ها معادل با هر سطر یا مشاهده است.

سرپرست خانوار

یکی از اعضای خانوار که در خانوار به عنوان سرپرست شناخته میشود.

نحوه تصرف منزل مسکونی خانوار

انواع نحوه ی تصرف به شرح زیر است:

1-ملکی عرصه و اعیان : خانوار مالک زمین و بنای منزل سکونتی خود است.

2-ملکی اعیان : خانوار تنها مالک بنای منزل سکونتی خود است.

3-اجاری : خانوار منزل سکونتی خود را اجاره کرده است.

4-رهنی : خانوار منزل سکونتی خود را به ازای پرداخت مقداری پول به صورت قرض الحسنه به مالک برای مدت معینی تصرف کرده است.

5-در برابر خدمت : خانوار منزل سکونتی خود را در مقابل انجام کار یک یا چند نفر از اعضایش، تصرف کرده است.

6-رایگان: هیچ یک از اعضای خانوار مبلغ یا خدمتی را برای منزل خود نمی پردازند و نه مالک زمین و نه بنای منزل سکونتی خود، هستند.

7-سایر

تعداد اتاق

هر اتاق، فضای محصور و سقفداری است. منظور تنها اتاق خواب ها نیست.

آدرس خانوار

عبارت است از کد پستی محل سکونت هر خانوار.

باقی متغییر ها نیازی به توضیح ندارند و تنها نام آن ها را برده میشود.

تعریف متغیرها

ردیف	تعریف متغیر	نام متغیر
1	آدرس خانوار	**Address**
2	کد استان	**C.O**
3	ماه مراجعه به خانوار	**MahMorajeh**
4	فصل مراجعه به خانوار	**Fasl**
5	جنسیت سرپرست خانوار	**Jens**
6	سن سرپرست خانوار	**Sen**
7	میزان سواد سرپرست خانوار	**Savad**
8	سرپرست خانوار تحصیل می‌کند یا خیر؟	Tahsil.Mikonad
9	مدرک تحصیلی سرپرست خانوار	Madrak
10	وضعیت فعالیت سرپرست خانوار	Faaliat
11	وضعیت زناشویی سرپرست خانوار	Zanashoi
12	تعداد اعضای خانوار	tedad.a
13	نحوه تصرف منزل مسکونی	n.t.m
14	تعداد اتاق در اختیار	**t.o**
15	سطح زیر بنای محل سکونت	s.z
16	نوع اسکلت بنای محل سکونت	**n.e**
17	مصالح عمده بنای محل سکونت	m.o.b
18	اتومبیل شخصی	oto
19	موتورسیکلت	**mo**

20	دوچرخه	do
21	رادیو	radio
22	ضبط	zabt
23	تلویزیون سیاه و سفید	tv.s
24	تلویزیون رنگی	tv.r
25	انواع ویدئو، VCD و DVD	video
26	انواع یارانه و تبلت	pc
27	تلفن همراه	mobile
28	فریزر	freeizer
29	یخچال	yakhchal
30	یخچال فریزر	yakhchal.f
31	اجاق گاز	gaz
32	جارو برقی	jaro.b
33	ماشین لباسشویی	m.lebas
34	چرخ خیاطی	charkh.kh
35	پنکه	panke
36	کولر آبی متحرک	cooler.a
37	کولر گازی متحرک	cooler.g
38	ماشین ظرفشویی	m.zarf
39	مایکروویو و انواع فرهای هالوژن دار	microfer
40	آب لوله‌کشی	ab.l
41	برق	bargh
42	گاز لوله‌کشی	gaz.l
43	تلفن ثابت	tel
44	دسترسی به اینترنت	internet
45	حمام	hamam
46	آشپزخانه	ashpazkhane
47	کولر آبی ثابت	cooler.a.s
48	برودت مرکزی	broodat.m
49	حرارت مرکزی	hararat.m
50	پکیج	package
51	کولر گازی ثابت	cooler.g.s
52	شبکه عمومی فاضلاب	fazelab
ردیف	نوع سوخت عمده مصرفی خانوار	
	تعریف متغیر	نام متغیر
53	نوع سوخت برای پخت و پز	sookht.p
54	نوع سوخت برای ایجاد گرما	sookht.g
55	نوع سوخت برای تهیه آب گرم	sookht.ab

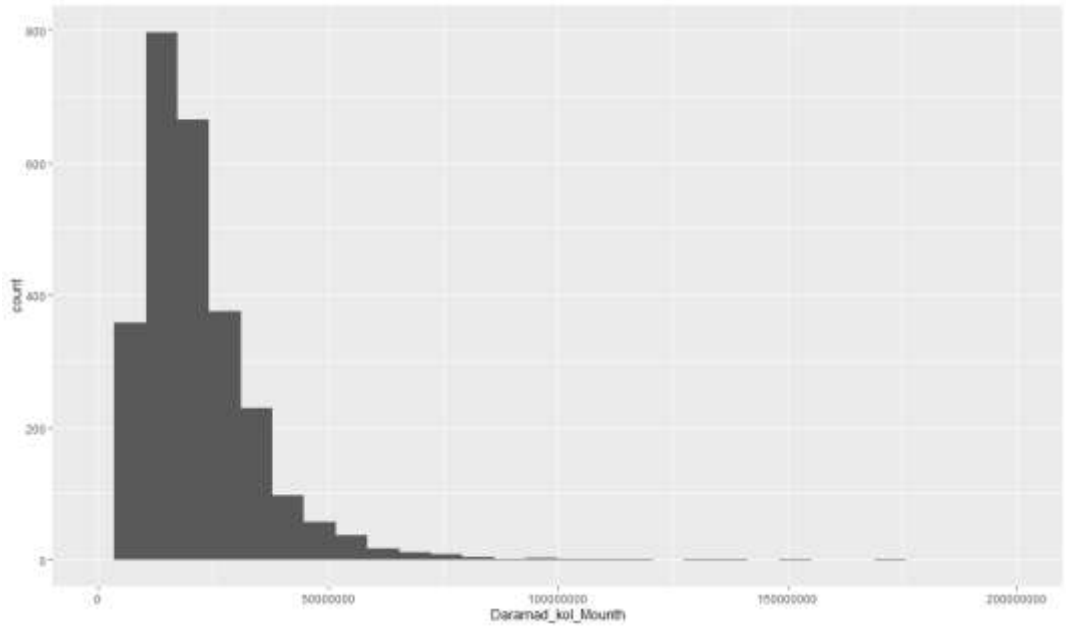
ردیف	هزینه‌های خانوار	
	تعریف متغیر	نام متغیر
56	هزینه‌های بهداشتی خانوار در یکماه گذشته	Hazine_Behdashti
57	هزینه ارتباطات خانوار در یکماه گذشته	Hazine_Ertebatat
58	هزینه‌های غذای آماده هتل و رستوران‌های خانوار در یکماه گذشته	Hazine_Ghazayeamade
59	هزینه‌های حمل و نقل خانوار در یکماه گذشته	Hazine_Hamlonaghl
60	هزینه کالاها یا خدمات متفرقه خانواردر یکماه گذشته	Hazine_kalavakhadamat
61	هزینه‌های خوراکی و دخانیات خانوار در یکماه گذشته	Hazine_Khorakivadokhani
62	هزینه‌های لوازم خانگی خاوار در یکماه گذشته	Hazine_lavazemkhanegi
63	هزینه‌های مسکن- آب، سوخت، روشنایی و...	Hazine_Maskan
64	ارزش اجاری رهن ، به ازای هر ۱ میلیون تومان ۳۰ هزار تومان اجاره که اعداد ستون از ضرب مبلغ رهن در 0.03 بدست آمده است.	Rahn
65	هزینه‌های نوشیدنی خانوار در یکماه گذشته	Hazine_Noshidani
66	هزینه های تفریحات خانوار در ماه گذشته	Hazine_Tafrihat
67	هزینه‌های پوشاک خانوار در یکماه گذشته	Hazine_Pushak
ردیف	درآمدهای خانوار	
	تعریف متغیر	نام متغیر
68	مبلغ دریافتی یارانه نقدی در ۱۲ ماه گذشته	Daramad_Yarane
69	درآمد آزاد خانوار در 12 ماه گذشته	Daramad_Azad
70	درآمدهای متفرقه خانوار در 12 ماه گذشته	Daramad_Motefaraghe
71	درآمد مزد خانوار در یک ماه گذشته	Daramad_Mozd_Month
72	درآمد مزد خانوار در یک سال گذشته	Daramad_Mozd_Year

فصل دوم:

کشف پاکسازی و پیش پردازش داده ها:

نکته1: همانطور که قرار بود همه ی درآمد ها به ماه را در 12 ضرب کرده و با درآمد های سالانه جمع کرده و جواب پایانی را در آخر به 12 تقسیم می کنم تا متغییر برآمد خود را به نام درآمد کل ماهانه(Daramad_Kol_Mounth) بدست آید.

که نمودار فراوانی آن به ترتیب زیر است:



نکته2: با توجه به نزدیک نبودن 12 برابر مزد ماهانه به مزد سالانه در بسیاری از مشاهدات با بررسی کردن این دو متغییر به این نتیجه رسیدیم برای متغییر مزد همان مزد سالانه را در فرمول بدست آوردن درآمد کل ماهانه قرار دهیم.

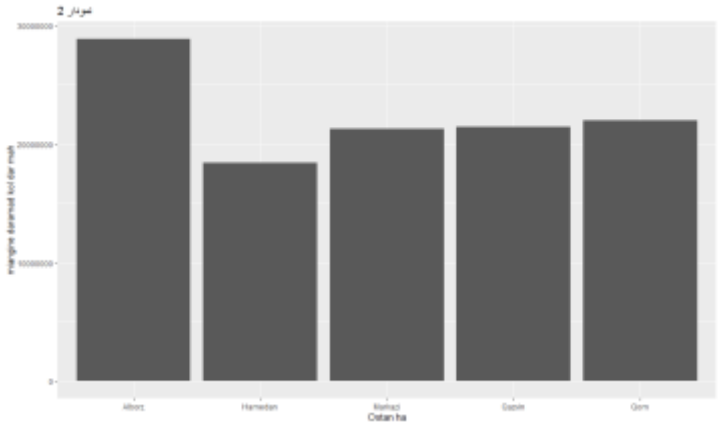
نکته3: در داده ها در هرستون مقدار بسیار زیادی داده گم شده(NA) موجود است که با توجه به هر ستون تصمیمات لازم اتخاذ خواهد شد، اما نکته این جاست که با توجه

به این که این داده ها از پرسش نامه ها بدست آمده اکثر مقادیر NA صفر هستند به طور مثال: در داده های ما در ستون درآمد آزاد هیچ مورد صفری گزارش نشده است اما بسیاری از مشاهدات NA هستند، چراکه افردی که درآمد آزاد ندارند این قسمت را خالی گذاشته اند.

نکته4: در مواجهه با متغییر های درآمد به جای صفر، برای NA ها، یک گذاشته میشود چراکه میدانیم احتمال نیاز به استفاده از لگاریتم درآمد ها کم نیست و از طرفی چون اندازه درآمد ها به ریال است این کار تاثیر چندانی بر میانگین و دیگر پارامتر ها و ویژگی های متغییر ها اعمال نمی کند.

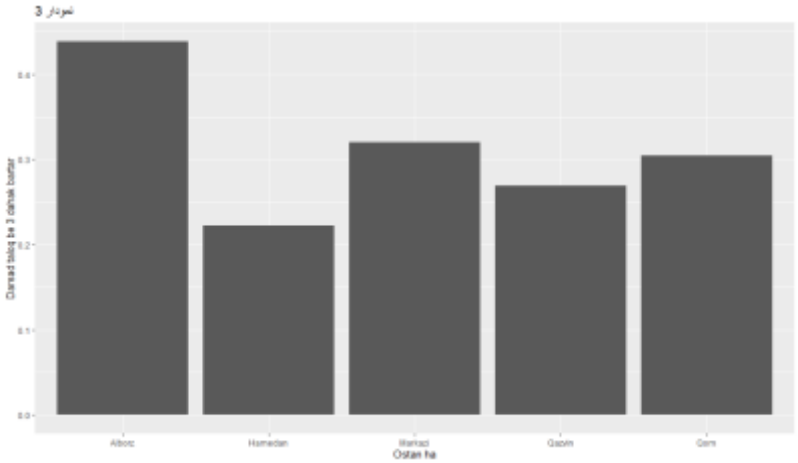
حال متغییر برآمد رسته ای را نیز می سازیم: اگر متغییر درآمد کل ماهانه عضو سه دهک اول بود 1 و اگر نبود صفر.

و اکنون به تصویر سازی داده ها و رسم نمودار ها می پردازم:



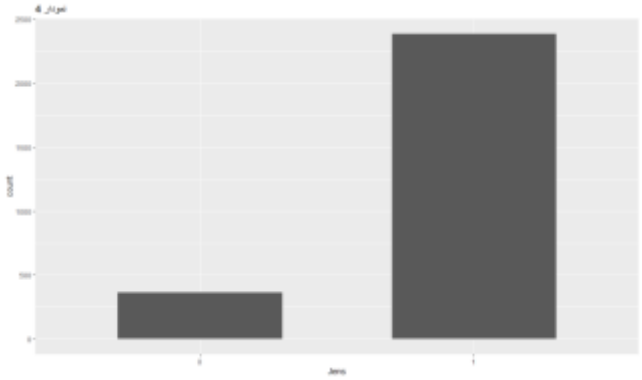
نمودار 2: نمودار میله ای که در آن محور افقی بیانگر استان ها و محور عمودی بیانگر میانگین درآمد هر یک از استان ها است.

از نمودار 2 می توان برداشت کرد که : استان های مرکزی قزوین و قم میانگین درآمد کل ماهانه یکسانی دارند، استان البرز به نسبت درآمد بیشتر ی دارد و همدان کمی کمتر.(می توان در ادامه مثلا استان ها را به البرز و غیره تقسیم کرد.)

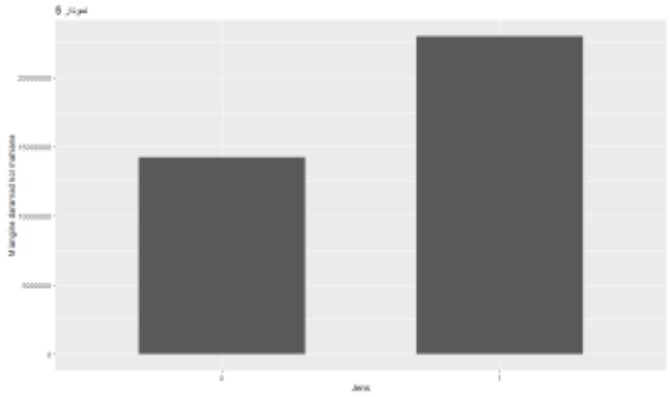


نمودار3 : نمودار میله ای که در آن محور افقی بیانگر استان ها و محور عمودی بیانگر درصد افرادی است که عضو سه دهک اول اقتصادی هستند.

نتیجه نمودار 3 مانند نمودار قبلی است فقط کمی بین قم، قزوین و مرکزی تفاوت ایجاد شد .

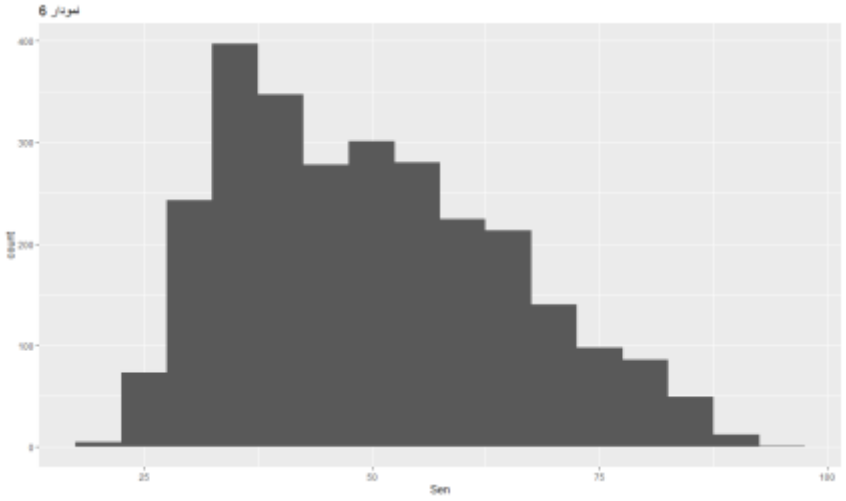


نمودار 4 : نمودار میله ای فراونی متغییر جنسیت سرپرست خانوار که همانطور که انتظار می رود تعداد خانوار هایی که سرپرست مذکر(1) دارند خیلی بیشتر از مونث(0) است.



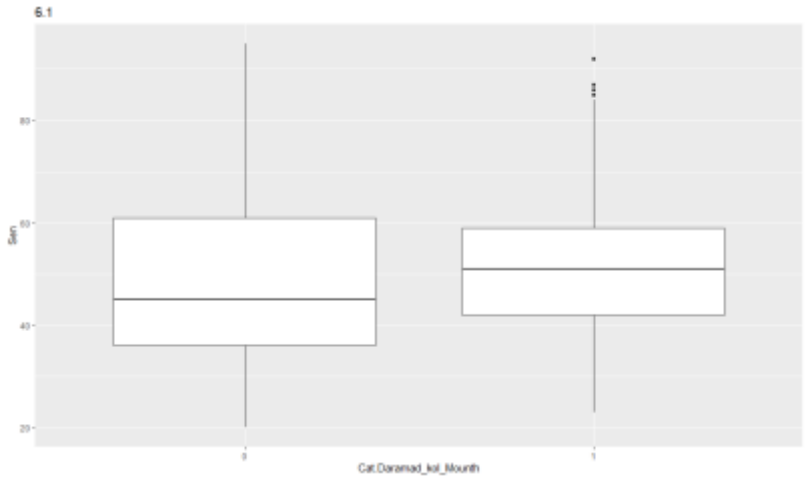
نمودار 5 : نمودار میله ای که در آن محور افقی بیانگر جنسیت و محور عمودی بیانگر میانگین درآمد است.

از نمودار 5 می توان برداشت کرد که : میانگین درآمد کل برحسب سرپرست خانوار که یک متغییر خوب است چرا که خانوار های با سرپرست مذکر میانگین درآمد بیشتری دارند.



نمودار 6: نمودار بافت نگار که نمایانگر فراوانی سنین مختلف برای سرپرست خانوار ها.

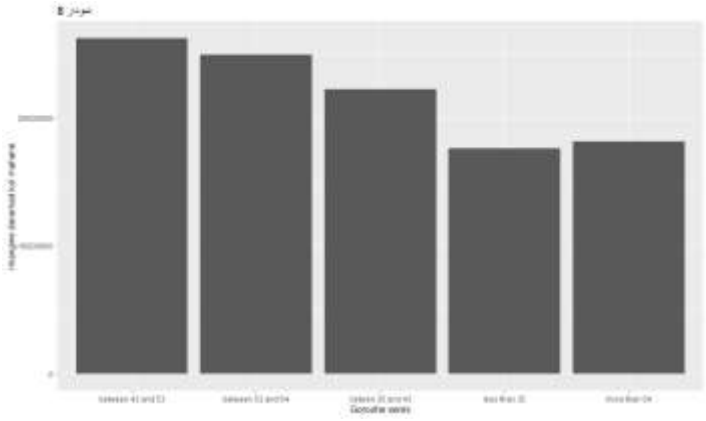
(این نمودار تقریبا نرمال است کمی چولگی به راست دارد.)



نمودار 6.1 : نمودار جعبه ای که محور افقی در آن متغییر بر آمد پروژه و محور عمودی متغییر سن است.

از نمودار 7 می توان برداشت کرد که : متغییر سن سرپرست خانوار خیلی موثر نیست.

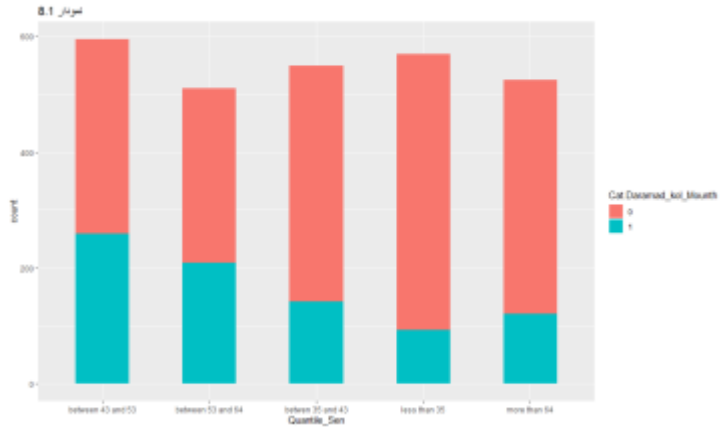
پس آن را به صورت رسته ای تبدیل می کنیم:



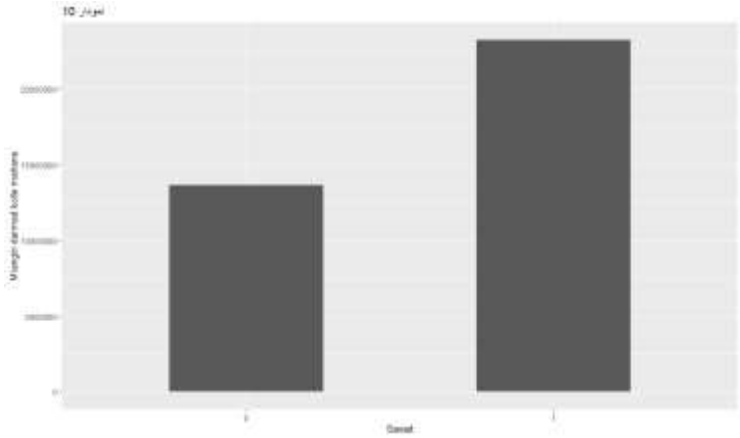
نمودار8: نموداری میله ای که محور افقی بیانگر متغیر رسته ای سن است و محور افقی بیانگر میانگین درآمد.

در مورد این نمودار می توان گفت: از نمودار قبل قابل درک تر است و بین بازه های سنی مختلف یک گپ کوچک می اندازد.

نکته: برای کاهش بعد جلو تر به ازای این بازه ها برای متغییر سن از میانهی هر بازه استفاده خواهم کرد.



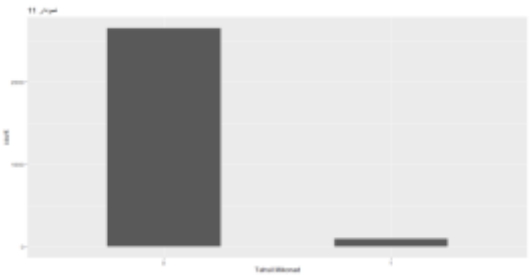
نمودار 8.1 : نمودار میله ای که فراوانی متغیر تازه رسته ای شده سن را نشان می دهد که بوسیله ی رنگ آمیزی متغییر برآمددر آن مشخص شده است.



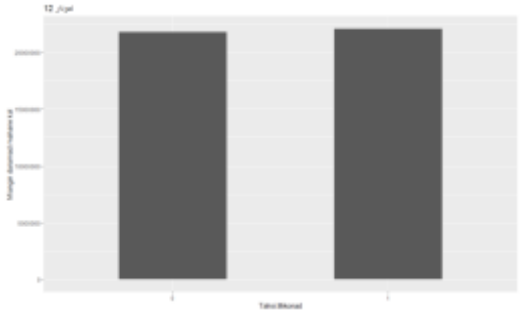
نمودار 10 : نمودار میله ای که محور افقی بیانگر داشتن (1)و یا نداشتن سواد (0) است، محور عمودی نیز بیانگر میاگین درآمد.

از نمودار 10 می توان برداشت کرد که داشتن یا نداشتن سواد متغییری خوب و موثر است چرا که میانگین درآمد افراد با سواد و بی سواد بسیار متفاوت است..

نکته 5 : در متغییر Tahsil.Mikonad از آنجایی که تمام مقادیر گم شده سرپرست خانوار هایی بودند که سواد نداشتند به این نتیجه رسیدم که همه ی آن ها چون تحصیل نمی کردند این مقدار را خالی گذاشته اند و به همه ی آن ها مقدار صفر یعنی تحصیل نمیکند را نسبت دادم.



نمودار 11: این نمودار فراوانی سرپرست های در حال تحصیل است که با توجه به این که تعداد افرادی که در حال تحصیل هستند خیلیی کم است احتمالا متغییر کار آمدی نمی باشد اما باز با استفاده از تجمیع aggregation نموداری رسم میکنم تا تاثیر را مطالعه کنم.



نمودار 12 : در این نمودار محور افقی متغیر رسته ای تحصیل میکند است که (0) به معنی خیر و (1) به معنی بله است، محور عمودی نیز نشانگر میانگین درآمد است.

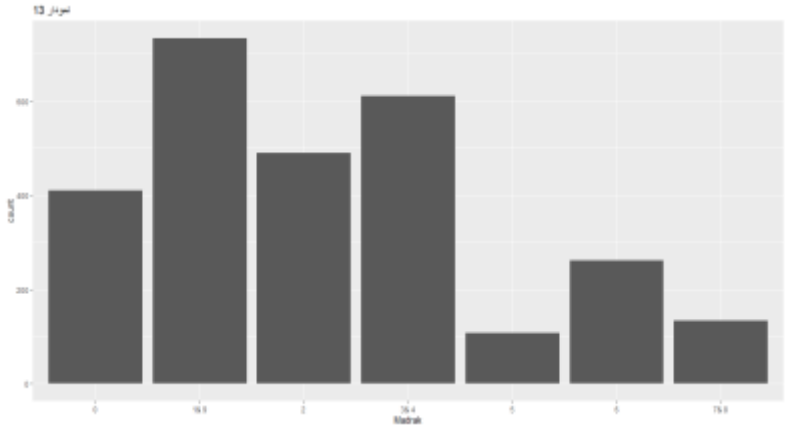
با دیدن این نمودارمعلوم میشود که این متغییر کار آمد نیست چراکه علاوه بر کم بودن فراوانی سرپرست های محصل میانگین آن ها نیز تفاوت چندانی با افراد عادی ندارد.

نکته6: در این متغییر نیز تمام مقادیر NA در واقع همان افرادی هستند که سواد ندارند.

نکته7: طبق پرسش نامه می دانیم که :

سوادآموزی /ابتدایی	1
1 متوسطه / راهنمایی	2
2 متوسطه /متوسطه	3
پیش دانشگاهی و دیپلم	4
کاردانی/دیپلم فوق	5
کارشناسی/لیسانس	6
کارشناسی ارشد و دکترای حرفه ای	7
دکترای تخصصی	8
سایر و غیر رسمی	9

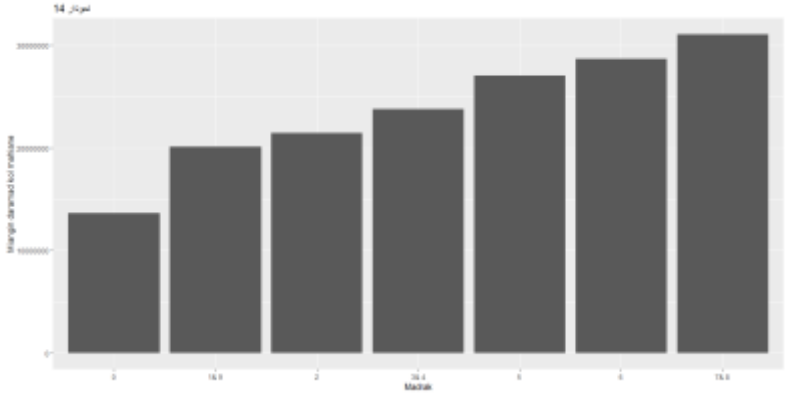
با بر رسی ها ی لازم و توجه به این که فراوانی مورد 9 کم است ومیانگین کل آن شبیه 1 است و از نظر ارزش مدرک حدودا در همان رده است آن را با 1 ادغام کردم.و به طریق مشابه، 3 با 4 و 7 با 8 ادغام شد.



نمودار 13 : نمودار فراونی بر حسب مدارک مختلف پس از ترکیب کردن چند رسته.

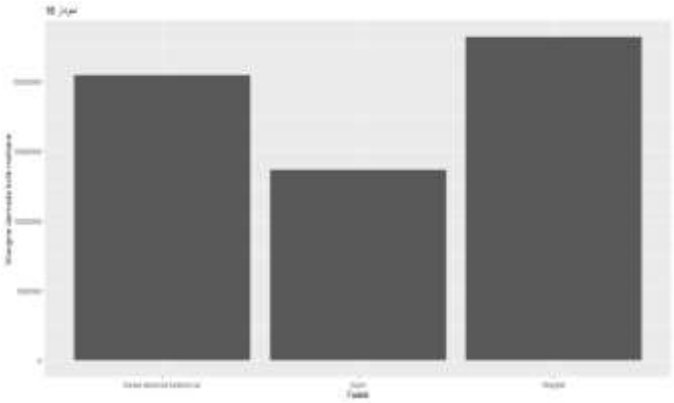
نکته8 : اگر این متغییر نگه داشته شود متغییر سواد باید حذف شود. چون متغییر سواد از این متغییر به راحتی بدست می آید.

نکته9: فراوانی متغییر های 3و8 یعنی: دبیرستان و دکترای تخصصی بسیار کم است.



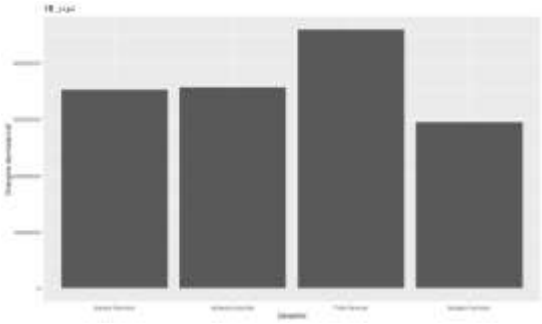
نمودار14 : نمودار میله ای که در آن محور افقی بیانگر مدرک و محور عمودی بیانگر میانگین درآمد است.

از نمودار 14 میتوان دریافت که: به سادگی می توان رابطه ی مستقیمی بین مدرک و میانگین درآمد دریافت، پس قطعا این متغییر کار آمد خواهد بود.



نمودار 16: نمودار میله ای که در آن محور افقی بیانگر فعالیت و محور عمودی بیانگر میانگین درآمد است.

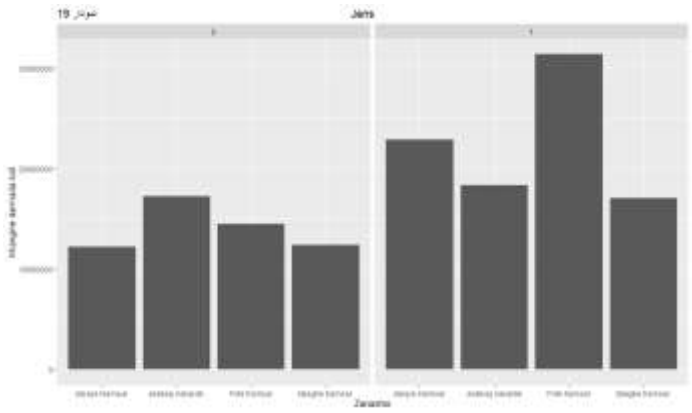
از نمودار 16 میتوان دریافت کرد که برخلاف انتظار تفاوت میانگین در آمد در افرادی که دارای کاری نیستند اما درآمد دارند با افرادی که در حال حاضر دارای کار هستند تفاوت چندانی ندارد .



نمودار 18: نمودار میله ای که در آن محور افقی بیانگر وضعیت زناشویی و محور عمودی بیانگر میانگین درآمد است.

از نمودار18 میتوان نتیجه گرفت که: افراد دارای همسر و ازدواج نکرده مانند هم هستند. فوت همسر میانگین درامد کل ماهانه بالاتر و طلاق پایین تر خواهند داشت.

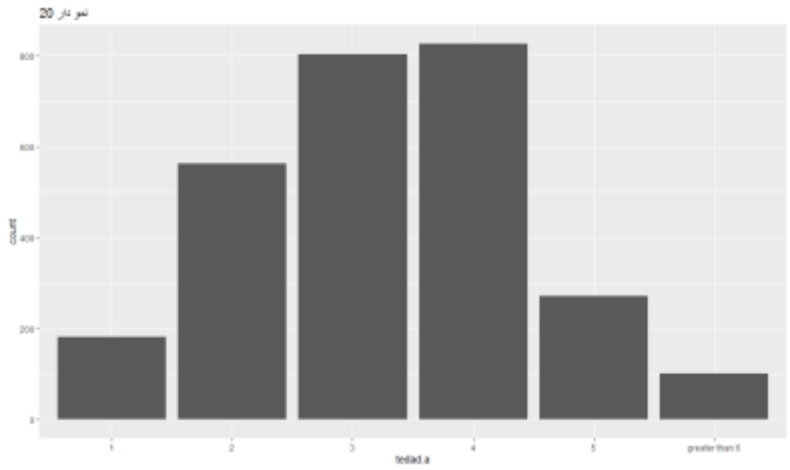
حال نموداری دقیق تر با پنل های چند گانه رسم می کنم.



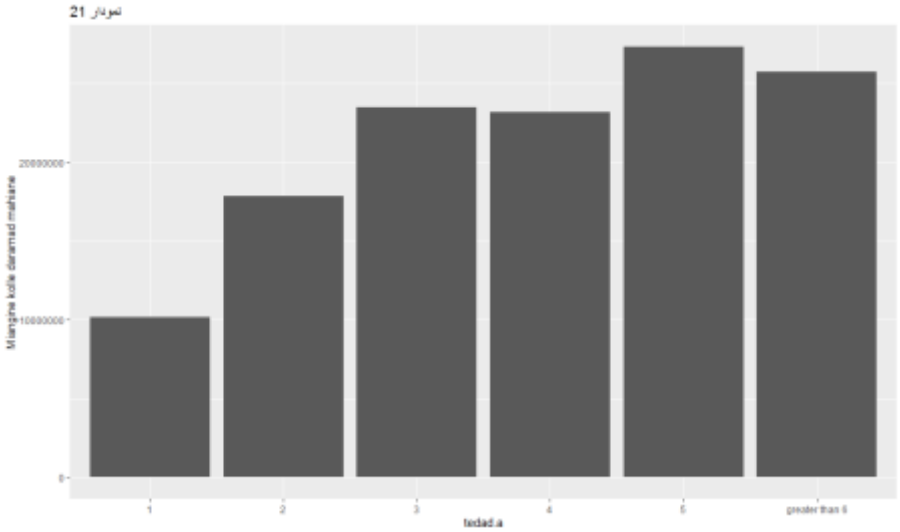
نمودار19: این نمودار از دو پنل جدا تشکیل شده که بر اساس جنسیت این دو از هم جدا شده اند. و محور افقی در هر پنل نماینده ی وضعیت زناشویی و محور عمودی میانگین در امد است.

این نمودار بسیار کار آمد است چراکه امکان بررسی تاثیرات همزمان دو متغییر وضعیت زناشویی و جنسیت را به ما می دهد. نکات جالبی که می توان برداشت کرد مثلا: در زنان افرادی که ازدواج نکرده اند نسبت به دیگران میانگین در آمد کلی بیشتری دارند ولی در مردان برعکس. در مورد فوت همسر نیز بین مرد و زن بسیار تفاوت موجود است.

نکته10: بدلیل کم بودن فراوانی خانوار های با تعداد اعضای 7 وبیشتر همه را در گروهی به نام خانوار های 6 و بزرگ تر گذاشتیم.

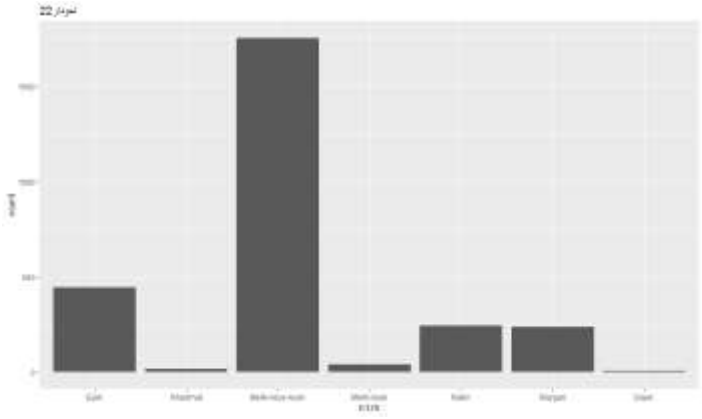


نمودار20: نمودار فراونی تعداد اعضای خانوارها پس از یکی کردن خانوار های بزرگ تر از 6 به یک متغیر بزرگ تر از 6.

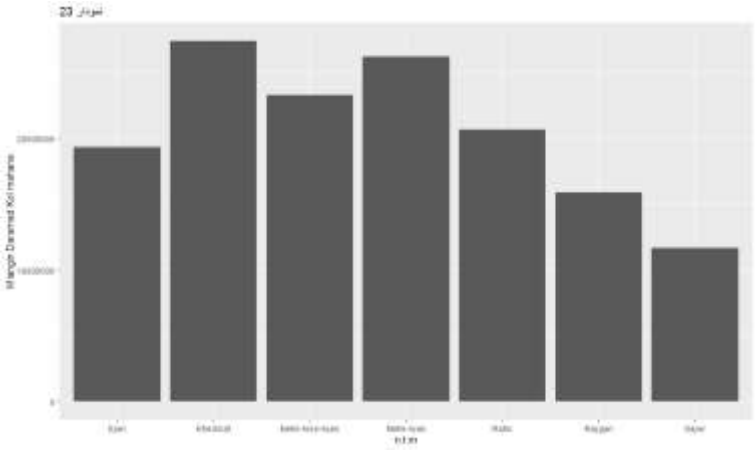


نمودار21: نمودار میله ای که در آن محور افقی بیانگر تعداد اعضا و محور عمودی بیانگر میانگین درآمد است.

با دقت در نمودار 21 میتوان ادعا کرد که: با بیشتر شدن اعضای خانوار درآمد افزایش پیدا می کند. البته به نظر می توان خانوار های 3و 4 عضوی را در یک گروه و همچنین 5 و بزرگتر از 6 در یک گروه.

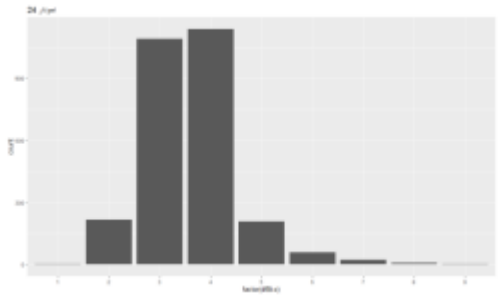


نمودار 22: فراوانی نحوه تصرف منزل که همانظور که می بینیم سایر، ملکی-اعیان خیلی کم یاب هستند پس بعد از استفاده از تجمیع و رسم نمودار سعی میشود آن ها را با توجه به ویژگی هایشان در گروه های دیگری گنجاند.



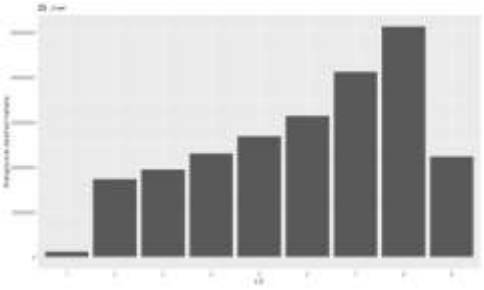
نمودار 23: نمودار میله ای که در آن محور افقی بیانگر نحوه تصرف منزل و محور عمودی بیانگر میانگین درآمد است.

با توجه به نمودار 23: رایگان وسایر را با یکدیگر و ملکی-اعیان و خدمت با هم در نظر گرفته میشود.



نمودار24: نمودار فراوانی تعداد اتاق.

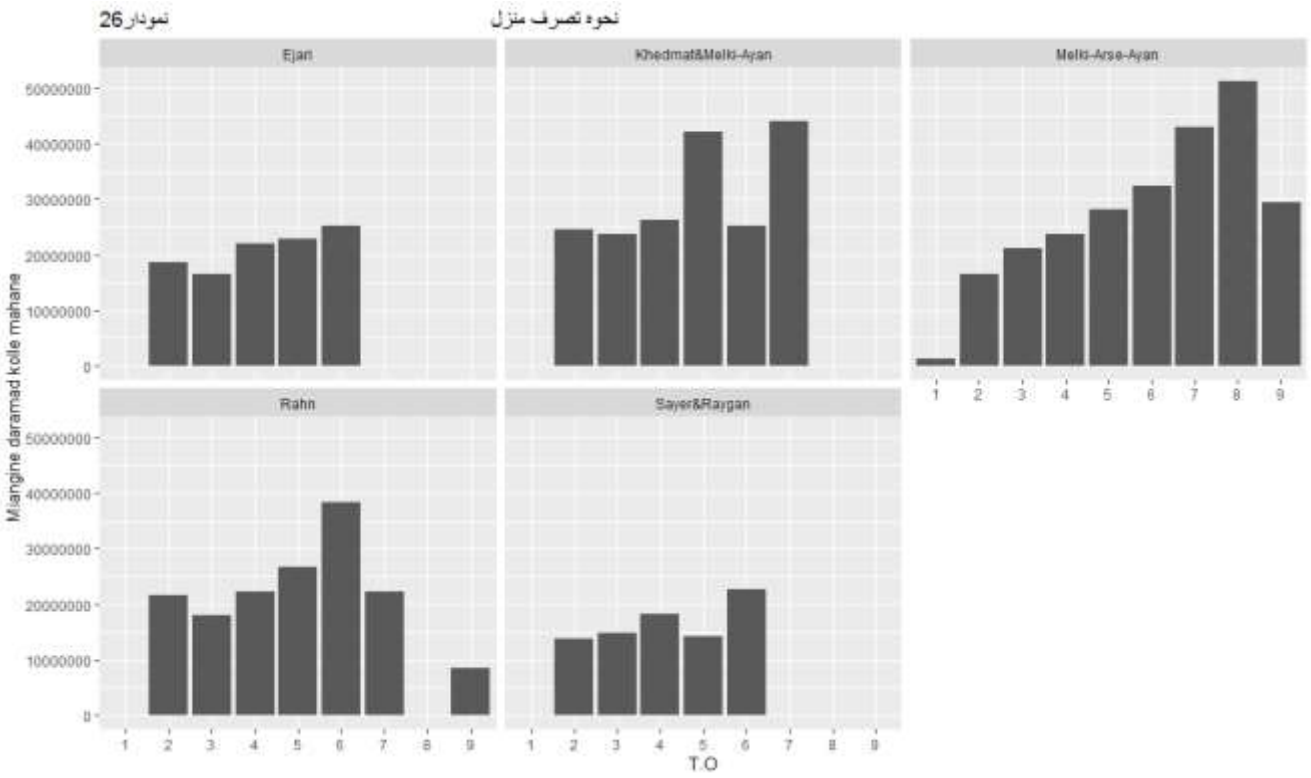
فراوانی تعداد اتاق 1و 8و 9 بسیار کم است.



نمودار25: نمودار میله ای که در آن محور افقی بیانگر تعداد اتاق و محور عمودی بیانگر میانگین درآمد است.

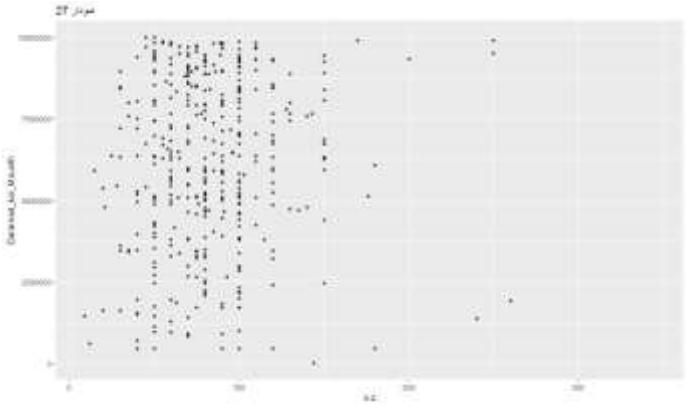
با دقت در نمودار 25 مشخص میشود که: این متغییر مفید خواهد بود چراکه جز منزل 9 اتاقه در باقی با افزایش تعداد اتاق میانگین درآمد افزایش داشته است.

حال با کمک پنل های چند گانه این متغییر با نحوه تصرف منزل رسم می شود:



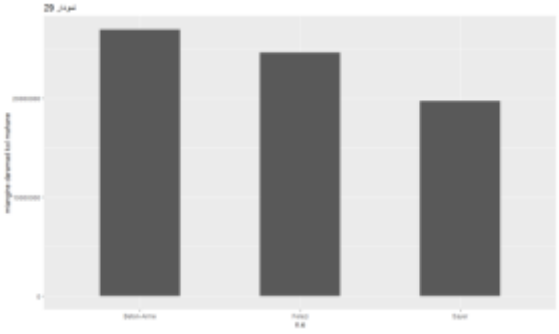
نمودار26: این نمودار از پنج پنل جدا تشکیل شده که بر اساس نحوه تصرف منزل شده اند. و محور افقی در هر پنل نماینده ی تعداد اتاق و محور عمودی میانگین در آمد است.

به نظرمن نکته ی اصلی ای که می توان از این نمودار برداشت کرد این است که متغییر تعداد اتاق در مواردی که نحوه تصرف ملکشان: ملکی-عرصه- اعیان است متغییر کار آمد تری است.



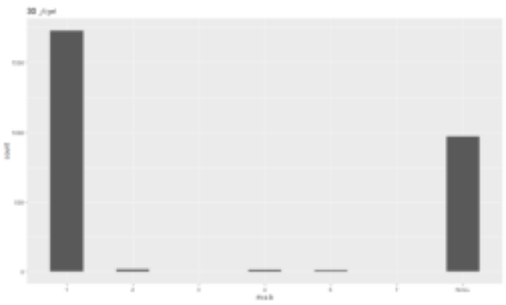
نمودار27: نمودار نقطه ای بر حسب سطح زیر بنا و در آمد کل ماهانه:

نشان می دهد در هر سطح زیر بنا، در آمد پایین و بالا داریم و نمی توان رابطه ی محسوسی بین آن ها یافت.

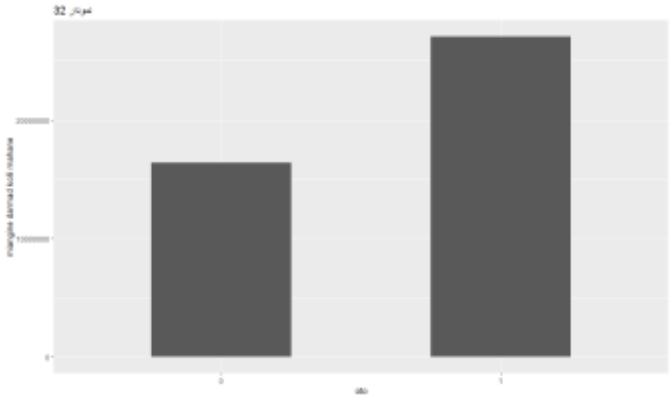


نمودار29: نمودار میله ای که در آن محور افقی بیانگر نوع اسکلت و محور عمودی بیانگر میانگین درآمد است.

به نظر، ساختمان های با اسکت بنا ی متفرقه درامد های کمتری دارندو شاید بتوان بتن آرمه و فلزی را یکی کرد.

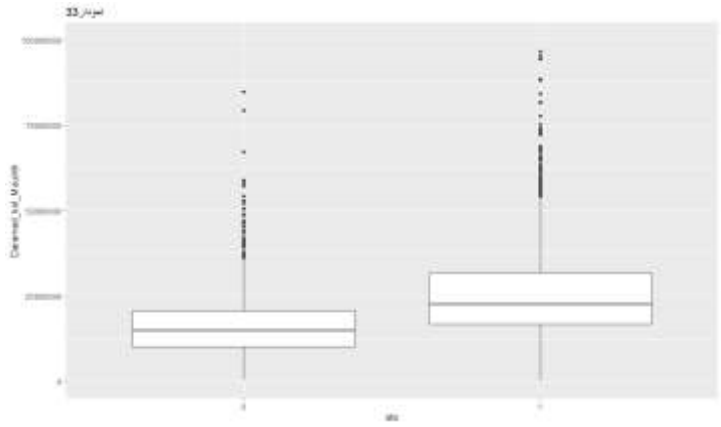


نمودار 30: فراوانی مصالح عمده ی بنا است، که اکثر داده ها یا گمشده هستند یا 1 که همان آجر یا سنگ و آهن است نتیجتا این متغییر به نظر اصلا کار آمد نمی اید.



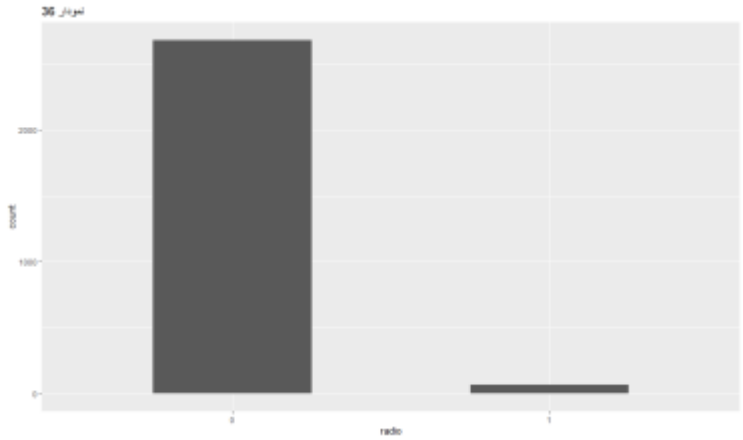
نمودار 32 : نمودار میله ای که در آن محور افقی بیانگر داشتن یا نداشتن اتو و محور عمودی بیانگر میانگین درآمد است.

بر خلاف انتظار به نظر داشتن اتو در یک خانوار یک ملاک برای حدس درامد خواهد بود.



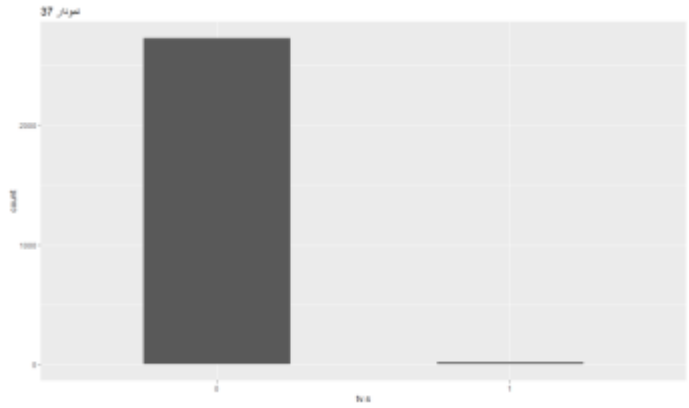
نمودار 33: نمودار جعبه ای با محور افقی متغییر دو دویی داشتن یا نداشتن اتو و محور افقی میانگین درآمد.

نمودار جعبه ای برای اتو نیز نشان می دهد تا حدی این متغییر تاثیر گذار به نظر می رسد.



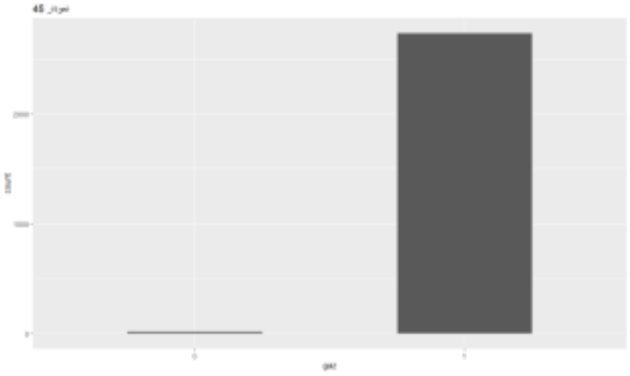
نمودار 36: امروزه در اکثر خانه ها رادیو نداریم.

پس این که فراوانی داشتن رادیو انقدر کمیاب است آن را به یک کاندید حذف شدن تبدیل می کند.



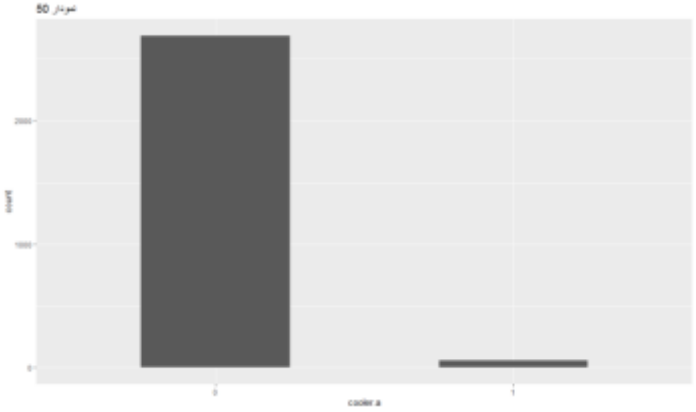
نمودار 37: نمودار فراوانی تلوزیون سیاه و سفید.

امروزه در اکثر خانه ها تلویزیون سیاه و سفید نداریم پس این که فراوانی داشتن انقدر کمیاب است آن را به یک کاندید حذف شدن تبدیل می کند.

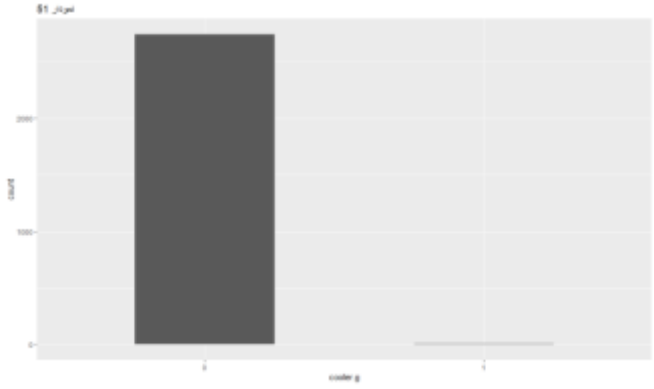


نمودار 45: نمودار فراوانی داشتن یا نداشتن متغییر گاز.

قطعا با توجه به کم یاب بودن نداشتن متغییر گاز، باید این متغییر حذف شود.



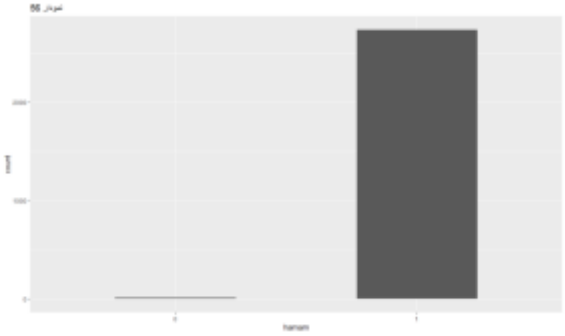
نمودار 50:فراوانی داشتن یا نداشتن کولر آبی متحرک که با توجه به بسیار پایین بودن فراوانی گویا این متغییر کاربردی نخواهد بود.



نمودار 51: فراوانی داشتن یا نداشتن کولر گازی متحرک. با توجه به بسیار کمیاب بودن این متغییر قطعا حذف خواهدشد.

نکته11: متغییر آب لوله کشی و برق حذف می شود چون همه ی خانوارها ی ما دارای این متغییر هستند.

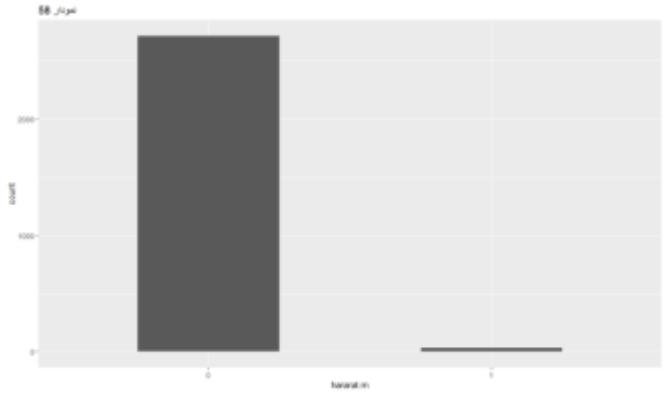
نکته 12: فراوانی نداشتن متغییر لوله کشی گاز انگشت شمار است. پس باید حذف شود.



نمودار 56: فراوانی نداشتن متغییر حمام انگشت شمار است. پس باید حذف شود.

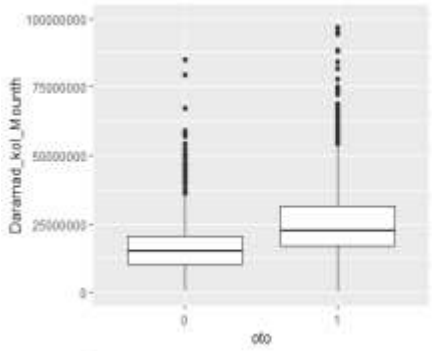
نکته13: آشپزخانه نیز چون حمام است.

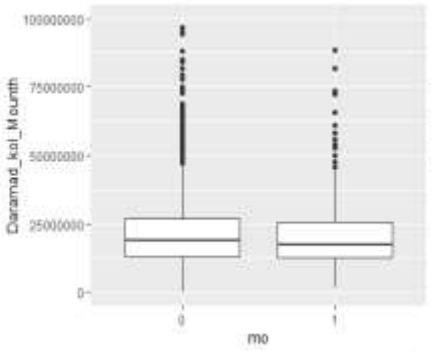
نکته14: برودت مرکزی نیز جز تعدادی انگشت شمار، در هیچ خانواری نبود پس باید حذف شود.

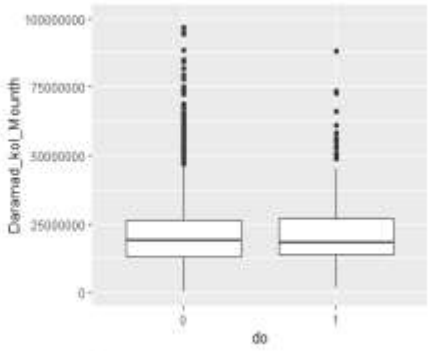


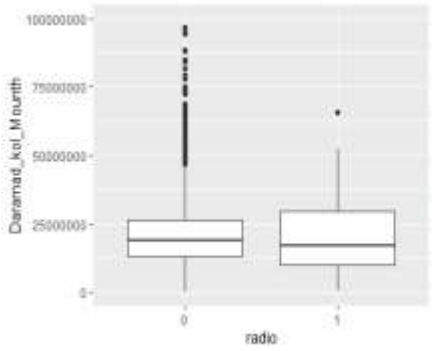
نمودار 58: باز هم فراوانی بسیار کم استفاده از حرارت مرکزی پس کاندید حذف خواهد بود.

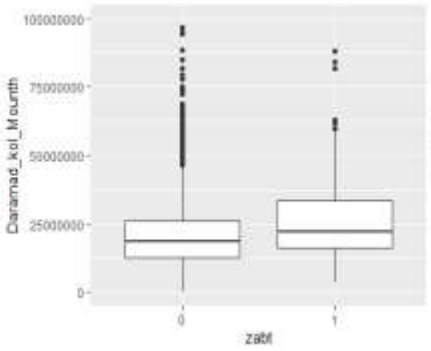
حال برای این متغییر هایی که بیانگر داشتن یا نداشتن امکاناتی یا وسایلی در منزل است از نمودار های جعبه ای پهلو به پهلو استفاده می کنیم که در هر نمودار محور افقی متغییر دودویی مورد نظر و محور عمودی میانگین درآمد است و پس از نمودار ها توضیحاتی از آن ها می دهیم.

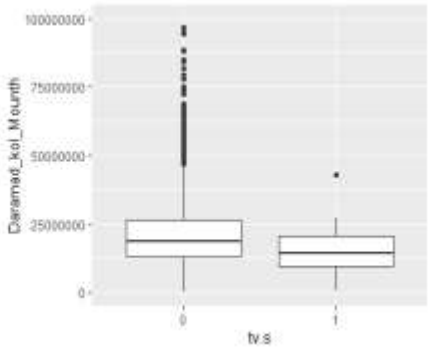

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
oto

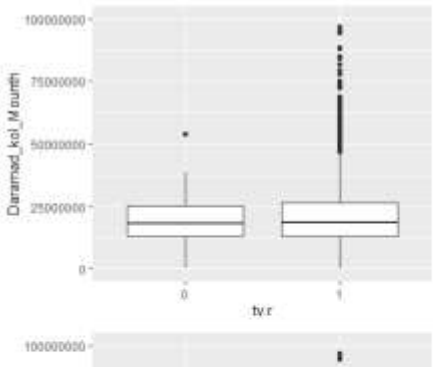

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
mo

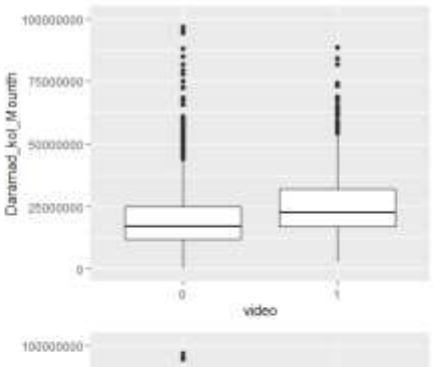

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
do

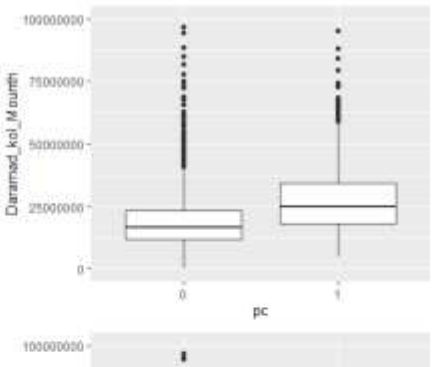

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
radio

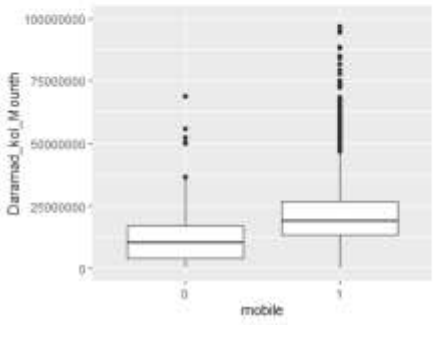

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
zabt

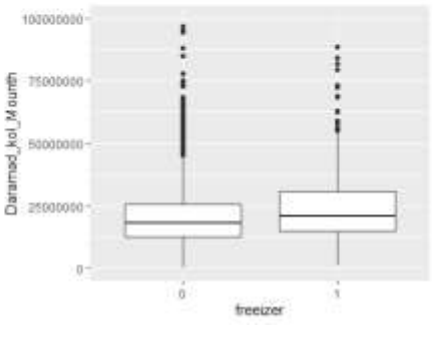

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
tv.s

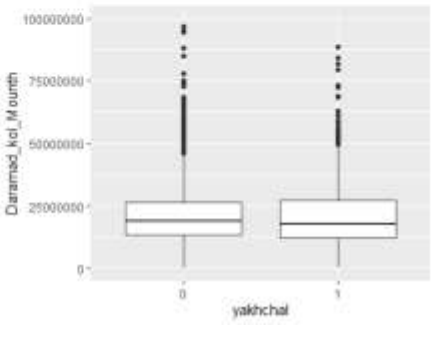

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
tv.r

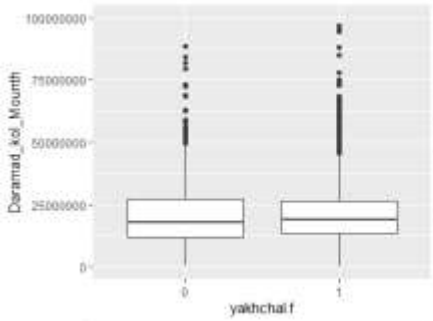

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
video

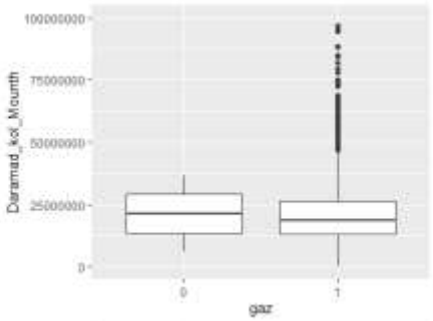

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
pc

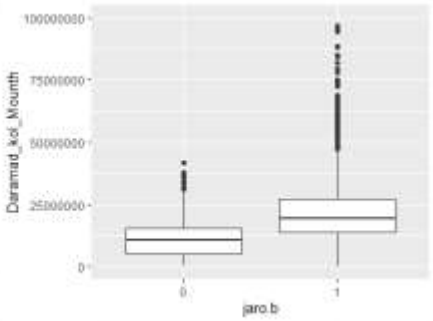

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
mobile

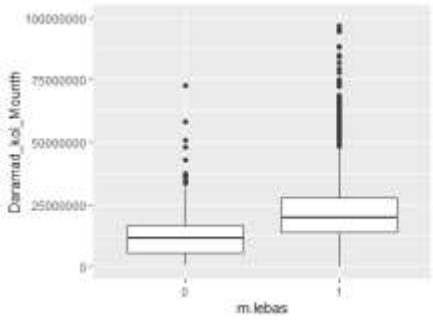

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
freeizer

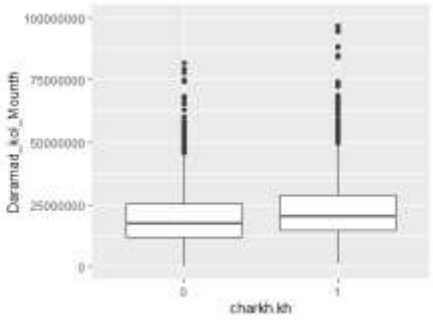

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
yakhchal

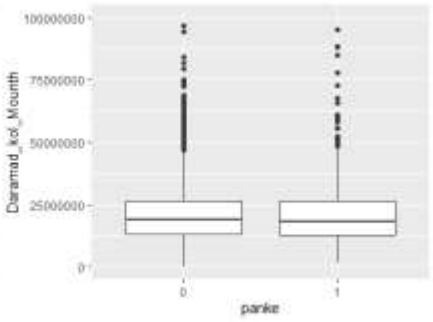

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
yakhchal.f

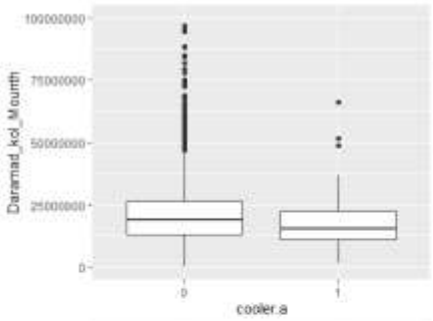

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
gaz

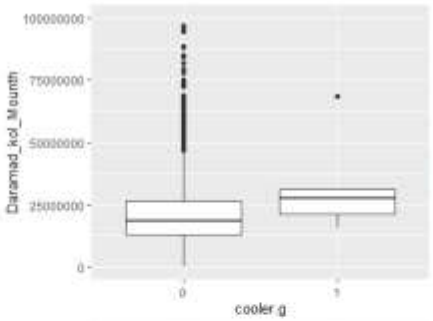

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
jaro.b

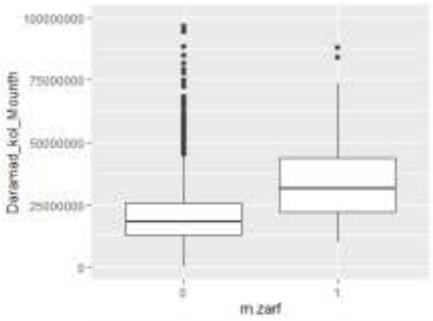

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
m.lebas

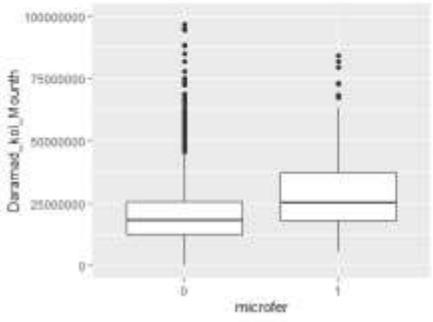

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
charkh.kh

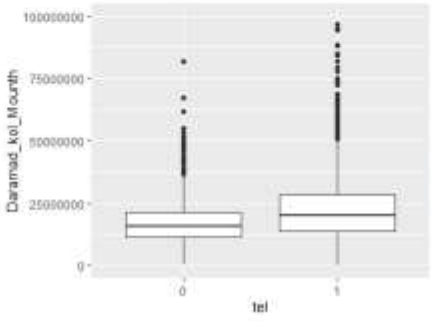

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
panke

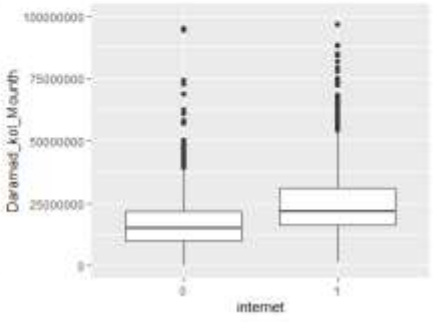

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
cooler.a

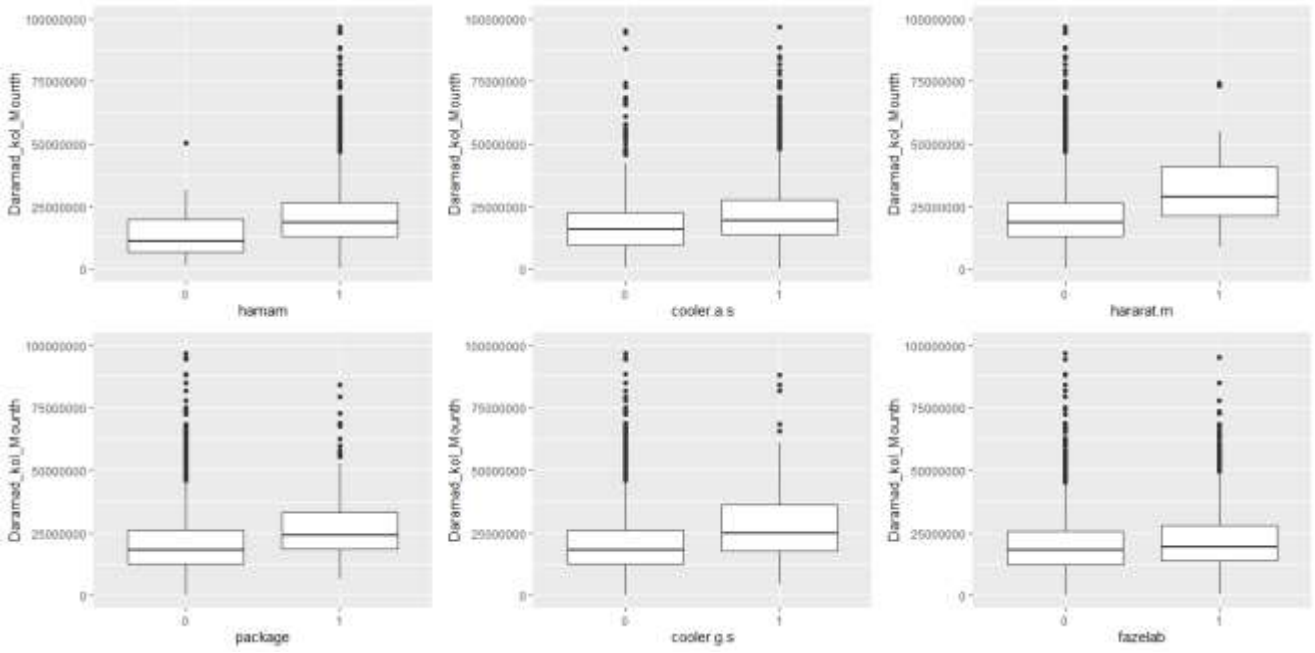

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
cooler.g


Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
m.zarf


Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
microfer

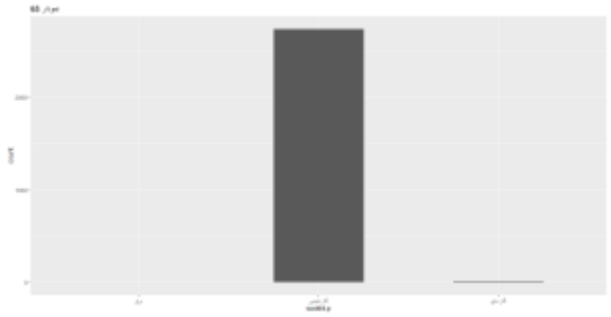

Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
tel


Daramad_kol_Mounth
100000000
75000000
50000000
25000000
0
0
1
internet

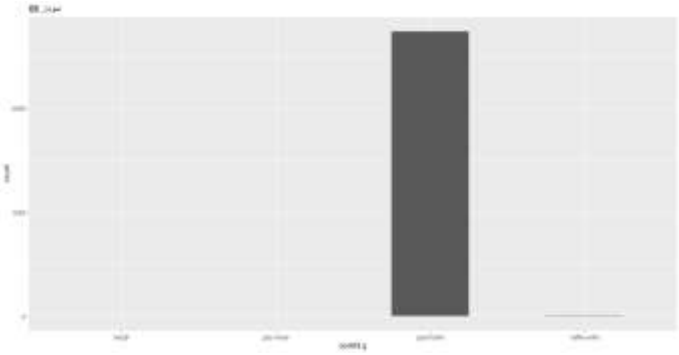


نمودار های جعبه ای بالا: در نمودار های بالا 30 متغییر مربوط به داشتن یا نداشتن وسایل یا امکاناتی را با نمودار جعبه ای پهلو به پهلو بررسی کرده ایم. در این نمودار ها متغییر هایی که 1و 0 آن ها (داشتن یا نداشتن) اشتراک کمتری داشته باشد متغییر های کارآمد تری هستند. هرچند اگر 0و 1 هم از هم خوب جدا شده باشد و فراوانی یکی از آن ها خیلی کم باشد آن متغییر کارآمدی نخواهد بود.

نکته15: با توجه به توضیحی که زیر نمودار قبل داده شد متغییر هایی چون حمام، کولر گازی متحرک و حرارت مرکزی با وجود کار آمد بودنشان به علت نادر بودن یکی از فراوانی هایشان بهتر است حذف شوند. ولی متغییر های : اتو ، ویدیو، کامپیوتر، موبایل، جاروبرقی، ماشین لباس شویی، ماشین ظرف شویی، ماکروویو، اینترنت و پکیج علاوه بر کارآمد بودن فراوانی های قابل قبول تری دارند.



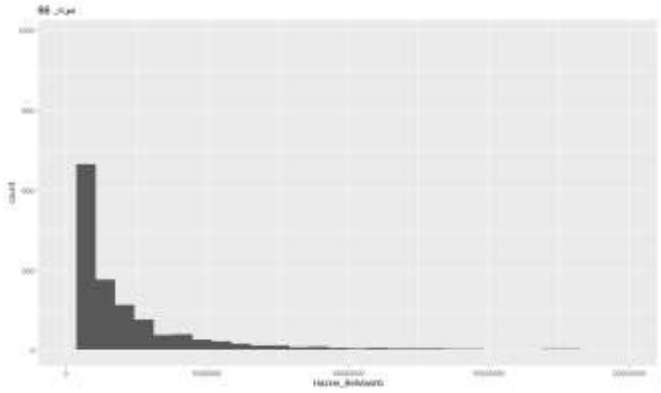
نمودار65: فراوانی سوخت مورد استفاده در پخت و پز که جز تعدادی انگشت شمار همه از گاز طبیعی استفاده می کنند پس این متغییر نیز باید حذف شود.



نمودار66: فراوانی سوخت مورد استفاده در ایجاد گرما که جز تعدادی انگشت شمار همه از گاز طبیعی استفاده می کنند پس این متغییر نیز باید حذف شود.

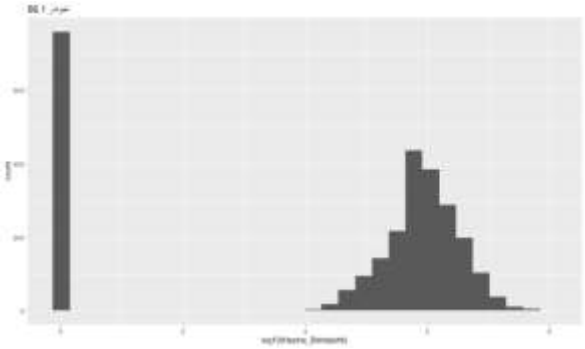
نکته16: متغییر sookht.ab نیز مانند دو متغییر قبلی است.

حال به سراغ هزینه ها میرویم همانند استدلالی که برای درامد ها داشتیم جای مقادیر NA ، می گذاریم.



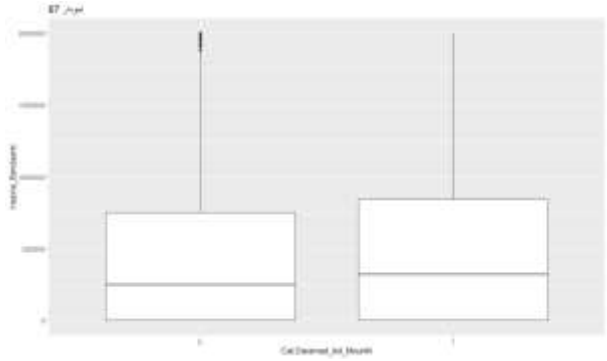
نمودار66: نمودار بافت نگار که در آن محور افقی هزینه بهداشت و محور عمودی آن فراوانی است.

نمودار 66 نشان می دهدکه: هرچه هزینه افزایش یابد فراوانی کاهش می یابد.



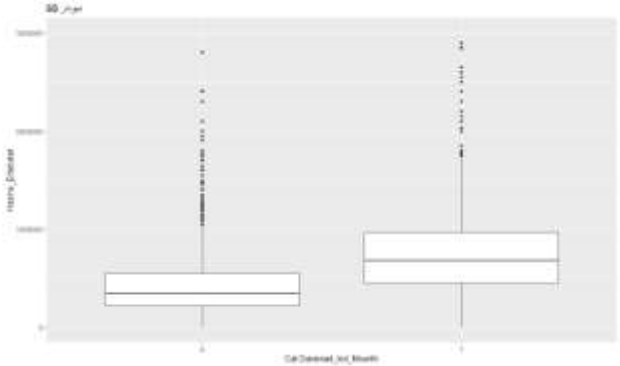
نمودار 66.1: با لگاریتم گرفتن شبیه نرمال می شود و چولگی کاهش می یابد.

نکته : برای تمام متغییر های هزینه با رسم بافت نگار نتیجه مشابهی میبینیم و با لگاریتم گرفتن به توزیع نرمال نزدیک می شود.



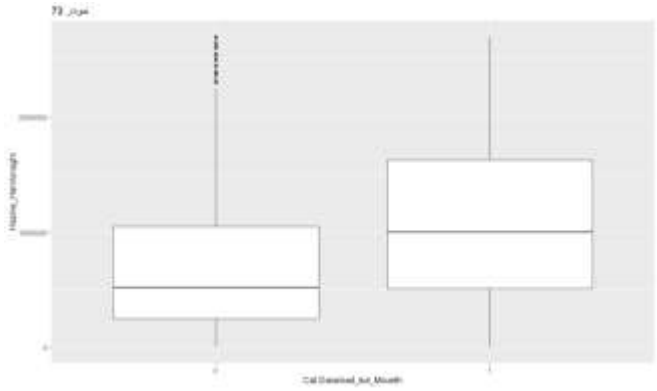
نمودار67: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه بهداشتی.

با توجه به نمودار 67: به نظر متغییر هزینه بهداشتی تاثیر اندکی دارد.



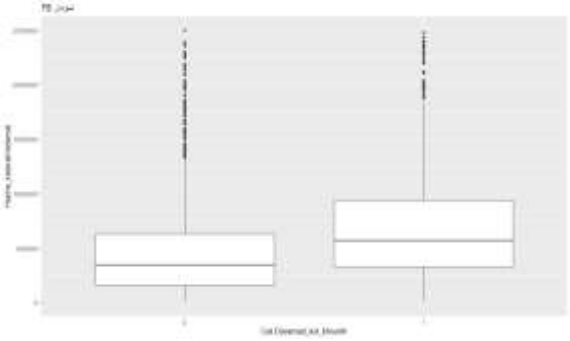
نمودار69 : نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه ارتباطات.

نمودار 69 بیانگر : تاثیر گذاری هزینه ارتباطات بر بودن یا نبودن در سه دهک اول اقتصادی است و می توان اضافه کرد که این متغییر، از هزینه بهداشت موثرتر است.



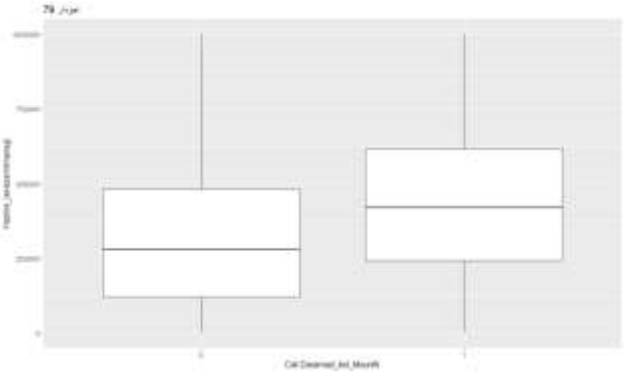
نمودار73: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه حمل و نقل است.

نمودار 73 نمایانگر تاثیر گذاری هزینه حمل و نقل بر بودن یا نبودن در سه دهک اول اقتصادی است.



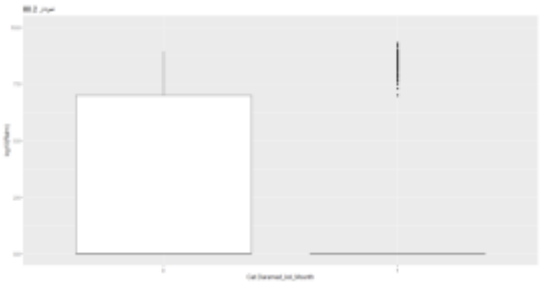
نمودار 75: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه کالا و خدمت.

نمودار 75 نشان دهنده ی تاثیر گذاری هزینه کالا و خدمت بر بودن یا نبودن در سه دهک اول اقتصادی است.



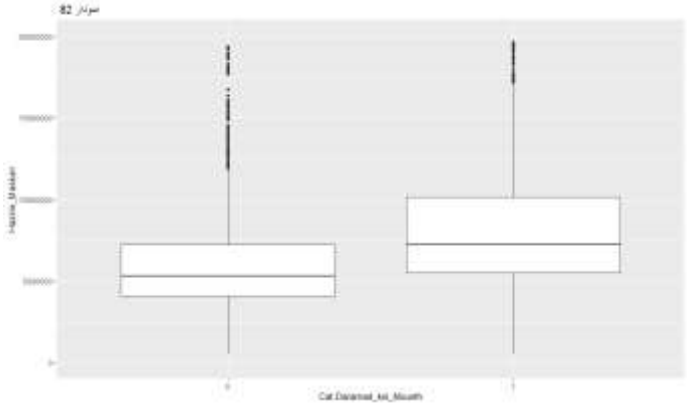
نمودار 79: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه لوازم خانگی است.

نمودار 79 نشان دهنده ی تاثیر گذاری هزینه لوازم خانگی بر بودن یا نبودن در سه دهک اول اقتصادی است.



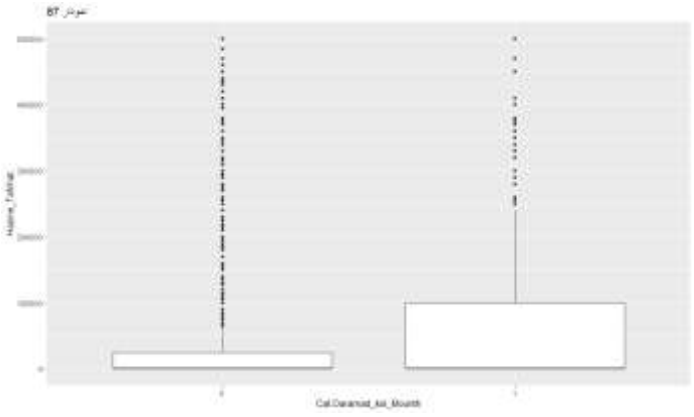
نمودار 80.2: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی لگاریتم رهن است.

از نمودار 80.2 نتیجه میگیریم افرادی که در سه دهک اول هستند اکثرا پولی برای رهن هزینه نمی کنند.



نمودار 82: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه مسکن.

نمودار 82 نشان دهنده ی تاثیر گذاری هزینه مسکن بر بودن یا نبودن در سه دهک اول اقتصادی است.

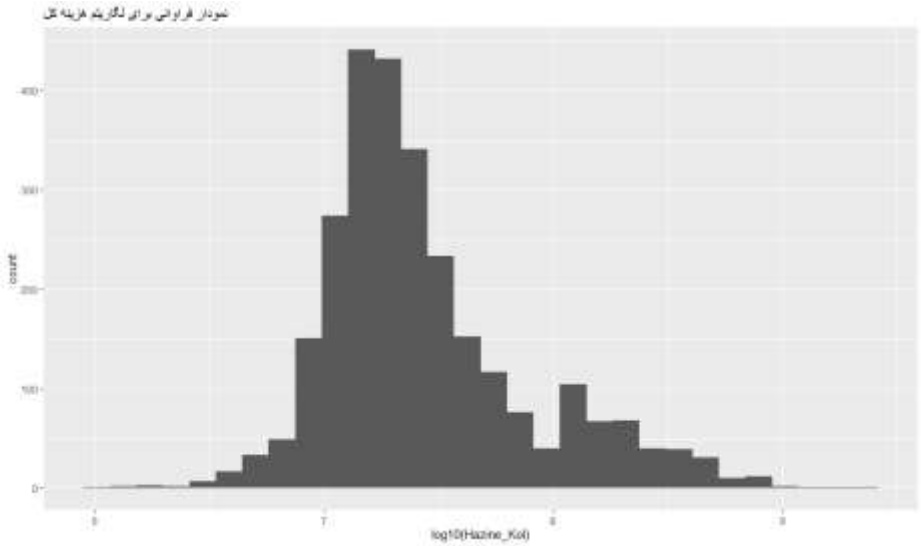


نمودار 87: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه تفریحات است.

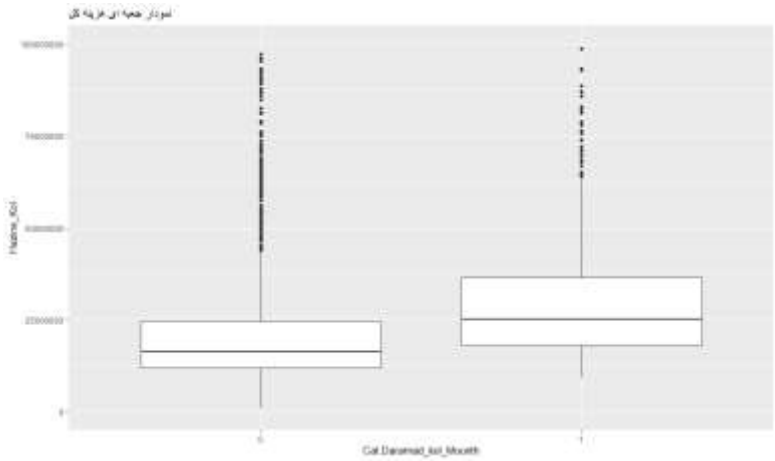
نمودار 87 نشانگر تاثیر گذاری هزینه تفریحات بر بودن یا نبودن در سه دهک اول اقتصادی است.

نکته: به نظر در بین متغییر های هزینه، خوراکی و دخانی ، غذای آماده، پوشاک کم ترین تاثیرات را دارند.

و در آخر یک ستون جدید بنام هزینه کل می سازیم و جمع تمام هزینه ها را می کذاریم نمودار هایش را می کشیم:

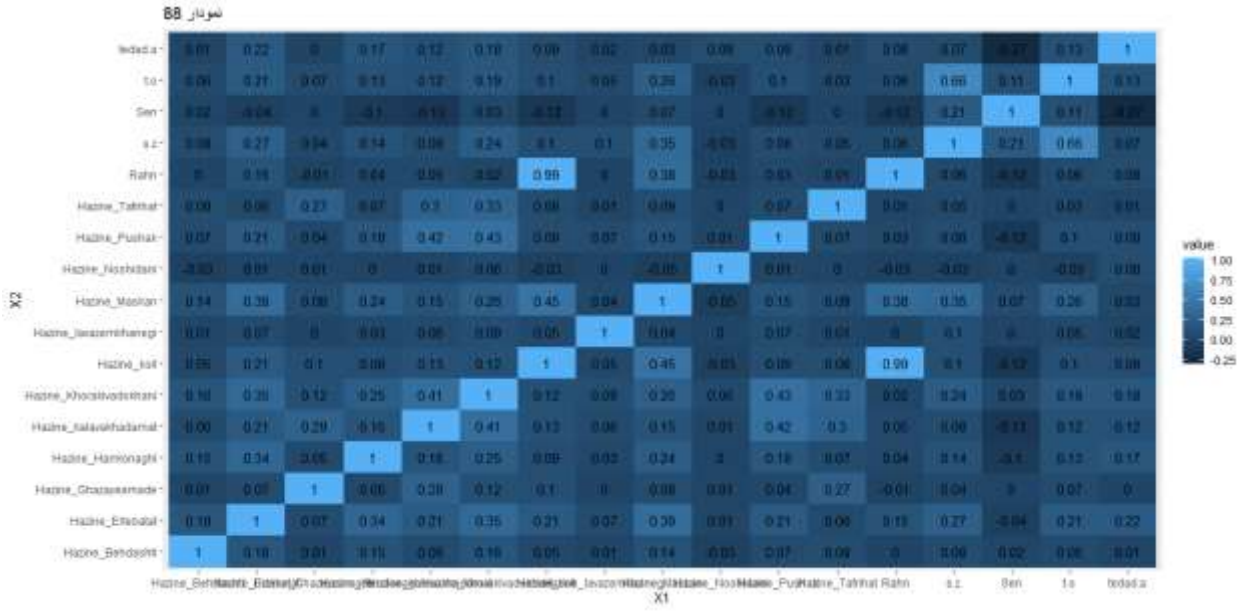


نمودار فراوانی برای لگاریتم هزینه کل: این نمودار فراوانی مقداری چولگی به راسن را نمابش می دهد.



نمودار جعبه ای هزینه کل: نمودار جعبه ای که محور افقی آن بودن یا نبودن در سه دهک اول اقتصادی است و محور عمودی هزینه کل.

این نمودار تاثیر گذاری هزینه کل بر بودن یا نبودن در سه دهک اول اقتصادی را نمایش می دهد.



نمودار 88 : نمودار حرارتی که نمایانگر همبستگی متغییر های عددی است.

می دانیم چنانچه همبستگی بین دو متغییر پیشگو زیاد باشد احتمال زیاد یکی از آن دو باید حذف شود یا این که از هر دو آن ها یک متغییر بسازیم. به طور مثال با توجه به همبستگی 0.99 بین متغییر رهن و هزینه کل می باشد که متغییر رهن بهتر است حذف شود.

فصل سوم:

1: کاهش بعد داده ها

2: ترجمه هدف داده کاوی به یک سوال داده کاوی مشخص تر

3: افراز داده ها

4: انتخاب فنون داده کاوی مناسب استفاده

5: به کار بردن الگوریتم های لازم

6: تفسیر نتایج الگوریتم ها

7: ارزیابی مدل ها و انتخاب مدل نهایی

1: کاهش بعد داده ها

با توجه به تصویر سازی تصمیم گرفتیم که:

1:متغییر استان از 5 رسته به 3 رسته ی: البرز، همدان، سایر(مرکزی یا قزوین یا قم) تبدیل می شود. (به علت نزدیک بودن میانگین درآمد هایشان)

2: متغییر جنس کار آمد خواهد بود.

3: سن به 5 بازه تبدیل شد و برای کاهش بعد هر فرد متعلق به هر بازه ای بود مقدار میانه ی آن بازه به او نسبت داده شد.

4: متغییر "تحصیل می کند" باید حذف شود.(هم فراوانی 1 خیلی کم است هم نمودار تجمیع نشان داد میانگین درامد تغییری نمی کند.)

5: در متغییر مدرک 1 و 9 را که بیانگر ابتدایی و سایر و غیر رسمی اند را بدلیل نزدیک بودن ارزش مدرک ها و میانگین درآمد ها در یک گروه می گنجانیم. به طریق مشابه 3و 4 را با هم ، 7و 8 هم با یکدیگر در یک گروه می گنجانم.(کاهش بعد)

(باتوجه به متغیرتحصیل میکند، مقادیر NA در این متغییر افراد بی سواد اند و به ان ها 0 نسبت دادم.)

6: از آن جا که متغییر مدرک کار آمد است و قصد دارم آن را نگه دارم پس باید متغییر سواد که در واقع زیر مجموعه ی مدرک است حذف شود.

7: در متغییر فعالیت، 4 مقدار: بیکار جویای کار، خانه دار، محصل، سایر را بدلیل فراوانی های کمشان و نزدیکی میانگین درآمدشان در یک گروه می بریم.

8: متغییر زناشویی اگر با جنسیت بررسی شود کارامد تر است.

9: تعداد اعضا به نظر متغییر کار آمدی است و ان را به 4 رسته: 1، 2، (3 یا 4)، (5 یا بزرگ تر) تبدیل می کنیم.

10: متغیر نحوه تصرف منزل : ملکی-اعیانی و خدمت در یک رسته. رایگان و سایر در یک رسته قرار می گیرد.

11: در متغییر تعداد اتاق، 1 و2 را با هم در نطر گرفته و همه ی اتاق های بزرگ تر از 7 را نیز یک رسته در نظر گرفتم.

12: متغییر مصالح عمده بنا کنار می رود.

13: این متغییر ها که بیانگر داشتن یا نداشتن چیزی است، کنار می روند:

تلویزیون سیاه و سفید، تلویزیون رنگی، گاز، کولر آبی متحرک، کولر گازی متحرک، آب لوله کشی، برق، لوله کشی گاز، حمام، آشپز خانه، برودت مرکزی، حرارت مرکزی، سوخت: آب، سوخت: گرما، سوخت: پخت و پز.

14: این متغییر ها که بیانگر داشتن یا نادشتن چیزی است، کارآمدند:

اتو، ویدیو، کامپیوتر، موبایل، جارو برقی، ماشین لباسشویی، ماشین ظرف شویی، ماکروویو، اینترنت، پکیج.

15: متغییر رهن به خاطر همبستگی 0.99 با متغییر هزینه کل کاندیدی برای حذف شدن است.

16: تحلیل مولفه های اصلی(PCA):

با توجه به نمودار حرارتی دیدیم که همبستگی سطح زیر بنا و تعداد اتاق نسبتا بالا است به همین دلیل با استفاده از PCA یک ترکیب خطی مناسب از آن ها را میابیم و نام آن را c_t.oands.z میگذاریم، که شامل 83 درصد از واریانس این دو متغییر است و آن را جایگزین این دو مولفه می کنیم.

2 : ترجمه هدف داده کاوی به یک سوال داده کاوی مشخص تر:

همانطور که اشاره شد، مسأله اکنون به زبان داده کاوی یک مسأله رده بندی دودویی است که باید برای هر خانوار پیشگویی شود که آیا خانوار عضو سه دهک اول جامعه هست یا خیر، که رده ی توفیقمان، متعلق بودن به سه دهک برتر جامعه است .

3: افراز داده ها

برای افراز داده ها، به طور تصادفی 70 درصد داده ها را به مجموعه داده ی آموزشی (Training Set)، 30 درصد به مجموعه داده ی اعتبارسنجی (Validation) منتسب میشود.

با استفاده از داده های آموزشی مدل را می سازم سپس از مدل ساخته شده؛ دقت ها و ماتریس های در هم ریختگی را برای هر دو مجموعه ی آموزشی و اعتبار سنجی می آورم، که می دانیم دقت روی داده های اعتبار سنجی مهم تر است چرا که در فرایند ساخته شدن مدل اثری نداشته است اما داشتن دقت روی مجوعه آموزشی نیز به ما کمک هایی می کند، مثلا در مواردی که بیش برازش رخ می دهد.

4: انتخاب فنون داده کاوی مناسب استفاده

با توجه به بخش دوم از فصل سوم که هدف را به طور کامل تر شرح دادم در این مسأله از تمامی روش های یادگیری راهنماییده می توان استفاده کرد، که هر کدام مزیت ها و معایبی دارند، که روش های : لجستیک، K- نزدیک ترین همسایگی، درخت رده بندی پیش فرض، درخت رده بندی عمیق، درخت رده بندی حرص شده بوسیله ی cp پایین تر و جنگل تصادفی را بر رسی خواهم کرد و نتایج و مزایا و معایب هر یک را شرح خواهم داد.

5 و 6: به کار بردن الگوریتم های داده کاوی و تفسیر نتایجشان:

روش اول: لجستیک

با بررسی های لازم، بهترین مدلی که بر روی داده های آموزشی بدست آمد شامل 23 متغییر انتخاب شده بوسیله Backward Elimination است. که این متغییر ها عبارتند از:

استان، جنسیت، مدرک، تعداد اعضا، نحوه تصرف منزل، اتو، ضبط، ویدیو، کامپیوتر، یخچال، ماشین لباسشویی، چرخ خیاطی، ماشین ظرفشویی، تلفن، اینترنت، کولرگازی ثابت، سن رسته ای شده و لگاریتم هزینه های: ارتباطات، غذای آماده، خوراکی-دخانی، لوازم منزل، مسکن و پوشاک.

در این مدل به جای متغییر های هزینه بدلیل چولگیشان از لگاریتمشان استفاده شد که باعث افزایش بسیار اندک accuracy و افزایش حدود 3 درصدی Specificity می شود.

حال خروجی ماتریس درهم ریختگی برای training set , validation set را می آورم: (برای هر دو از یک مدل که از مجموعه آموزشی آمده استفاده شده است.)

Training set: همانطور که می بینیم از1922 خانوار درون مجموعه آموزشی، مدل از 1342 خانواری که به آن ها یارانه تعلق نمی گرفت، 1195 مورد را درست رده بندی کرده که به معنای Sensitivity : 0.8905 است و از 580 خانواری که یارانه به آن ها تعلق می گرفت، 322 مورد درست رده بندی شده داریم، یعنی Specificity : 0.5552.

و در کل دقت Accuracy: 0.7893 خواهد بود.

```
          Reference
Prediction    0    1
         0 1195  258
         1  147  322

               Accuracy : 0.7893
                 95% CI : (0.7704, 0.8073)
    No Information Rate : 0.6982
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.4712

 Mcnemar's Test P-Value : 0.00000004604

            Sensitivity : 0.8905
            Specificity : 0.5552
         Pos Pred Value : 0.8224
         Neg Pred Value : 0.6866
             Prevalence : 0.6982
         Detection Rate : 0.6217
   Detection Prevalence : 0.7560
      Balanced Accuracy : 0.7228
```

validation set: با استفاده از همان مدل قبل این بار روی مجموعه اعتبار سنجی داریم: از 825 خانوار درون مجموعه اعتبار سنجی، مدل از 581 خانواری که به آن ها یارانه تعلق نمی گرفت، 523 مورد را درست رده بندی کرده که به معنای

Sensitivity : 0.9002 است و از 244 خانواری که یارانه به آن ها تعلق می گرفت، 125 مورد درست رده بندی شده داریم، یعنی Specificity : 0.5123.

و در کل دقت Accuracy: 0.7855 خواهد بود.

```
          Reference
Prediction   0   1
         0 523 119
         1  58 125

               Accuracy : 0.7855
                 95% CI : (0.7558, 0.813)
    No Information Rate : 0.7042
    P-Value [Acc > NIR] : 0.00000008645

                  Kappa : 0.4447

 Mcnemar's Test P-Value : 0.00000648655

            Sensitivity : 0.9002
            Specificity : 0.5123
         Pos Pred Value : 0.8146
         Neg Pred Value : 0.6831
             Prevalence : 0.7042
         Detection Rate : 0.6339
   Detection Prevalence : 0.7782
      Balanced Accuracy : 0.7062
```

روش دوم: K-نزدیک ترین همسایگی

برای پیاده سازی این روش لازم است متغییر های پیوسته ی خود را استاندارد کنیم تا مقیاس متغییر ها یکی شود.

توجه: در مواجه با متغییر های رسته ای با 3 یا بیش 3 رسته از متغییر های dummy استفاده می شود.

این الگوریتم در یک حلقه به ازای k های مختلف بر رسی شد، که بهترین دقت برای مجموعه اعتبار سنجی همراه با کم ترین بیش برازش متعلق به k= 14 بود.

```
1: (1, 0.6872727) - 2: (0.8459938, 0.6545455) - 3: (0.8584807, 0.6933333) -
4: (0.8132154, 0.710303) - 5: (0.817898, 0.7224242) - 6: (0.8059313,
0.7163636) - 7: (0.8059313, 0.750303) - 8: (0.792924, 0.7466667) - 9:
(0.7861602, 0.7466667) - 10: (0.780437, 0.7406061) - 11: (0.7861602,
0.7478788) - 12: (0.7778356, 0.750303) - 13: (0.7830385, 0.7539394) - 14
:(0.776795, 0.7614242) - 15: (0.7773153, 0.76) - 16: (0.7851197, 0.7587879) -
17: (0.7778356, 0.7418182) -  18: (0.7747138, 0.7406061) - 19: (0.7726327,
0.7563636) - 20: (0.7715921, 0.7612121)
```

حال برای مجموعه آموزشی و اعتبار سنجی آن، ماتریس در هم ریختگی را می آورم:

Training set: همانطور که می بینیم از1922 خانوار درون مجموعه آموزشی، مدل از 1342 خانواری که به آن ها یارانه تعلق نمی گرفت، 1263 مورد را درست رده بندی کرده که به معنای Sensitivity : 0.9411 است و از 580 خانواری که یارانه به آن ها تعلق می گرفت، 231 مورد درست رده بندی شده داریم، یعنی Specificity : 0.3983.

و در کل دقت Accuracy: 0.7773 خواهد بود.

```
          Reference
Prediction    0    1
         0 1263  349
```

```
         1   79  231

               Accuracy : 0.7773
                 95% CI : (0.758, 0.7957)
    No Information Rate : 0.6982
    P-Value [Acc > NIR] : 0.000000000000004768

                  Kappa : 0.3911

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9411
            Specificity : 0.3983
         Pos Pred Value : 0.7835
         Neg Pred Value : 0.7452
             Prevalence : 0.6982
         Detection Rate : 0.6571
   Detection Prevalence : 0.8387
      Balanced Accuracy : 0.6697
```

validation set: با استفاده از همان مدل قبل این بار روی مجموعه اعتبار سنجی داریم: از 825 خانوار درون مجموعه اعتبار سنجی، مدل از 581 خانواری که به آن ها یارانه تعلق نمی گرفت، 540 مورد را درست رده بندی کرده که به معنای Sensitivity : 0.9294 است و از 244 خانواری که یارانه به آن ها تعلق می گرفت، 87 مورد درست رده بندی شده داریم، یعنی Specificity : 0.3566.

و در کل دقت Accuracy: 0.76 خواهد بود.

```
          Reference
Prediction   0   1
         0 540 157
         1  41  87

               Accuracy : 0.76
                 95% CI : (0.7294, 0.7888)
    No Information Rate : 0.7042
    P-Value [Acc > NIR] : 0.0002034

                  Kappa : 0.3317
```

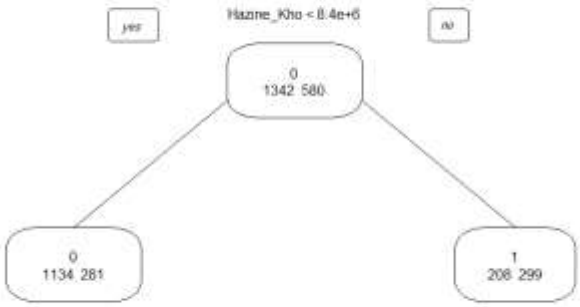
```
Mcnemar's Test P-Value : 0.0000000000000003016

           Sensitivity : 0.9294
           Specificity : 0.3566
        Pos Pred Value : 0.7747
        Neg Pred Value : 0.6797
            Prevalence : 0.7042
        Detection Rate : 0.6545
  Detection Prevalence : 0.8448
     Balanced Accuracy : 0.6430
```

روش سوم: درخت رده بندی

مزیت اصلی این روش عدم نیاز به پیش پردازش خاص است و همچنین می توان تمام مراحل برای رسیدن به رده بندی پایانی را متوجه شد.

و در آخر مدل پایانی از تمام متغییر ها استفاده نمی کند که در پیدا کردن داده های جدید موجب صرفه جویی در وقت و هزینه می شود.

برای مثال اگر بخواهیم تنها با یک سوال به رده بندی برسیم، بوسیله درخت ها ممکن است:



یعنی اگر بنا بر استفاده از تنها یک متغییر در رده بندی درخت روی مجموعه آموزشی ما باشد، آن متغییر هزینه خوراکی و دخانی خواهد بود. که دقت این درخت برای مجموعه آموزشی با یک متغییر با وجود این که از محک خام تنها 4 درصد بیشتر است، اما specifity در آن بیش از 50 درصد خواهد بود.

Training set:

```
          Reference
Prediction    0    1
         0 1134  281
         1  208  299

               Accuracy : 0.7456
                 95% CI : (0.7255, 0.7649)
    No Information Rate : 0.6982
    P-Value [Acc > NIR] : 0.000002475

                  Kappa : 0.3739

 Mcnemar's Test P-Value : 0.00113

            Sensitivity : 0.8450
            Specificity : 0.5155
         Pos Pred Value : 0.8014
         Neg Pred Value : 0.5897
```

```
           Prevalence : 0.6982
       Detection Rate : 0.5900
 Detection Prevalence : 0.7362
    Balanced Accuracy : 0.6803
```

و سپس با بر رسی همین درختی که روی مجموعه آموزشی بدست آمده ، روی مجموعه اعتبار سنجی، خواهیم دید دقت اندکی روی این مجموعه بیشتر می شود.

Validation set:

```
          Reference
Prediction   0   1
         0 498 118
         1  83 126

               Accuracy : 0.7564
                 95% CI : (0.7256, 0.7853)
    No Information Rate : 0.7042
    P-Value [Acc > NIR] : 0.0004879

                  Kappa : 0.3898

 Mcnemar's Test P-Value : 0.0164770

            Sensitivity : 0.8571
            Specificity : 0.5164
         Pos Pred Value : 0.8084
         Neg Pred Value : 0.6029
             Prevalence : 0.7042
         Detection Rate : 0.6036
   Detection Prevalence : 0.7467
      Balanced Accuracy : 0.6868
```

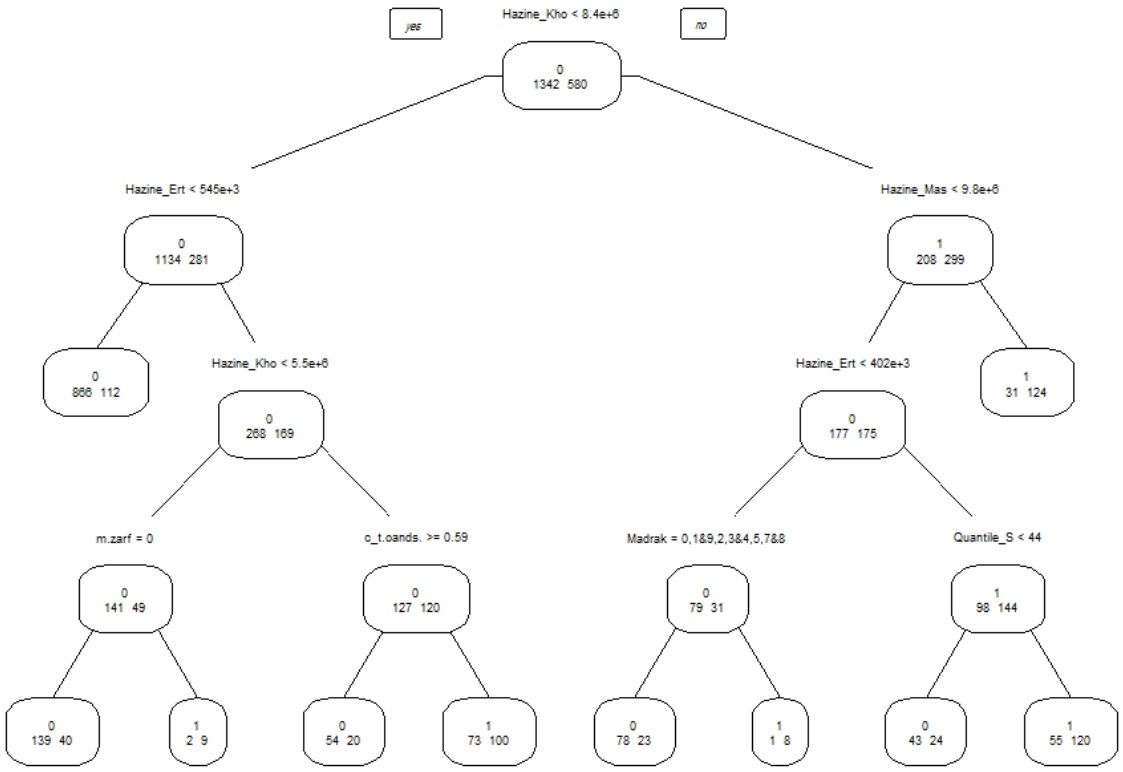
برا ی پیدا کردن مناسب ترین درخت که هم دقت بالایی داشته باشد هم با بیش برازش مواجه نباشیم، سه مدل از روی مجموعه ی آموزشی ساختم و و ماتریس های در هم ریختگی را روی این سه مدل هم روی مجموعه آموزشی هم اعتبارسنجی بررسی کردم که این سه مدل عبارت بودند از:

1: درخت رده بندی پیش فرض: به نظر مقداری بیش برازش وجودداشت.

2: درخت رده بندی عمیق: کاملا بیش برازش وجودداشت و با وجود دقت 100 درصدی روی مجموعه آموزشی، دقت روی مجموعه اعتبار سنجی چندان از محک خام بیشتر نبود.

3: درخت رده بندی حرص شده بوسیله ی cp پایین تر.

که در درخت رده بندی حرص شده بوسیله cp پایین تر هم بیش برازش رخ نداد هم دقت روی مجموعه اعتبار سنجی از همه بالا تر بود و متغییر های کم تری مورد استفاده ما قرار گرفته و رسم شکل و درک نحوه عملکرد الگوریتم نیز بسیار ساده است که در ادامه شکل درخت و ماترسی در هم ریختگی را برای این درخت آورده شده:



نکته1: با توجه به شکل در این مدل تنها از 7 متغییر پیشگو استفاده شده که همانطور که گفتیم در جمع آوری داده جدید بسیار مفید است، این 7 متغییر عبارتند از:

1: هزینه خوراکی و دخانی-2: هزینه مسکن- 3: هزینه ارتباطات-4: متغییری که بوسیله ی pca ساختیم- 5: ماشین ظرف شویی- 6: مدرک- 7: سن

نکته2: در صورتی که با یک ثبت در داده مواجه شویم، به سادگی با استفاده از این شکل خود ما نیز قادر هستیم جواب نهایی الگوریتم را بیاوریم.

حال ماتریس در هم ریختگی را برای با توجه به مدل ساخته شده از روی مجموعه آموزشی ، برای مجموعه آموزشی و اعتبار سنجی می آورم:

Train set: همانطور که می بینیم از1922 خانوار درون مجموعه آموزشی، مدل از 1342 خانواری که به آن ها یارانه تعلق نمی گرفت، 1180 مورد را درست رده بندی کرده که به معنای Sensitivity : 0.8793 است و از 580 خانواری که یارانه به آن ها تعلق می گرفت، 361 مورد درست رده بندی شده داریم، یعنی Specificity : 0.6224.

و در کل دقت Accuracy: 0.8018 خواهد بود.

```
          Reference
Prediction    0    1
         0 1180  219
         1  162  361

               Accuracy : 0.8018
                 95% CI : (0.7832, 0.8194)
    No Information Rate : 0.6982
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.5161
```

```
 Mcnemar's Test P-Value : 0.004118

            Sensitivity : 0.8793
            Specificity : 0.6224
         Pos Pred Value : 0.8435
         Neg Pred Value : 0.6902
             Prevalence : 0.6982
         Detection Rate : 0.6139
   Detection Prevalence : 0.7279
      Balanced Accuracy : 0.7508
```

Validation set: با استفاده از همان مدل قبل این بار روی مجموعه اعتبار سنجی داریم: از 825 خانوار درون مجموعه اعتبار سنجی، مدل از 581 خانواری که به آن ها یارانه تعلق نمی گرفت، 500 مورد را درست رده بندی کرده که به معنای Sensitivity : 0.8606 است و از 244 خانواری که یارانه به آن ها تعلق می گرفت، 144 مورد درست رده بندی شده داریم، یعنی Specificity : 0.5902.

و در کل دقت Accuracy: 0.7806 خواهد بود.

```
          Reference
Prediction   0   1
         0 500 100
         1  81 144

               Accuracy : 0.7806
                 95% CI : (0.7508, 0.8084)
    No Information Rate : 0.7042
    P-Value [Acc > NIR] : 0.0000004765

                  Kappa : 0.4612
```

```
Mcnemar's Test P-Value : 0.1809

           Sensitivity : 0.8606
           Specificity : 0.5902
        Pos Pred Value : 0.8333
        Neg Pred Value : 0.6400
            Prevalence : 0.7042
        Detection Rate : 0.6061
  Detection Prevalence : 0.7273
     Balanced Accuracy : 0.7254
```

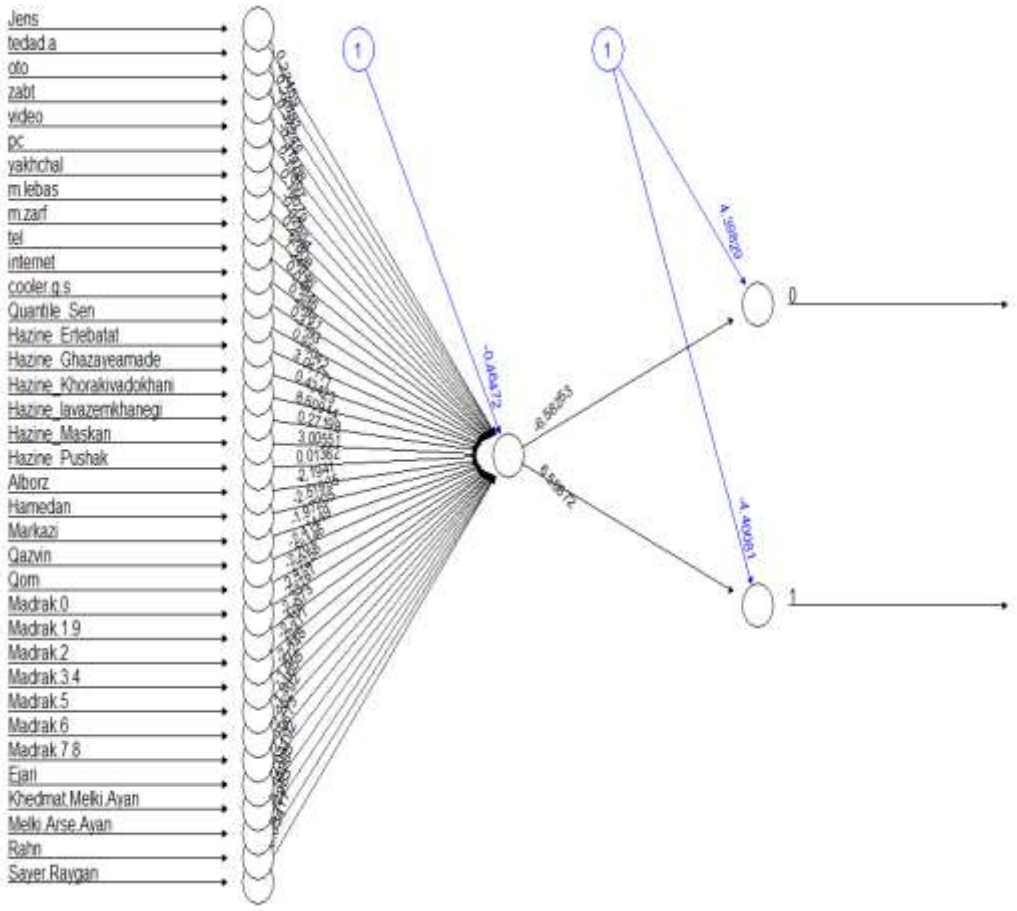
روش سوم: شبکه های عصبی

در اجرای این مدل، از متغییر هایی که بوسیله ی Backward Elimination در مدل لجستیک بدست آمدند، مورد استفاده قرار می گیرند.

همچنین در این روش باید تمام متغییر های کمی را نرمال کرده و تمام متغییر های رسته ای با بیش از دو رسته را به صورت متغییر dummy در آورد و متغییر های رسته ای با دو رسته را 0 و 1 کرد.

این الگوریتم را به ازای لایه های پنهان مختلفی امتحان کردم که در لایه پنهان اول هم دقت مناسب بود هم با بیش برازش مواجه نشدم، ولی برای لایه های پنهان بیشتر به شدت بیش برازش رخ می دهد.

نمودار مدل شبکه های عصبی با یک لایه پنهان:



حال بوسیله ی مدل بدست آمده از روی مجموعه آموزشی، برای مجموعه آموزشی و اعتبار سنجی ، ماتریس در هم ریختگی را می آورم:

Train set: همانطور که می بینیم از 1859 خانوار درون مجموعه آموزشی، مدل از 1299 خانواری که به آن ها یارانه تعلق نمی گرفت، 1166 مورد را درست رده بندی کرده که به معنای `Sensitivity : 0.8976` است و از 560 خانواری که یارانه به آن

ها تعلق می گرفت، 351 مورد درست رده بندی شده داریم، یعنی Specificity : 0.6268.

و در کل دقت Accuracy: 0.816 خواهد بود.

```
Prediction    0    1
         0 1166  209
         1  133  351

               Accuracy : 0.816
                 95% CI : (0.7977, 0.8334)
    No Information Rate : 0.6988
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5455

 Mcnemar's Test P-Value : 5.002e-05

            Sensitivity : 0.8976
            Specificity : 0.6268
         Pos Pred Value : 0.8480
         Neg Pred Value : 0.7252
             Prevalence : 0.6988
         Detection Rate : 0.6272
   Detection Prevalence : 0.7396
      Balanced Accuracy : 0.7622
```

Validation set: با استفاده از همان مدل قبل این بار روی مجموعه اعتبار سنجی داریم: از 888 خانوار درون مجموعه اعتبار سنجی، مدل از 624 خانواری که به آن ها یارانه تعلق نمی گرفت، 527 مورد را درست رده بندی کرده که به معنای Sensitivity : 0.8446 است و از 264 خانواری که یارانه به آن ها تعلق می گرفت، 161 مورد درست رده بندی شده داریم، یعنی Specificity : 0.6098.

و در کل دقت Accuracy: 0.7748 خواهد بود.

```
          Reference
Prediction   0   1
         0 527 103
         1  97 161
```

```
               Accuracy : 0.7748
                 95% CI : (0.7458, 0.8019)
    No Information Rate : 0.7027
    P-Value [Acc > NIR] : 8.663e-07

                  Kappa : 0.4574

 Mcnemar's Test P-Value : 0.7237

            Sensitivity : 0.8446
            Specificity : 0.6098
         Pos Pred Value : 0.8365
         Neg Pred Value : 0.6240
             Prevalence : 0.7027
         Detection Rate : 0.5935
   Detection Prevalence : 0.7095
      Balanced Accuracy : 0.7272
```

7: ارزیابی مدل ها و انتخاب مدل نهایی

با تفاسیری که از هر مدل آورده شد، و با توجه به این که درخت رده بندی حرص شده بوسیله ی cp پایین تر، از نظر دقت بسیار نزدیک به مدل لجستیک است ولی specificity آن بیشتر است و همچنین از متغییر های کمتری استفاده کرده و توضیح و درک آن بسیار ساده است، این مدل، یه عنوان مدل نهایی گزینه ی بهتری خواهد بود.

فصل چهارم:

پیاده سازی مدل:

در این بخش با توجه به مدل انتخاب شده ی ما که درخت رده بندی حرص شده بوسیله ی cp پایین تر است، با توجه به شکلی که آوردیم، یک مدل با 8 متغییر ساخته شد:

```
for (i in 1:nrow(df.piadesazi)) {

  if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 8399945& df.piadesazi[i,'Hazine_Ertebatat']<= 545000) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 5545500& df.piadesazi[i,'Hazine_Ertebatat']> 545000&
df.piadesazi[i,'m.zarf']== 0) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 5545500& df.piadesazi[i,'Hazine_Ertebatat']> 545000&
df.piadesazi[i,'m.zarf']== 1) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 8399945& df.piadesazi[i,'Hazine_Ertebatat']> 545000&
df.piadesazi[i,'Hazine_Khorakivadokhani']> 5545500& df.piadesazi[i,'c_t.oands.z']>=.587) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 8399945& df.piadesazi[i,'Hazine_Ertebatat']> 545000&
df.piadesazi[i,'Hazine_Khorakivadokhani']> 5545500& df.piadesazi[i,'c_t.oands.z']<.587) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i, 'Hazine_Maskan']>9805000) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i, 'Hazine_Maskan']<=9805000&
df.piadesazi[i, 'Hazine_Ertebatat']>= 402500& df.piadesazi[i, 'Quantile_Sen']> 44) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i, 'Hazine_Maskan']<=9805000&
df.piadesazi[i, 'Hazine_Ertebatat']>= 402500& df.piadesazi[i, 'Quantile_Sen']<= 44) {
```

```
   df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i, 'Hazine_Maskan']<=9805000&
df.piadesazi[i, 'Hazine_Ertebatat']< 402500& df.piadesazi[i, 'Madrak']== 6) {

   df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i, 'Hazine_Maskan']<=9805000&
df.piadesazi[i, 'Hazine_Ertebatat']< 402500& df.piadesazi[i, 'Madrak']!= 6) {

   df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }

}
```

ما با 15 پرسش نامه و از روی این مدل به این ماتریس در هم ریختگی رسیدیم:

```
          Reference
Prediction 0 1
         0 8 0
         1 2 4

               Accuracy : 0.8571
                 95% CI : (0.5719, 0.9822)
    No Information Rate : 0.7143
    P-Value [Acc > NIR] : 0.1904

                  Kappa : 0.6957

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 0.8000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.6667
             Prevalence : 0.7143
         Detection Rate : 0.5714
   Detection Prevalence : 0.5714
      Balanced Accuracy : 0.9000
```

که یعنی از 15 خانوار، تنها 2 مورد از آن هایی که باید به آن ها یارانه تعلق نمی گرفت، تعلق گرفته است.

البته این تعداد خانوار برای پیاده سازی مقداری کم است ولی به علت دشواری جمع آوری داده های جدید به نحوی که دارای توزیع مناسبی باشد، مقدور نبود به همین تعداد، قناعت کردم.

پیوست: تمامی کد هایی که در R استفاده شد:

```
df1 <-read.csv('data.csv',header = TRUE)

df <- read.csv('data.csv',header = TRUE)

unique(df$Address)

str(df$Address)

unique(df$C.O)


# according to our pre knowledge (0 is 'Markazi')&(13 is 'Hamedan')

# &(25 is 'Qom')&(26 is 'Qazvin')&(30 is 'Alborz')

for (i in 1:nrow(df)) {

if(is.na(df$Daramad_Mozd_Month[i])){

  df$Daramad_month_menhaie_year[i] = NA

}

else df$Daramad_month_menhaie_year[i]= 12*(df$Daramad_Mozd_Month[i])-
df$Daramad_Mozd_Year[i]


}

ggplot(df, aes(x= Daramad_kol_Mounth))+

  xlim(0, 200000000)+

  geom_histogram()




nrow(df[is.na(df$Daramad_Azad),])

#1960 Na value in Daramad_Azad which we will consider them as 1

for (i in 1:nrow(df)) {
```

```
  if (is.na(df$Daramad_Azad[i])) {
    df$Daramad_Azad[i]= 1
  }
}
nrow(df[which(df$Daramad_Azad==1),])
str(df$Daramad_Azad)
#In Following Lines we will do the same thing to other

for (i in 1:nrow(df)) {
  if (is.na(df$Daramad_Yarane[i])) {
    df$Daramad_Yarane[i]= 1
  }
}

for (i in 1:nrow(df)) {
  if (is.na(df$Daramad_Motefaraghe[i])) {
    df$Daramad_Motefaraghe[i]= 1
  }
}
for (i in 1:nrow(df)) {
  if (is.na(df$Daramad_Mozd_Month[i])) {
    df$Daramad_Mozd_Month[i]= 1
  }
}
for (i in 1:nrow(df)) {
  if (is.na(df$Daramad_Mozd_Year[i])) {
    df$Daramad_Mozd_Year[i]= 1
  }
}
```

```
Daramad_month_menhaie_year<- df$Daramad_month_menhaie_year

df$Daramad_month_menhaie_year <- NULL


#Makinig y

df$Daramad_kol_Mounth <-( df$Daramad_Azad+ df$Daramad_Motefaraghe+
df$Daramad_Mozd_Year+ df$Daramad_Yarane)/12

quantile(df$Daramad_kol_Mounth,probs = 0.7)

unique(df$C.O)

str(df$C.O)

table(df$C.O)

for (i in 1:nrow(df)) {
  if (df$C.O[i]== 0) {
    df$C.O[i] = 'Markazi'
  }else if (df$C.O[i]==13) {
    df$C.O[i]= 'Hamedan'
  }else if (df$C.O[i]==25){
    df$C.O[i]= 'Qom'
  }else if (df$C.O[i]==26) {
    df$C.O[i]= 'Qazvin'
  }else if (df$C.O[i]==30) {
    df$C.O[i]= 'Alborz'
  }
}


quantile(df$Daramad_kol_Mounth,probs = 0.7)

# = 24464000

for (i in 1:nrow(df)) {
  if (df$Daramad_kol_Mounth[i] > 24464000) {
```

```
   df$Cat.Daramad_kol_Mounth[i]= 1
  }else df$Cat.Daramad_kol_Mounth[i]= 0
}
#visualization
options(scipen =  999)
library(ggplot2)
ggplot(df, aes(x= C.O))+
  geom_bar(width = .6)+
  ggtitle('1 نمودار')

C.O1<- aggregate(df[,73], by= list(df$C.O), FUN= mean)
ggplot(C.O1, aes(x= Group.1, y= x))+
  geom_bar(stat = 'identity')+
  xlab('Ostan ha')+
  ylab('miangine daramad kol dar mah')+
  ggtitle('2 نمودار')
C.O2<- aggregate(df[,74],by= list(df$C.O), FUN= mean)
df$Cat.Daramad_kol_Mounth <- as.factor(df$Cat.Daramad_kol_Mounth)
ggplot(C.O2, aes(x= Group.1, y= x))+
  geom_bar(stat = 'identity')+
  xlab('Ostan ha')+
  ylab('Darsad taloq be 3 dahak bartar')+
  ggtitle('3 نمودار')
str(df$Jens)
for (i in 1:nrow(df)) {
  if (df$Jens[i]==2) {
    df$Jens[i]=0
  }
```

```
}
table(df$Jens)
#0 female and 1 is male
df$Jens <- as.factor(df$Jens)
ggplot(df, aes(x= Jens))+
  geom_bar(width = 0.6)+
  ggtitle('4 نمودار')

jens1<- aggregate(df[,73],by= list(df$Jens),FUN= mean)

ggplot(jens1,aes(x= Group.1, y=x))+
  geom_bar(width = 0.6, stat = 'identity')+
  xlab('Jens')+
  ylab('Miangine daramad kol mahiane')+
  ggtitle('5 نمودار')

table(df$Sen)

ggplot(df, aes(x= Sen))+
  geom_histogram(binwidth =5)+
  ggtitle('6 نمودار ')


ggplot(df, aes(x= Cat.Daramad_kol_Mounth, y= Sen))+
  geom_boxplot()+
  ggtitle('6.1')
```

```
ggplot(df, aes(x= Sen, y= Daramad_kol_Mounth))+
  geom_point(color= 'navy', alpha= 0.2)+
  ylim(0,100000000)+
  ggtitle('نمودار 7')

quantile(df$Sen, probs = c(0.2, 0.4, 0.6, 0.8))
Quantile_Sen= array()
for (i in 1:nrow(df)) {
  if (df$Sen[i]<= 35) {
    df$Quantile_Sen[i] = 'less than 35'
  }else if (df$Sen[i]>35 &df$Sen[i]<=43) {
    df$Quantile_Sen[i]='betwen 35 and 43'
  }else if (df$Sen[i]>43 & df$Sen[i]<=53) {
    df$Quantile_Sen[i]= 'between 43 and 53'
  }else if (df$Sen[i]>53& df$Sen[i]<=64) {
    df$Quantile_Sen[i]= 'between 53 and 64'
  }else if (df$Sen[i]>64) {
    df$Quantile_Sen[i]= 'more than 64'

  }
}
Quantile_Sen
unique(df$Quantile_Sen)


quantil1<- aggregate(df[,73],by= list(df$Quantile_Sen),FUN= mean)
q1 <- quantil1[4,]
```

```
q2 <- quantil1[3,]
q3 <- quantil1[1,]
q4 <- quantil1[2,]
q5 <- quantil1[5,]
quantil11<- rbind(q1,q2,q3,q4,q5)

ggplot(quantil11,aes(x=Group.1, y= x))+
  geom_bar(stat = 'identity')+
  xlab('Goroohe senni')+
  ylab('miangine daramad kol mahane')+
  ggtitle('8 نمودار')



ggplot(df, aes(x= Quantile_Sen, fill= factor(Cat.Daramad_kol_Mounth)))+
  geom_bar(width = .5)+
  labs(fill='Cat.Daramad_kol_Mounth' )+
  ggtitle('8.1 نمودار')



for (i in 1:nrow(df)) {
  if (df$Quantile_Sen[i]=='less than 35') {
    df$Quantile_Sen[i] = 28
  }else if (df$Quantile_Sen[i]=='betwen 35 and 43') {
    df$Quantile_Sen[i]= 40
  }else if (df$Quantile_Sen[i]== 'between 43 and 53') {
    df$Quantile_Sen[i]= 48
  }else if (df$Quantile_Sen[i]== 'between 53 and 64') {
```

```
   df$Quantile_Sen[i]= 58
  }else if (df$Quantile_Sen[i]== 'more than 64') {
   df$Quantile_Sen[i] = 78
  }
}

unique(df$Quantile_Sen)
str(df$Quantile_Sen)
df$Quantile_Sen <- as.integer(df$Quantile_Sen)
str(df$Quantile_Sen)
#Savad 1= ba savad va 2 = bi savad

for (i in 1:nrow(df)) {
  if (df$Savad[i]==2) {
   df$Savad[i]= 0
  }
}
#Savad 1 mean darad , savad 0 means nadarad
df$Savad <- as.factor(df$Savad)
ggplot(df, aes(x= Savad))+
  geom_bar(width = 0.5)+
  ggtitle('9 نمودار ')

Savad1 <- aggregate(df[,73],by= list(df$Savad),FUN= mean)
ggplot(Savad1, aes(x= Group.1, y= x))+
  geom_bar(width = .5, stat = 'identity')+
  xlab('Savad')+
  ylab('Miangin darmad kolle mahane')+
```

```
  ggtitle('10 نمودار')



#tahsil mikonad:
nulltahsilat <- df[which(is.na(df$Tahsil.Mikonad)),]

table(df$Tahsil.Mikonad)
for (i in 1:nrow(df)) {
  if (is.na(df$Tahsil.Mikonad[i])) {
    if(df$Savad[i]==0){
      df$Tahsil.Mikonad[i]= 0
    }
  }else if (df$Tahsil.Mikonad[i] ==2) {
    df$Tahsil.Mikonad[i]= 0
  }
}
table(df$Tahsil.Mikonad)
df$Tahsil.Mikonad <- as.factor(df$Tahsil.Mikonad)

ggplot(df, aes(x= Tahsil.Mikonad))+
  geom_bar(width = .5)+
  ggtitle('11 نمودار')

Tahsil.1<- aggregate(df[,73],by= list(df$Tahsil.Mikonad), FUN= mean)

ggplot(Tahsil.1, aes(x= Group.1,y =x))+
  geom_bar( width = .5, stat = 'identity')+
```

```
  xlab('Tahsi.Mikonad')+
  ylab('Miangin daramad mahane kol')+
  ggtitle('12 نمودار')
#Madrak variable:
NA.Madrak<- df[which(is.na(df$Madrak)),c(7,9)]
for (i in 1:nrow(df)) {
  if (is.na(df$Madrak[i])) {
    df$Madrak[i]= 0
  }else if (df$Madrak[i]==9|df$Madrak[i]==1) {
    df$Madrak[i]= '1& 9'
  }else if (df$Madrak[i]==8|df$Madrak[i]==7) {
    df$Madrak[i]= '7& 8'
  }else if (df$Madrak[i]==3|df$Madrak[i]==4) {
    df$Madrak[i]= '3& 4'
  }
}

unique(df$Madrak)

df$Madrak <- as.factor(df$Madrak)

ggplot(df, aes(x= Madrak))+
  geom_bar()+
  ggtitle('13 نمودار')

Madrak.1 <- aggregate(df[,73],by= list(df$Madrak), FUN= mean)
ggplot(Madrak.1, aes(x= Group.1, y= x))+
  geom_bar(stat = 'identity')+
```

```
  xlab('Madrak')+
  ylab('Miangin daramad kol mahiane')+
  ggtitle('نمودار 14')

# faaliat variable

faaaliat.Nul <- df[which(is.na(df$Faaliat)),c(4,5,6,7,8,9)]
#So no Null value is in  faaliat variable
for (i in 1:nrow(df)) {
  if (df$Faaliat[i]== 1) {
    df$Faaliat[i]= 'Shaghel'
  }else if (df$Faaliat[i]== 3) {
    df$Faaliat[i]= 'Daraie daramad bedune kar'
  }else if (df$Faaliat[i]== 6| df$Faaliat[i]== 5|df$Faaliat[i]== 2|df$Faaliat[i]== 4) {
    df$Faaliat[i]= 'Sayer'
  }
}
unique(df$Faaliat)
table(df$Faaliat)
#since we just hav one person who is Mohassel we put it in Sayer
ggplot(df, aes(x= Faaliat))+
  geom_bar()+
  ggtitle('نمودار 15')

Faaliat.1<- aggregate(df$Daramad_kol_Mounth,by= list(df$Faaliat),FUN= mean)
ggplot(Faaliat.1, aes(x= Group.1, y=x))+
  geom_bar(stat = 'identity')+
  xlab('Faaliat')+
```

```
  ylab('Miangine darmada kolle mahiane')+
  ggtitle('16 نمودار')
#Zanashooi variable
table(df$Zanashoi)

for (i in 1:nrow(df)) {
  if (df$Zanashoi[i]== 1) {
    df$Zanashoi[i]= 'daraye Hamsar'
  }else if (df$Zanashoi[i]== 2) {
    df$Zanashoi[i]= 'Fote hamsar'
  }else if (df$Zanashoi[i]== 3) {
    df$Zanashoi[i]= 'talaghe hamsar'
  }else if (df$Zanashoi[i]== 4) {
    df$Zanashoi[i]= 'ezdevaj nakarde'
  }
}
unique(df$Zanashoi)
table(df$Zanashoi)


ggplot(df, aes(x= Zanashoi))+
  geom_bar()+
  ggtitle('17 نمودار')


zanashoi.1<- aggregate(df$Daramad_kol_Mounth,by=list(df$Zanashoi,df$Jens),FUN= mean)
zanashoi.2<- aggregate(df$Daramad_kol_Mounth,by=list(df$Zanashoi,df$Jens),FUN= mean, drop=
FALSE)
ggplot(zanashoi.1, aes(x= Group.1, y= x))+
```

```
  geom_bar(stat = 'identity')+
  xlab('Zanashoi')+
  ylab('Miangine darmada koll')+
  ggtitle('18 نمودار ')
ggplot(zanashoi.1, aes(x= Group.1, y= x))+
  geom_bar(stat = 'identity')+
  facet_wrap(~Group.2)+
  xlab('Zanashoi')+
  ylab('Miangine darmada koll')+
  ggtitle('19 نمودار                                                    Jens')



#tedad.a variable
alaki <- df[which(df$tedad.a==10),]
str(df$tedad.a)
sort(unique(df$tedad.a))
table(df$tedad.a)
for (i in 1:nrow(df)) {
  if (df$tedad.a[i]>=6) {
    df$tedad.a[i]= 6
  }
}
unique(df$tedad.a)
table(df$tedad.a)

for (i in 1:nrow(df)) {
  if (df$tedad.a[i]==6) {
    df$tedad.a[i]= 'greater than 6'
```

```
  }
}

ggplot(df, aes(x= tedad.a))+
  geom_bar()+
  ggtitle('20 نمو دار')



tedad.1<- aggregate(df$Daramad_kol_Mounth,by= list(df$tedad.a),FUN= mean)
ggplot(tedad.1, aes(x= Group.1,y= x))+
  geom_bar(stat = 'identity')+
  xlab('tedad.a')+
  ylab('Miangine kolle daramad mahiane')+
  ggtitle('21 نمودار')



for (i in 1:nrow(df)) {
  if (df$tedad.a[i]== 3| df$tedad.a[i]== 4) {
    df$tedad.a[i]= '3& 4'
  } else if (df$tedad.a[i]== 5| df$tedad.a[i]== 'greater than 6') {
    df$tedad.a[i]= 'greater than 5'
  }
}

unique(df$tedad.a)
#Now n.t.m variable
#1 mean Melki arse o ayan
str(df$n.t.m)
```

```
table(df$n.t.m)


for (i in 1:nrow(df)) {
  if (df$n.t.m[i]== 1) {
    df$n.t.m[i]= 'Melki-Arse-Ayan'
  }else if (df$n.t.m[i]== 2) {
    df$n.t.m[i]= 'Melki-Ayan'
  }else if (df$n.t.m[i]== 3) {
    df$n.t.m[i]= 'Ejari'
  }else if (df$n.t.m[i]== 4) {
    df$n.t.m[i]= 'Rahn'
  }else if (df$n.t.m[i]== 5) {
    df$n.t.m[i]= 'Khedmat'
  }else if (df$n.t.m[i]== 6) {
    df$n.t.m[i]= 'Raygan'
  }else df$n.t.m[i]= 'Sayer'
}


ggplot(df, aes(x= n.t.m))+
  geom_bar()+
  ggtitle('22نمودار')
table(df$n.t.m)
n.tm.1 <- aggregate(df$Daramad_kol_Mounth,by= list(df$n.t.m),FUN= mean)


ggplot(n.tm.1, aes(x= Group.1, y= x))+
  geom_bar(stat = 'identity')+
  xlab('n.t.m')+
  ylab('Miangin Daramad Kol mahane')+
```

```
  ggtitle('23 نمودار')
for (i in 1:nrow(df)) {
  if (df$n.t.m[i]=='Sayer'|df$n.t.m[i]== 'Raygan') {
    df$n.t.m[i]= 'Sayer&Raygan'
  }else if (df$n.t.m[i]== 'Khedmat'| df$n.t.m[i]== 'Melki-Ayan') {
    df$n.t.m[i]= 'Khedmat&Melki-Ayan'
  }
}
table(df$n.t.m)
#t.o variable
unique(df$t.o)

ggplot(df, aes(x= factor(df$t.o)))+
  geom_bar()+
  ggtitle('24 نمودار')
table(df$t.o)

t.o.1 <- aggregate(df$Daramad_kol_Mounth,by= list(df$t.o),FUN= mean)

ggplot(t.o.1,aes(x=factor(Group.1), y= x))+
  geom_bar(stat = 'identity')+
  xlab('t.O')+
  ylab('Miangine kole daramad mahiane')+
  ggtitle('25 نمودار')
t.o.2<- aggregate(df$Daramad_kol_Mounth,by=list(df$t.o, df$n.t.m),drop= FALSE,FUN= mean)

ggplot(t.o.2, aes(x= factor(Group.1), y= x))+
  geom_bar(stat = 'identity')+
```

```
  facet_wrap(~factor(Group.2))+
  xlab('T.O')+
  ylab('Miangine daramad kolle mahane')+
  ggtitle('26نمودار                                    نحوه تصرف منزل')
unique(df$t.o)

for (i in 1:nrow(df)) {
  if (df$t.o[i]== 1| df$t.o[i]== 2) {
    df$t.o[i]= '1 or 2'
  }else if (df$t.o[i]>= 7) {
    df$t.o[i] = '7 or greater'
  }
}

#S.Z sathe zir bana variable

ggplot(df, aes(x= s.z, y= Daramad_kol_Mounth))+
  geom_point(alpha= 0.6)+
  ylim(0, 10000000)+
  ggtitle('27 نمودار')
#n.e noe eskelete bana
unique(df$n.e)
table(df$n.e)
for (i in 1:nrow(df)) {
  if (df$n.e[i]== 1) {
    df$n.e[i]= 'Felezi'
  }else if (df$n.e[i]== 2) {
    df$n.e[i]= 'Beton-Arme'
```

```
  }else if (df$n.e[i]== 3) {
    df$n.e[i]= 'Sayer'
  }
}
ggplot(df, aes(x= n.e))+
  geom_bar(width = .5)+
  ggtitle('28نمودار')
n.e.1 <- aggregate(df$Daramad_kol_Mounth,by= list(df$n.e),FUN= mean)
ggplot(n.e.1, aes(x= Group.1, y= x))+
  geom_bar(width = 0.5, stat = 'identity')+
  xlab('n.e')+
  ylab('miangine daramad kol mahane')+
  ggtitle('29 نمودار')




#m.o.b: masale omde bana variable
for (i in 1:nrow(df)) {
  if (is.na(df$m.o.b[i])) {
    df$m.o.b[i]='NULL'
  }
}
unique(df$m.o.b)
table(df$m.o.b)
ggplot(df,aes(x= m.o.b))+
  geom_bar(width = .5)+
```

```
  ggtitle('30 نمودار')
# otoo variable
for (i in 1:nrow(df)) {
  if (is.na(df$oto[i])) {
    df$oto[i]= 0
  }
}
tabel(df$oto)
df$oto <- as.factor(df$oto)
ggplot(df, aes(x= oto))+
  geom_bar(width = 0.5)+
  ggtitle('31 نمودار')

oto.1<- aggregate(df$Daramad_kol_Mounth,by=list(df$oto), mean)
ggplot(oto.1, aes(x= Group.1, y= x))+
  geom_bar(width = 0.5, stat = 'identity')+
  xlab('oto')+
  ylab('miangine darmad kolli mahane')+
  ggtitle('32 نمودار')
ggplot(df, aes(x= oto, y= Daramad_kol_Mounth))+
  geom_boxplot()+
  ylim(0,100000000)+
  ggtitle('33نمودار')
box.oto<- ggplot(df, aes(x= oto, y= Daramad_kol_Mounth))+
  geom_boxplot()+
  ylim(0,100000000)
```

به علت تکراری بودن عملیات روی متغییر های دو دویی، فقط برای متغییر اتو را آوردم.

```
# box plots

library(gridExtra)
grid.arrange(box.oto, box.mo, box.do, box.radio, box.zabt, box.tv.s, ncol= 3)
grid.arrange(box.tv.r,box.video, box.pc, box.mobile, box.freeizer, box.yakhchal, ncol= 3)
grid.arrange(box.yakhchal.f, box.gaz, box.jaro.b, box.m.lebas, box.charkh.kh, box.panke, ncol= 3)

grid.arrange(box.cooler.a, box.cooler.g,box.m.zarf, box.microfer, box.tel, box.internet, ncol= 3)

grid.arrange(box.hamam, box.cooler.a.s, box.hararat.m, box.package, box.cooler.g.s,
box.fazelab, ncol= 3)

grid.arrange(box.m.lebas, box.charkh.kh, box.panke(box.cooler.a, box.cooler.g,
        box.m.zarf, box.microfer, box.tel, box.internet,
        box.hamam, box.cooler.a.s, box.hararat.m, box.package, box.cooler.g.s,
        box.fazelab, ncol= 5, top= '64 نمودار')

#sookht.p variable

table(df$sookht.p)
for (i in 1:nrow(df)) {
  if (df$sookht.p[i]== 3) {
    df$sookht.p[i]= 'gaz maye'
  }else if (df$sookht.p[i]== 4) {
    df$sookht.p[i]= 'gaz tabiE'
```

```
  }else if (df$sookht.p[i]== 5) {
    df$sookht.p[i]= 'bargh'
  }

}

ggplot(df, aes(x= sookht.p))+
  geom_bar(width = .5)+
  ggtitle('65 نمودار')
#sookht.g variable
table(df$sookht.g)
for (i in 1:nrow(df)) {
  if (df$sookht.g[i]== 11) {
    df$sookht.g[i]= 'nafte-sefid'
  }else if (df$sookht.g[i]== 13) {
    df$sookht.g[i]= 'gaz-maye'
  }else if (df$sookht.g[i]== 14) {
    df$sookht.g[i]= 'gaze tabie'
  }else if (df$sookht.g[i]== 15) {
    df$sookht.g[i]= 'bargh'
  }
}
ggplot(df, aes(x= sookht.g))+
  geom_bar(width = 0.5)+
  ggtitle('66 نمودار')
#sookht.ab
table(df$sookht.ab)
```

```
#this is like sookht.g and and sookht.p
#Hazine ha
#hazine behdasht

for (i in 1:nrow(df)) {
  if (is.na(df$Hazine_Behdashti[i])) {
    df$Hazine_Behdashti[i]= 1
  }
}

ggplot(df, aes(x= Hazine_Behdashti))+
  geom_histogram()+
  xlim(0, 20000000)+
  ggtitle('66 نمودار')

#It is not normal distribution:
ggplot(df, aes(x= log10(Hazine_Behdashti)))+
  geom_histogram()+
#  xlim(0, 20000000)+
  ggtitle('66.1 نمودار')
ggplot(df, aes(x= Hazine_Behdashti, y= Daramad_kol_Mounth))+
  geom_point(alpha= 0.3,colour= 'navy')+
  xlim(0, 25000000)+
  ylim(0,200000000)+
  ggtitle('67 نمودار')
ggplot(df, aes(x= Cat.Daramad_kol_Mounth, y= Hazine_Behdashti))+
  geom_boxplot()+
  ylim(0, 2000000)+
```

```
 ggtitle('نمودار 67')
```

به علت تکراری بودن عملیات روی متغییر های هزینه، فقط برای متغییر هزینه بهداشتی را آوردم.

```
#Heat-Map
#making correlation matrix:
names(df)
#sen, tedad.a, t.o, s.z, hazine_(behdashti, ertebati, ghazaieamade, hamlonaghl,
#kalava khedamat, khorakivadokhani, lavazemkhanegi)
for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Behdashti[i])) {
    df1$Hazine_Behdashti[i]= 0
  }
}

for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Ertebatat[i])) {
    df1$Hazine_Ertebatat[i]= 0
  }
}



for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Ghazayeamade[i])) {
    df1$Hazine_Ghazayeamade[i]= 0
```

```
  }
}



for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Hamlonaghl[i])) {
    df1$Hazine_Hamlonaghl[i]= 0
  }
}

for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_kalavakhadamat[i])) {
    df1$Hazine_kalavakhadamat[i]= 0
  }
}

for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Khorakivadokhani[i])) {
    df1$Hazine_Khorakivadokhani[i]= 0
  }
}

for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_lavazemkhanegi[i])) {
    df1$Hazine_lavazemkhanegi[i]= 0
  }
}
```

```
for (i in 1:nrow(df1)) {
  if (is.na(df1$Rahn[i])) {
    df1$Rahn[i]= 0
  }
}

for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Noshidani[i])) {
    df1$Hazine_Noshidani[i]= 0
  }
}

for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Pushak[i])) {
    df1$Hazine_Pushak[i]= 0
  }
}

for (i in 1:nrow(df1)) {
  if (is.na(df1$Hazine_Tafrihat[i])) {
    df1$Hazine_Tafrihat[i]= 0
  }
}
```

```
df1$Hazine_koll <- df1$Hazine_Behdashti+ df1$Hazine_Ertebatat+ df1$Hazine_Ghazayeamade+
df1$Hazine_Hamlonaghl+ df1$Hazine_kalavakhadamat+ df1$Hazine_Khorakivadokhani+
df1$Hazine_lavazemkhanegi+ df1$Hazine_Maskan+ df1$Hazine_Noshidani+ df1$Hazine_Pushak+
df1$Hazine_Tafrihat+ df1$Rahn



df.numeric<- df1[,c('Sen','tedad.a','t.o', 's.z', 'Hazine_Behdashti', 'Hazine_Ertebatat',
                    'Hazine_Ghazayeamade', 'Hazine_Hamlonaghl', 'Hazine_kalavakhadamat',
                    'Hazine_Khorakivadokhani', 'Hazine_lavazemkhanegi', 'Rahn', 'Hazine_Maskan',
                    'Hazine_Noshidani', 'Hazine_Pushak', 'Hazine_Tafrihat','Hazine_koll')]



df1$Hazine_koll <- df1$Hazine_Behdashti+ df1$Hazine_Ertebatat+ df1$Hazine_Ghazayeamade+
df1$Hazine_Hamlonaghl+ df1$Hazine_kalavakhadamat+ df1$Hazine_Khorakivadokhani+
df1$Hazine_lavazemkhanegi+ df1$Hazine_Maskan+ df1$Hazine_Noshidani+ df1$Hazine_Pushak+
df1$Hazine_Tafrihat+ df1$Rahn


df$Hazine_Kol

str(df.numeric$Rahn)


for (i in 1:16) {
  print (typeof(df.numeric[i,i]))
}


str(df.numeric$tedad.a)


cor.matrix <- cor(df.numeric, method = 'pearson')

library(reshape)

cor.matrix <- round(cor.matrix, 2)

melted.cor.mat <- melt(cor.matrix)
```

```
ggplot(melted.cor.mat, aes(x= X1, y= X2, fill= value))+
  geom_tile()+
  geom_text(aes(x= X1, y= X2, label= value))+
  ggtitle('88 نمودار')




df$Hazine_Kol <- df1$Hazine_koll

for (i in 1:nrow(df)) {
  if (df$Hazine_Kol[i]== 0) {
    df$Hazine_Kol[i]= 1
  }
}

ggplot(df, aes(x= log10(Hazine_Kol)))+
  geom_histogram()+
  ggtitle('نمودار فراوانی برای لگاریتم هزینه کل')



ggplot(df, aes(x= Cat.Daramad_kol_Mounth, y= Hazine_Kol))+
  geom_boxplot()+
  ylim(0, 100000000)+
  ggtitle('نمودار جعبه ای هزینه کل')
```

```
#############################################################################################
#PCA
pca <- prcomp(data.frame(df.numeric$t.o, df.numeric$s.z), scale= TRUE)
pca
summary(pca)
options(scipen =  999)
pca$sdev
pca$rotation[2,1]
t.o.scale <- scale(df.numeric$t.o)
s.z.scale <- scale(df.numeric$s.z)
ss<- pca$rotation[1,1]* t.o.scale+ pca$rotation[2,1]* s.z.scale
df$c_t.oands.z<- ss[,1]
class(df$c_t.oands.z)
str(df$c_t.oands.z)

#Logestic

set.seed(2564)
library(caTools)
split <- sample.split(df$Daramad_kol_Mounth, SplitRatio = 0.7)
split
train <- subset(df, split== TRUE)
train.copy<- df[split== TRUE, which(names(df)%in%names(train))]
test <- subset(df, split== FALSE)
train$m.o.b <- NULL
train$ab.l <- NULL
train$bargh<- NULL
train$Tahsil.Mikonad <- NULL
```

```
train$Savad <- NULL

train$tv.s <- NULL

train$tv.r <- NULL

train$gaz <- NULL

train$cooler.a <- NULL

train$cooler.g <- NULL

train$gaz.l <- NULL

train$hamam <- NULL

train$ashpazkhane <- NULL

train$broodat.m <- NULL

train$hararat.m <- NULL

train$sookht.p <- NULL

train$sookht.g <- NULL

train$sookht.ab <- NULL

train$Sen <- NULL

train$Rahn <- NULL

train$Address <- NULL

train$MahMorajeh <- NULL

train$Fasl <- NULL

#View(train)

nrow(train)+ nrow(test)== nrow(df)

library(forecast)

str(train$Cat.Daramad_kol_Mounth)

#logestic <- glm(Cat.Daramad_kol_Mounth~., data = train[,c(1, 2, 5, 7, 52, 3, 6, 8, 11, 17, 22, 23, 27, 29, 34, 35, 37, 41, 44, 53, 51)], family = "binomial")

logestic <- glm(Cat.Daramad_kol_Mounth~., data = train[,c(1, 2, 5, 7, 52, 3, 6, 8, 11, 17, 22, 23, 27, 29, 34, 35, 37, 41, 44, 53, 51)], family = "binomial")

logestic
```

```
kh<-df[,c('Daramad_Mozd_Year',)]

summary(logestic)
#train set

glm.probs.train <- predict(logestic, newdata = train, type= 'response')
glm.probs.train
glm.pred.train <- ifelse(glm.probs.train > 0.5, 1, 0)
glm.pred.train
table(glm.pred.train, train$Cat.Daramad_kol_Mounth)
aa <- table(glm.pred.train, train$Cat.Daramad_kol_Mounth)
(aa[1,1]+ aa[2,2])/(aa[1,1]+ aa[2,2]+ aa[1,2]+ aa[2,1])

confusionMatrix(as.factor(glm.pred.train), train$Cat.Daramad_kol_Mounth)

#Test set
glm.probs.test = predict(logestic, newdata = test, type = "response")

glm.probs.test
glm.pred.test <- ifelse(glm.probs.test > 0.5, 1, 0)
glm.pred.test
table(glm.pred.test, test$Cat.Daramad_kol_Mounth)
bb <- table(glm.pred.test, test$Cat.Daramad_kol_Mounth)
(bb[1,1]+ bb[2,2])/(bb[1,1]+ bb[2,2]+ bb[1,2]+ bb[2,1])
confusionMatrix(as.factor(glm.pred.test), test$Cat.Daramad_kol_Mounth)
# acc <-  0.7927273

#Using backward for feature selection
logestic.back <- glm(Cat.Daramad_kol_Mounth~., data = train[,-c(45:50)], family = "binomial")
```

```
logestic.back



step(logestic.back, direction = 'backward')
logestic.back <- glm(formula = Cat.Daramad_kol_Mounth ~ C.O + Jens + Madrak +
    tedad.a + n.t.m + oto + zabt + video + pc + yakhchal + m.lebas +
    charkh.kh + m.zarf + tel + internet + cooler.g.s + Hazine_Ertebatat +
    Hazine_Ghazayeamade + Hazine_Khorakivadokhani + Hazine_lavazemkhanegi +
    Hazine_Maskan + Hazine_Pushak + Quantile_Sen, family = "binomial",
  data = train[, -c(45:50)])

glm.probs.train.back <- predict(logestic.back, newdata = train, type= 'response')

glm.probs.train.back

glm.pred.train.back <- ifelse(glm.probs.train.back > 0.5, 1, 0)
glm.pred.train.back
table(glm.pred.train.back, train$Cat.Daramad_kol_Mounth)
cc <- table(glm.pred.train.back, train$Cat.Daramad_kol_Mounth)
(cc[1,1]+ cc[2,2])/(cc[1,1]+ cc[2,2]+ cc[1,2]+ cc[2,1])
confusionMatrix(as.factor(glm.pred.train.back), train$Cat.Daramad_kol_Mounth)



glm.probs.test.back <- predict(logestic.back, newdata = test, type= 'response')

glm.probs.test.back

glm.pred.test.back <- ifelse(glm.probs.test.back > 0.5, 1, 0)
glm.pred.test.back
```

```
table(glm.pred.test.back, test$Cat.Daramad_kol_Mounth)
dd <- table(glm.pred.test.back, test$Cat.Daramad_kol_Mounth)
(dd[1,1]+ dd[2,2])/(dd[1,1]+ dd[2,2]+ dd[1,2]+ dd[2,1])
#0.7830303

confusionMatrix(as.factor(glm.pred.test.back), test$Cat.Daramad_kol_Mounth)

#Add log to see diffrences
train.log<- train
train.log$logHazine_Ertebatat <- log10(train$Hazine_Ertebatat)
train.log$logHazine_Ghazayeamade <- log10(train$Hazine_Ghazayeamade)
train.log$logHazine_Khorakivadokhani <- log10(train$Hazine_Khorakivadokhani)
train.log$logHazine_lavazemkhanegi <- log10(train$Hazine_lavazemkhanegi)
train.log$logHazine_Maskan <- log10(train$Hazine_Maskan)
train.log$logHazine_Pushak <- log10(train$Hazine_Pushak)
test.log<- test
test.log$logHazine_Ertebatat <- log10(test$Hazine_Ertebatat)
test.log$logHazine_Ghazayeamade <- log10(test$Hazine_Ghazayeamade)
test.log$logHazine_Khorakivadokhani <- log10(test$Hazine_Khorakivadokhani)
test.log$logHazine_lavazemkhanegi <- log10(test$Hazine_lavazemkhanegi)
test.log$logHazine_Maskan <- log10(test$Hazine_Maskan)
test.log$logHazine_Pushak <- log10(test$Hazine_Pushak)

logestic.back.log <- glm(formula = Cat.Daramad_kol_Mounth ~ C.O + Jens + Madrak +
                         tedad.a + n.t.m + oto + zabt + video + pc + yakhchal + m.lebas +
                         charkh.kh + m.zarf + tel + internet + cooler.g.s + logHazine_Ertebatat +
                         logHazine_Ghazayeamade + logHazine_Khorakivadokhani + logHazine_lavazemkhanegi +
                         logHazine_Maskan + logHazine_Pushak + Quantile_Sen, family = "binomial",
                         data = train.log[, -c(45:50)])
```

```
#TrainSet

glm.probs.train.back.log <- predict(logestic.back.log, newdata = train.log, type= 'response')

glm.probs.train.back.log

glm.pred.train.back.log <- ifelse(glm.probs.train.back.log > 0.5, 1, 0)
glm.pred.train.back.log
table(glm.pred.train.back.log, train$Cat.Daramad_kol_Mounth)
ee <- table(glm.pred.train.back.log, train$Cat.Daramad_kol_Mounth)
(ee[1,1]+ ee[2,2])/(ee[1,1]+ ee[2,2]+ ee[1,2]+ ee[2,1])
confusionMatrix(as.factor(glm.pred.train.back.log), train$Cat.Daramad_kol_Mounth)

#Test Set
glm.probs.test.back.log <- predict(logestic.back.log, newdata = test.log, type= 'response')

glm.probs.test.back.log

glm.pred.test.back.log <- ifelse(glm.probs.test.back.log > 0.5, 1, 0)
glm.pred.test.back.log
table(glm.pred.test.back.log, test$Cat.Daramad_kol_Mounth)
ff <- table(glm.pred.test.back.log, test$Cat.Daramad_kol_Mounth)
(ff[1,1]+ ff[2,2])/(ff[1,1]+ ff[2,2]+ ff[1,2]+ ff[2,1])

confusionMatrix(as.factor(glm.pred.test.back.log), test$Cat.Daramad_kol_Mounth)



#KNN
library(caret)
```

```
str(df)
df.KNN <- df1
df.KNN[is.na(df.KNN)] <- 0
df.KNN$n.t.m<- df$n.t.m
df.KNN$n.e<- df$n.e
#KNN works only numeric so we shuld do some pre processing
df.KNN$m.o.b <- NULL
df.KNN$ab.l <- NULL
df.KNN$bargh<- NULL
df.KNN$Tahsil.Mikonad <- NULL
df.KNN$Savad <- NULL
df.KNN$tv.s <- NULL
df.KNN$tv.r <- NULL
df.KNN$gaz <- NULL
df.KNN$cooler.a <- NULL
df.KNN$cooler.g <- NULL
df.KNN$gaz.l <- NULL
df.KNN$hamam <- NULL
df.KNN$ashpazkhane <- NULL
df.KNN$broodat.m <- NULL
df.KNN$hararat.m <- NULL
df.KNN$sookht.p <- NULL
df.KNN$sookht.g <- NULL
df.KNN$sookht.ab <- NULL
#df.KNN$Rahn <- NULL
#df.KNN$Address <- NULL
#df.KNN$MahMorajeh <- NULL
#df.KNN$Fasl <- NULL
df.KNN$C.O <- df$C.O
```

```
str(df.KNN$oto)


df.KNN$Cat.Daramad_kol_Mounth <- df$Cat.Daramad_kol_Mounth


df.KNN$Hazine_koll<- df$Hazine_Kol
df.KNN$Madrak<- df$Madrak
df.KNN$Faaliat<- df$Faaliat
df.KNN$Zanashoi <- df$Zanashoi
str(df.KNN$Address)
library(dummies)
df.KNN <- cbind(df.KNN, dummy(df.KNN$Madrak, sep = "_"))
df.KNN <- cbind(df.KNN, dummy(df.KNN$Faaliat, sep = "_"))
df.KNN <- cbind(df.KNN, dummy(df.KNN$Zanashoi, sep = "_"))
df.KNN <- cbind(df.KNN, dummy(df.KNN$C.O, sep = "_"))
df.KNN <- cbind(df.KNN, dummy(df.KNN$n.t.m, sep = "_"))
df.KNN <- cbind(df.KNN, dummy(df.KNN$n.e, sep = "_"))



df.KNN$n.e<- NULL
df.KNN$n.t.m<- NULL
df.KNN$Madrak<- NULL
df.KNN$Faaliat<- NULL
df.KNN$Zanashoi<- NULL
df.KNN$C.O<- NULL
set.seed(2564)
library(caTools)

#Making a data normalized
```

```
df.KNN$Cat.Daramad_kol_Mounth<-NULL
for (i in 1:ncol(df.KNN)) {
  df.KNN[,i]<- (df.KNN[,i]- min(df.KNN[,i]))/(max(df.KNN[,i])- min(df.KNN[,i]))
}

df.KNN$Cat.Daramad_kol_Mounth<- df$Cat.Daramad_kol_Mounth



split.knn <- sample.split(df.KNN$Cat.Daramad_kol_Mounth, SplitRatio = 0.7)
split.knn

train.knn <- subset(df.KNN, split== TRUE)
#train.copy<- df[split== TRUE, which(names(df)%in%names(train))]

test.knn <- subset(df.KNN, split== FALSE)
train.knn$Daramad_Motefaraghe<-NULL
train.knn$Daramad_Mozd_Month<-NULL
train.knn$Daramad_Mozd_Year<-NULL
train.knn$Daramad_Yarane<- NULL
train.knn$Daramad_Azad<- NULL
test.knn$Daramad_Motefaraghe<-NULL
test.knn$Daramad_Mozd_Month<-NULL
test.knn$Daramad_Mozd_Year<-NULL
test.knn$Daramad_Yarane<- NULL
test.knn$Daramad_Azad<- NULL
drop <- 'Cat.Daramad_kol_Mounth'
library(class)

for (i in 1:20) {
```

```
  knn.pred.test <- knn(train = train.knn[,!(names(train.knn)%in%drop)],
                       test = test.knn[,!(names(test.knn)%in%drop)],
                       cl = train.knn[,names(train.knn)%in%drop],
                       k= i)
  knn.pred.train <- knn(train = train.knn[,!(names(train.knn)%in%drop)],
                        test = train.knn[,!(names(test.knn)%in%drop)],
                        cl = train.knn[,names(train.knn)%in%drop],
                        k= i)
  knn.pred
  kh<- confusionMatrix(knn.pred.test, test.normalized.knn$Cat.Daramad_kol_Mounth)
  kh1<- confusionMatrix(knn.pred.train, train.normalized.knn$Cat.Daramad_kol_Mounth)
  print(i)
  print(kh1[["overall"]][["Accuracy"]])
  print(kh[["overall"]][["Accuracy"]])
}
knn.pred.train.15 <- knn(train = train.knn[,!(names(train.knn)%in%drop)],
                         test = train.knn[,!(names(test.knn)%in%drop)],
                         cl = train.knn[,names(train.knn)%in%drop],
                         k= 15)
confusionMatrix(knn.pred.train.15, train.normalized.knn$Cat.Daramad_kol_Mounth)

knn.pred.test.15 <- knn(train = train.knn[,!(names(train.knn)%in%drop)],
                        test = test.knn[,!(names(test.knn)%in%drop)],
                        cl = train.knn[,names(train.knn)%in%drop],
                        k= 15)

confusionMatrix(knn.pred.test.15, test.knn$Cat.Daramad_kol_Mounth)

#######Neural
```

```
library(neuralnet)
library(caret)
library(nnet)
library(caTools)

df.neural <- df
df.neural$ab.l<- NULL
df.neural$tedad.a<- df1$tedad.a
df.neural$t.o<- df1$t.o
str(df.neural)

df.neural$Hazine_Behdashti<- log10(df.neural$Hazine_Behdashti)
df.neural$Hazine_Ertebatat<- log10(df.neural$Hazine_Ertebatat)
df.neural$Hazine_Ghazayeamade<- log10(df.neural$Hazine_Ghazayeamade)
df.neural$Hazine_Hamlonaghl<- log10(df.neural$Hazine_Hamlonaghl)
df.neural$Hazine_kalavakhadamat<- log10(df.neural$Hazine_kalavakhadamat)
df.neural$Hazine_lavazemkhanegi<- log10(df.neural$Hazine_lavazemkhanegi)
df.neural$Rahn<- log10(df.neural$Rahn)
df.neural$Hazine_Maskan<- log10(df.neural$Hazine_Maskan)
df.neural$Hazine_Noshidani<- log10(df.neural$Hazine_Noshidani)
df.neural$Hazine_Pushak<- log10(df.neural$Hazine_Pushak)
df.neural$Hazine_Tafrihat<- log10(df.neural$Hazine_Tafrihat)
df.neural$Hazine_Kol<- log10(df.neural$Hazine_Kol)

df.neural <- df.neural[, c(2, 5, 9, 12, 13, 18, 22, 25, 26, 29, 33, 38, 42, 43, 50, 74, 56, 57, 60, 61, 63, 65,
73)]

for (i in 16:22) {
```

```
  df.neural[,i]<- (df.neural[,i]- min(df.neural[,i]))/(max(df.neural[,i])- min(df.neural[,i]))
}

z1<-data.frame(class.ind(df.neural$C.O))
z2<-data.frame(class.ind(df.neural$Madrak))
z3<- data.frame(class.ind(df.neural$n.t.m))

df.neural <- cbind(df.neural, z1, z2, z3)
df.neural <- df.neural[, -c(1, 3, 5)]
names(df.neural)[c(26, 27, 28, 29, 30, 31, 32)]=c("Madrak.0","Madrak.1.9","Madrak.2","Madrak.3.4",
"Madrak.5","Madrak.6","Madrak.7.8")
df.neural$Jens <- as.numeric(as.character(df.neural$Jens))
df.neural$oto <- as.numeric(as.character(df.neural$oto))
df.neural$zabt <- as.numeric(as.character(df.neural$zabt))
df.neural$video <- as.numeric(as.character(df.neural$video))
df.neural$pc <- as.numeric(as.character(df.neural$pc))
df.neural$yakhchal <- as.numeric(as.character(df.neural$yakhchal))
df.neural$m.lebas <- as.numeric(as.character(df.neural$m.lebas))
df.neural$m.zarf <- as.numeric(as.character(df.neural$m.zarf))
df.neural$tel <- as.numeric(as.character(df.neural$tel))
df.neural$internet <- as.numeric(as.character(df.neural$internet))
df.neural$cooler.g.s <- as.numeric(as.character(df.neural$cooler.g.s))
df.neural$tedad.a <- as.numeric(as.character(df.neural$tedad.a))

set.seed(2564)
split.neural <- sample.split(df.neural, SplitRatio = 0.7)
train.neural<- subset(df.neural, split.neural==TRUE)
test.neural<- subset(df.neural, split.neural== FALSE)
str(train.neural)
```

```
library(doSNOW)
c1<- makeCluster(7, type = 'SOCK')
registerDoSNOW(c1)

nn<- neuralnet(as.factor(Cat.Daramad_kol_Mounth)~Jens+ tedad.a+ oto+ zabt+ video+ pc+ yakhchal+
m.lebas+ m.zarf+ tel+ internet+ cooler.g.s+
        Quantile_Sen+ Hazine_Ertebatat+ Hazine_Ghazayeamade+ Hazine_Khorakivadokhani+
        Hazine_lavazemkhanegi+ Hazine_Maskan+ Hazine_Pushak+ Alborz+ Hamedan+ Markazi+
Qazvin+
        Qom+ Madrak.0+ Madrak.1.9+ Madrak.2+ Madrak.3.4+ Madrak.5+ Madrak.6+ Madrak.7.8+
Ejari+
        Khedmat.Melki.Ayan+ Melki.Arse.Ayan+ Rahn+ Sayer.Raygan, data= train.neural, linear.output =
F,
        hidden = 1, threshold = 0.1)

stopCluster(c1)
nn$weights
prediction(nn)
plot(nn, rep="best")
train.p=compute(nn,train.neural)

train.c=apply(train.p$net.result,1,which.max)-1

confusionMatrix(as.factor(train.c),as.factor(train.neural$Cat.Daramad_kol_Mounth))



test.p=compute(nn,test.neural)
test.c=apply(test.p$net.result,1,which.max)-1
confusionMatrix(as.factor(test.c),as.factor(test.neural$Cat.Daramad_kol_Mounth))
```

```
#TREE
library(caTools)
library(caret)
df.tree<- df
df.tree$m.o.b <- NULL
df.tree$ab.l <- NULL
df.tree$bargh<- NULL
df.tree$Tahsil.Mikonad <- NULL
df.tree$Savad <- NULL
df.tree$tv.s <- NULL
df.tree$tv.r <- NULL
df.tree$gaz <- NULL
df.tree$cooler.a <- NULL
df.tree$cooler.g <- NULL
df.tree$gaz.l <- NULL
df.tree$hamam <- NULL
df.tree$ashpazkhane <- NULL
df.tree$broodat.m <- NULL
df.tree$hararat.m <- NULL
df.tree$sookht.p <- NULL
df.tree$sookht.g <- NULL
df.tree$sookht.ab <- NULL
df.tree$Sen <- NULL
df.tree$Rahn <- NULL
df.tree$Address <- NULL
df.tree$MahMorajeh <- NULL
df.tree$Fasl <- NULL
df.tree$Daramad_Motefaraghe<-NULL
```

```
df.tree$Daramad_Mozd_Month<-NULL

df.tree$Daramad_Mozd_Year<-NULL

df.tree$Daramad_Yarane<- NULL

df.tree$Daramad_Azad<- NULL

df.tree$Daramad_Motefaraghe<-NULL

df.tree$Daramad_Mozd_Month<-NULL

df.tree$Daramad_Mozd_Year<-NULL

df.tree$Daramad_Yarane<- NULL

df.tree$Daramad_Azad<- NULL

df.tree$Daramad_kol_Mounth<- NULL


library(rpart)

library(rpart.plot)

set.seed(2564) # partition

split.tree <- sample.split(df.tree$Cat.Daramad_kol_Mounth, SplitRatio = 0.7)

split.tree


train.tree <- subset(df.tree, split== TRUE)

#train.copy<- df[split== TRUE, which(names(df)%in%names(train))]


test.tree <- subset(df.tree, split== FALSE)


#plotting a tree with just one variable for clustering

class.tree <- rpart(Cat.Daramad_kol_Mounth ~ ., data = train.tree, control = rpart.control(maxdepth = 2),
method = "class")

prp(class.tree, type = 1, extra = 1, split.font = 1, varlen = -10)

#compue accuracy for one.variable tree

one.variable.pred.train <- predict(class.tree,train.tree,type = "class")

# generate confusion matrix for training data
```

```
confusionMatrix(one.variable.pred.train, train.tree$Cat.Daramad_kol_Mounth)


one.variable.pred.test <- predict(class.tree,test.tree,type = "class")
# generate confusion matrix for test data
confusionMatrix(one.variable.pred.test, test.tree$Cat.Daramad_kol_Mounth)



#derakht pish farz
# plot tree
default.ct <- rpart(Cat.Daramad_kol_Mounth ~ ., data = train.tree, method = "class")


prp(default.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)



#derakht amigh
deeper.ct <- rpart(Cat.Daramad_kol_Mounth ~ ., data = train.tree, method = "class", cp = 0, minsplit = 1)
# count number of leaves
length(deeper.ct$frame$var[deeper.ct$frame$var == "<leaf>"])
# plot tree
prp(deeper.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,
box.col=ifelse(deeper.ct$frame$var == "<leaf>", 'gray', 'white'))



#compue accuracy for default tree
default.ct.point.pred.train <- predict(default.ct,train.tree,type = "class")
# generate confusion matrix for training data
confusionMatrix(default.ct.point.pred.train, train.tree$Cat.Daramad_kol_Mounth)
```

```
# repeat the code for the validation set
default.ct.point.pred.test <- predict(default.ct,test.tree,type = "class")

# generate confusion matrix for test data
confusionMatrix(default.ct.point.pred.test, test.tree$Cat.Daramad_kol_Mounth)



#compue accuracy for deep tree

deeper.ct.point.pred.train <- predict(deeper.ct,train.tree,type = "class")
# generate confusion matrix for training data
confusionMatrix(deeper.ct.point.pred.train, train.tree$Cat.Daramad_kol_Mounth)
# repeat the code for the validation set
deeper.ct.point.pred.test <- predict(deeper.ct,test.tree,type = "class")
# generate confusion matrix for test data
confusionMatrix(deeper.ct.point.pred.test, test.tree$Cat.Daramad_kol_Mounth)



##Etebar sanji moteqate va prune kardan
# argument xval refers to the number of folds to use in rpart's built-in
# cross-validation procedure
# argument cp sets the smallest value for the complexity parameter.
cv.ct <- rpart(Cat.Daramad_kol_Mounth ~ ., data = train.tree, cp = 0.00001, minsplit = 7, xval = 3)
# use printcp() to print the table.
printcp(cv.ct)

#prune by lower cp
pruned.ct <- prune(cv.ct, cp = cv.ct$cptable[which.min(cv.ct$cptable[,"xerror"]),"CP"])
```

```
length(pruned.ct$frame$var[pruned.ct$frame$var == "<leaf>"])
prp(pruned.ct, type = 1, extra = 1, split.font = 1, varlen = -10, digits=-3)



pruned.ct.point.pred.train <- predict(pruned.ct,train.tree,type = "class")
# generate confusion matrix for training data
confusionMatrix(pruned.ct.point.pred.train, train.tree$Cat.Daramad_kol_Mounth)
# repeat the code for the validation set
pruned.ct.point.pred.test <- predict(pruned.ct,test.tree,type = "class")
# generate confusion matrix for test data
confusionMatrix(pruned.ct.point.pred.test, test.tree$Cat.Daramad_kol_Mounth)
##Implimention

df.piadesazi <- read.csv('implimantion - Copy.csv', header = TRUE)
df.piadesazi<- df.piadesazi[1:14, 1:9]
t.o.scale.piadesazi <- scale(df.piadesazi$t.o)
s.z.scale.piadesazi <- scale(df.piadesazi$s.z)
ss<- pca$rotation[1,1]* t.o.scale.piadesazi+ pca$rotation[2,1]* s.z.scale.piadesazi
df.piadesazi$c_t.oands.z<- ss[,1]

nrow(df.piadesazi)

for (i in 1:nrow(df.piadesazi)) {
  if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 8399945& df.piadesazi[i,'Hazine_Ertebatat']<= 545000)
{
    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0
  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 5545500& df.piadesazi[i,'Hazine_Ertebatat']>
545000& df.piadesazi[i,'m.zarf']== 0) {
```

```
    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 5545500& df.piadesazi[i,'Hazine_Ertebatat']>
545000& df.piadesazi[i,'m.zarf']== 1) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 8399945& df.piadesazi[i,'Hazine_Ertebatat']>
545000& df.piadesazi[i,'Hazine_Khorakivadokhani']> 5545500& df.piadesazi[i,'c_t.oands.z']>=.587) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']<= 8399945& df.piadesazi[i,'Hazine_Ertebatat']>
545000& df.piadesazi[i,'Hazine_Khorakivadokhani']> 5545500& df.piadesazi[i,'c_t.oands.z']<.587) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i,
'Hazine_Maskan']>9805000) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i,
'Hazine_Maskan']<=9805000& df.piadesazi[i, 'Hazine_Ertebatat']>= 402500& df.piadesazi[i,
'Quantile_Sen']> 44) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i,
'Hazine_Maskan']<=9805000& df.piadesazi[i, 'Hazine_Ertebatat']>= 402500& df.piadesazi[i,
'Quantile_Sen']<= 44) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i,
'Hazine_Maskan']<=9805000& df.piadesazi[i, 'Hazine_Ertebatat']< 402500& df.piadesazi[i, 'Madrak']==
6) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 1

  }else if (df.piadesazi[i,'Hazine_Khorakivadokhani']> 8399945& df.piadesazi[i,
'Hazine_Maskan']<=9805000& df.piadesazi[i, 'Hazine_Ertebatat']< 402500& df.piadesazi[i, 'Madrak']!=
6) {

    df.piadesazi[i,'pred.Cat.Daramad_kol_Mounth'] = 0

  }

}
```

```
confusionMatrix(as.factor(df.piadesazi$Cat.Daramad_kol_Mounth),as.factor(df.piadesazi$pred.Cat.Daramad_kol_Mounth))
```