

Odločitveni sistemi - 1. seminarska

Katerega ponudnika za internetne, televizijske in telefonske storitve izbrati

Uvod

Za dotični problem sem se odločil, ker sem ravno to jesen menjal telekomunikacijski paket. Podatki so (seveda) prilagojeni za moj osebni primer. Pri internetni opciji izbral hitrost po optiki le pri tistih ponudnikih, ki nudijo optiko na moji lokaciji - to pa je le T2, vsi ostali (Telekom in A1 - simobil) imajo le vdsl. Dodati moram še, da je potrebno pri T2 izbrati programsko shemo (pri nekaterih paketih), saj je drugače nimaš. Tam sem izbral programsko shemo L (osebna preferenca).

Podatki

Tabela telekomunikacijskih paketov

Paket	Cena (prvo leto, mesečna)	Cena (redna, mesečna)	Cena vezava dve leti	Internet (dol)	Internet (gor)	TV programi	TV Zamik	Mobitel(enote/min)	Mobit
Oranžni Diamant HBO	39.99	64.99	1259.76	300	50	230	Da	Inf	
Oranžni Optimum	19.99	49.99	839.76	100	10	175	Ne	1000	
Oranžni Diamant	39.99	64.99	1259.76	300	50	230	Da	Inf	
Top Trio Start	19.95	35.95	670.8	10	2	120	Da	0	
Top Trio A	19.95	39.95	718.8	10	2	145	Da	0	
Top Trio B	19.95	49.95	838.8	20	2	180	Da	0	
Top Trio C	19.95	58.95	946.8	30	5	255	Da	0	
T4 tera	47.23	58.98	1274.52	300	50	175	Da	100	
T4 giga	41.98	51.98	1127.52	200	20	175	Ne	50	
T4 mega	38.98	47.98	1043.52	100	10	175	Ne	0	

T3 tera	44.98	55.98	1211.52	200	20	175	Da	50
T3 giga	39.73	48.98	1064.52	20	10	175	Ne	0
T3 mega	37.48	45.98	1001.52	10	20	175	Ne	0
T2 tvnet	38.98	47.98	1043.52	50	10	175	Ne	0
A1 kombo S	18.99	32.99	623.76	20	1	95	Ne	0
A1 kombo M	18.99	43.99	755.76	30	5	160	Ne	0
A1 kombo L	18.99	52.99	863.76	40	10	210	Da	0

Hierarhično drevo

Prikaz kategorij v hierarhični strukturi.



Komentar drevesa Posamezne kategorije sem izpeljal v smiselne nadkategorije.

- Redna cena in cena dveh let je izpeljana v ceno.
- Internetne storitve so združene v Internet
- TV paketi so združeni v TV pakete, ti pa skupaj s časovnim zamikom in številom programov v televizijske storitve
- Minute, prenos podatkov in sms/mms so združeni v prenosno telefonijo, ta pa skupaj s stacionarno telefonijo tvori telefonijo
- Zgornje tri kategorije skupaj sestavljajo kategorijo telekomunikacije, ki skupaj s ceno tvori celotni paket, ki ga ponudnik internetnih storitev ponuja

```
In [32]: # Vključi ustrezna zaglavja
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

Nastavimo nekaj grafičnih preferenc

```
In [64]: # Uporabimo stil za risanje :)
plt.style.use('fivethirtyeight')
```

```
In [34]: # file locations ~/Desktop/OS/seminarska1/paketi.csv
# Preberi podatke v podatkovni okvir

df = pd.read_csv("~/Desktop/OS/seminarska1/paketi.csv", encoding="utf-8")
```

Pregled podatkov

Tole je prvih pet vrstic podatkovnega okvirja

```
In [35]: df.head(5)
```

Out[35]:

	Paket	Cena (prvo leto mesečna)	Cena (redna mesečna)	Cena vezava dve leti	Internet (dol)	Internet (gor)	TV programi	TV Zamik	Mobitel(enote/min)	Mobitel(enote/sms)
0	Oranžni Diamant HBO	39.99	64.99	1259.76	300	50	230	Da	inf	inf
1	Oranžni Optimum	19.99	49.99	839.76	100	10	175	Ne	1000.000000	1000.000000
2	Oranžni Diamant	39.99	64.99	1259.76	300	50	230	Da	inf	inf
3	Top Trio Start	19.95	35.95	670.80	10	2	120	Da	0.000000	0.000000
4	Top Trio A	19.95	39.95	718.80	10	2	145	Da	0.000000	0.000000

Tole so kategorije telekomunikacijskih paketov:

```
In [36]: df.columns
```

Out[36]: Index(['Paket', 'Cena (prvo leto mesečna)', 'Cena (redna mesečna)', 'Cena vezava dve leti', 'Internet (dol)', 'Internet (gor)', 'TV programi', 'TV Zamik', 'Mobitel(enote/min)', 'Mobitel(enote/prenos)', 'Mobitel(enote/sms)', 'Telefonija (enote/min)', 'HBO (vključen)', 'HD programi', 'Oblak(GB)', 'Poljuben Dodaten Programski Paket'], dtype='object')

Tole so variante oziroma telekomunikacijski paketi, ki jih ponujajo ponudniki:

```
In [37]: df[df.columns[0]]
```

Out[37]:

0	Oranžni Diamant HBO
1	Oranžni Optimum
2	Oranžni Diamant
3	Top Trio Start
4	Top Trio A
5	Top Trio B
6	Top Trio C
7	T4 tera
8	T4 giga
9	T4 mega
10	T3 tera
11	T3 giga
12	T3 mega

13 T2 tvnet
14 A1 kombo S
15 A1 kombo M
16 A1 kombo L
Name: Paket, dtype: object

Uredimo podatkovni okvir.

Vrsticam dam imena prvega stolpca in odstranim prvi stolpec (imena paketov). Odstranim ceno naročnine za prvo leto, saj je skupaj z naročnino drugega leta všteta v ceno paketa za dve leti. Naročnino drugega leta pustim, saj je to cena, ki jo bomo nato redno plačevali, če se ne odločimo menjati paketa oziroma lahko bi jo tudi odstranil, saj obstaja možnost, da bodo čez dve leti boljši paketi in bomo ponovno vezali za novo leto nek nov paket. Jaz sem se za ta postopek odločil, da jo pustim v podatkih (redno ceno).

```
In [38]: df.index = df['Paket']  
# spustimo prva dva stolpca - ime paketa in prvo letno naročnino  
df = df.iloc[:, 2:]  
df.index
```

```
Out[38]: Index(['Oranžni Diamant HBO', 'Oranžni Optimum', 'Oranžni Diamant',  
            'Top Trio Start', 'Top Trio A ', 'Top Trio B', 'Top Trio C', 'T4 tera',  
            'T4 giga', 'T4 mega', 'T3 tera', 'T3 giga', 'T3 mega', 'T2 tvnet',  
            'A1 kombo S', 'A1 kombo M', 'A1 kombo L'],  
            dtype='object', name='Paket')
```

```
In [39]: # popravi tudi INF inf na neko številko, ki bo gotovo največja  
df = df.replace('inf', 10000)  
df.head(5)
```

Out[39]:

	Cena (redna mesečna)	Cena vezava dve leti	Internet (dol)	Internet (gor)	TV programi	TV Zamik	Mobitel(enote/min)	Mobitel(enote/pr
Paket								
Oranžni Diamant HBO	64.99	1259.76	300	50	230	Da	10000.0	
Oranžni Optimum	49.99	839.76	100	10	175	Ne	1000.0	
Oranžni Diamant	64.99	1259.76	300	50	230	Da	10000.0	
Top Trio Start	35.95	670.80	10	2	120	Da	0.0	
Top Trio A	39.95	718.80	10	2	145	Da	0.0	

Vrednosti v koristnost

Sledijo definicije funkcij, ki so bile uporabljene za pretvorbo vrednosti v koristnosti. Torej, da smo vse kategorije spravili na skupno skalo. Vse zvezne kategorije sem postavil na relativno skalo (minimalna vrednosti dobi vrednost 0, maksimalna 100), diskretne z dvema vrednostima pa na 0 ali 100. Zvezne kategorije so pretvorjene z linearno funkcijo. Pri internetni hitrosti (dol in gor) je ta dvodelna: od minimalne hitrosti do 100Mbit je razpon od 0 do 80 [koristnost], od 100Mbit do 300Mbit pa je od 80 do 100 [koristnost].

```
In [43]: ## Cena (redna mesecna)
def cena_redna(val):
    # max cena je 0
    # min cena je 100
    if val == max(df['Cena (redna mesecna)']):
        return 0
    if val == min(df['Cena (redna mesecna)']):
        return 100
    # lin funkcija
    k = (0-100)/(max(df['Cena (redna mesecna)'] - min(df['Cena (redna mesecna)']))
    # n je konst, ki jo moramo pristeti, da pridemo na pravo linijo
    n = abs(k*max(df['Cena (redna mesecna)'])) # tole bo negativno, moram dati +, in pristeti... da bi bila max v
rednost res 0
    return k * val + n

## Cena vezava dve leti
def cena_vsa(val):
    # max cena je 0
    # min cena je 100
    if val == max(df['Cena vezava dve leti']):
        return 0
    if val == min(df['Cena vezava dve leti']):
        return 100
    # lin funkcija
    k = (0 - 100) / (max(df['Cena vezava dve leti']) - min(df['Cena vezava dve leti']))
    n = abs(k * max(df['Cena vezava dve leti']))
    return k * val + n

## Internet (dol)
def intG(val):
    # max hitrost 100
    # min hitrost 0
    if val == max(df['Internet (dol)']):
        return 100
    if val == min(df['Internet (dol)']):
        return 0

    # od 100 Mbit naprej mi ni več tako pomembno
    if val >= 100:
        # lin funkcija za vrednosti od 100 naprej
        # y2 - y1 / x2 - x1
        k1 = (100 - 80) / (max(df['Internet (dol)']) - 100)
        # +70, ker je k = 1/10 in 100 = 1/10 * 300 + n ; n = 100 - 30 = 70
        return k1 * val + 70
    else:
        # prvi del lin funkcija do 100
        k2 = (80 - 0) / (100 - min(df['Internet (dol)']))
        # min*k + n = 0
```

```

    # n = -min*k
    return k2 * val - (min(df['Internet (dol)']))*k2

## Internet (gor)
def intD(val):
    # max hitrost 100
    # min hitrost 0
    if val == max(df['Internet (gor)']):
        return 100
    if val == min(df['Internet (gor)']):
        return 0

    # od 10 Mbit naprej mi ni več tako pomembno za upload
    if val >= 10:
        # lin funkcija za vrednosti od 100 naprej
        # y2 - y1 / x2 - x1
        k1 = (100 - 80) / (max(df['Internet (gor)']) - 10)
        #n
        n = 100 - k1 * max(df['Internet (gor)'])
        return k1 * val + n
    else:
        #prvi del lin funkcija do 10
        k2 = (80 - 0) / (10 - min(df['Internet (gor)']))
        # min*k + n = 0
        # n = -min*k
        return k2 * val - (min(df['Internet (gor)']))*k2

## TV programi
def tvP(val):
    maks = max(df['TV programi'])
    mini = min(df['TV programi'])
    if val == maks:
        return 100
    if val == mini:
        return 0

    # min * k + n = 0
    # n = -min*k
    k = (100 - 0)/(maks - mini)
    return k*val - mini*k

## TV Zamik
def tvZ(val):
    if val == 'Da':
        return 100
    else:
        return 0

## Mobitel(enote/min)
def mobiMin(val):
    maks = max(df['Mobitel(enote/min)'])
    mini = min(df['Mobitel(enote/min)'])
    if val == maks:
        return 100
    if val == mini:
        return 0

```

```
# min * k + n = 0
# n = -min*k
k = (100 - 0) / (maks - mini)
return k * val - mini * k

## Mobitel(enote/prenos)
def mobiP(val):
    maks = max(df['Mobitel(enote/prenos)'])
    mini = min(df['Mobitel(enote/prenos)'])
    if val == maks:
        return 100
    if val == mini:
        return 0

    # min * k + n = 0
    # n = -min*k
    k = (100 - 0) / (maks - mini)
    return k * val - mini * k

## Mobitel(enote/sms)
def mobiSms(val):
    maks = max(df['Mobitel(enote/sms)'])
    mini = min(df['Mobitel(enote/sms)'])
    if val == maks:
        return 100
    if val == mini:
        return 0

    # min * k + n = 0
    # n = -min*k
    k = (100 - 0) / (maks - mini)
    return k * val - mini * k

## Telefonija (enote/min)
def telefon(val):
    maks = max(df['Telefonija (enote/min)'])
    mini = min(df['Telefonija (enote/min)'])
    if val == maks:
        return 100
    if val == mini:
        return 0

    # min * k + n = 0
    # n = -min*k
    k = (100 - 0) / (maks - mini)
    return k * val - mini * k

## HBO (vključen)
def hbo(val):
    if val == 'Da':
        return 100
    else:
        return 0

## HD programi
def hd(val):
    maks = max(df['HD programi'])
```

```
mini = min(df['HD programi'])
if val == maks:
    return 100
if val == mini:
    return 0

# min * k + n = 0
# n = -min*k
k = (100 - 0) / (maks - mini)
return k * val - mini * k

## Oblak(GB)
def oblak(val):
    maks = max(df['Oblak(GB)'])
    mini = min(df['Oblak(GB)'])
    if val == maks:
        return 100
    if val == mini:
        return 0

    # min * k + n = 0
    # n = -min*k
    k = (100 - 0) / (maks - mini)
    return k * val - mini * k

## Poljuben Dodaten Programski Paket
def extra(val):
    if val == 1:
        return 100
    else:
        return 0
```

Sledi še pretvorba vseh kategorij v koristnosti:

```
In [45]: # SPREMENIMO VREDNOSTI KRITERIJEV V KORISTNOST, damo na skupno skalo
funkcije = [cena_redna, cena_vsa, intG, intD, tvP, tvZ, mobiMin, mobiP, mobiSms, telefon, hbo, hd, oblak, extra]
# skopiramo v nov podatkovni okvir
dfKorist = df.copy()
indeks = 0
for stolpec in dfKorist.columns:
    dfKorist[stolpec] = dfKorist[stolpec].apply(funkcije[indeks])
    indeks += 1

dfKorist.head(10)
```

Out [45]:

	Cena (redna mesečna)	Cena vezava dve leti	Internet (dol)	Internet (gor)	TV programi	TV Zamik	Mobitel(enote/min)	Mobitel
Paket								
Oranžni								
Diamant	0.00000	2.268117	100.000000	100.000000	84.375	100		100.0
HBO								

Oranžni Optimum	46.87500	66.808040	80.000000	80.000000	50.000	0	10.0
Oranžni Diamant	0.00000	2.268117	100.000000	100.000000	84.375	100	100.0
Top Trio Start	90.75000	92.771529	0.000000	8.888889	15.625	100	0.0
Top Trio A	78.25000	85.395538	0.000000	8.888889	31.250	100	0.0
Top Trio B	47.00000	66.955560	8.888889	8.888889	53.125	100	0.0
Top Trio C	18.87500	50.359580	17.777778	35.555556	100.000	100	0.0
T4 tera	18.78125	0.000000	100.000000	100.000000	50.000	100	1.0
T4 giga	40.65625	22.588973	90.000000	85.000000	50.000	0	0.5
T4 mega	53.15625	35.496957	80.000000	80.000000	50.000	0	0.0

Manjvredne variante

Izločiti moramo variante, ki jih model ne bi v nikakršni situaciji izbral. Spodaj je funkcija, ki odstrani manjvredne variante Varianta A je manjvredna od B, ko je varianta B vsaj boljša v eni kategoriji v ostalih pa enaka.

```
In [46]: # IZLOČIMO MANJVREDNE VARIANTE
def odstraniManjvredne(df):
    nrow, ncol = df.shape
    manjvredne = set()

    #izberemo prvo vrstico
    for row1 in range(nrow):
        #izberemo drugo vrstico
        for row2 in range(row1+1, nrow):
            #odstevamo od prve izbrane, drugo izbrano in dobimo vektor razlik
            vektorRazlik = df.iloc[row1] - df.iloc[row2]
            #če naslednji pogoj velja sum(vektorRazlik < 0) != 0 and sum(vektorRazlik > 0) != 0
            #pomeni da ima vektor razlik neg in poz vrednosti kar ni OK, če je ena varianta boljša od druge je vektor ali
            #cel neg ali cel poz :)
            #negiramo torej ta izraz, dodati moramo se pogoj da ne smejo biti vsi 0
            if not(sum(vektorRazlik < 0) != 0 and sum(vektorRazlik > 0) != 0) and sum(vektorRazlik == 0):
                #če smo tukaj notri je vektor ali cel poz ali cel neg
                #če je cel poz, pomeni da je row1 boljša od row2, pomeni da dodamo row2
                if sum(vektorRazlik) > 0:
                    #dodaj ime paketa, ki je manj vreden
                    manjvredne.add(df.index[row2])
                else:
                    manjvredne.add(df.index[row1])
            # vrni podatkovni okvir brez manjvrednih in manjvredne - imena
    for varianta in manjvredne:
        df = df.loc[df.index != varianta, :]
```

```
return (df, manjvredne)

dfKorist, tt = odstraniManjvredne(dfKorist)
```

Tole so manjvredne variante:

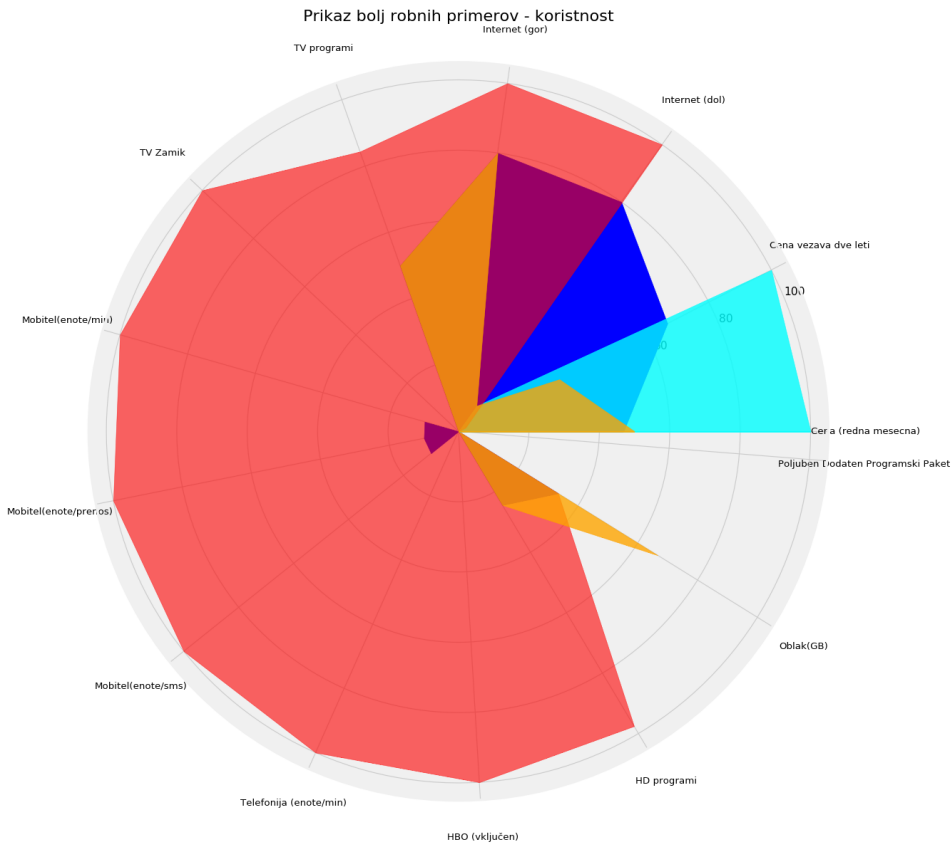
```
In [47]: tt

Out[47]: {'T2 tvnet'}
```

Zvezdnati prikaz

Grafični prikaz nekaterih kriterijev - par sredinskih, nekaj dragih in nekaj najcenejših kategorij.

```
In [76]: theta = np.linspace(0, 2*np.pi-0.08, len(dfKorist.columns))
vrednosti = dfKorist.loc['Oranžni Optimum']
vrednosti1 = dfKorist.loc['A1 kombo S']
vrednosti2 = dfKorist.loc['Oranžni Diamant HBO']
vrednosti3 = dfKorist.loc['T3 giga']
#nastavi velikost figure
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
ax = plt.subplot(111, projection='polar')
ax.fill_between(theta, vrednosti, 0, label='Oranžni Optimum', alpha = 1, color="blue")
ax.fill_between(theta, vrednosti1, 0, label='A1 kombo S', alpha = 0.8, color="cyan")
ax.fill_between(theta, vrednosti2, 0, label='Oranžni Diamant HBO', alpha = 0.6, color="red")
ax.fill_between(theta, vrednosti3, 0, label='T3 giga', alpha = 0.8, color="orange")
ax.set_title("Prikaz bolj robnih primerov - koristnost")
ax.set_xticks(theta)
ax.set_xticklabels(dfKorist.columns, size=12)
ax.legend(bbox_to_anchor=(-0.3, 1.2))
plt.show()
```



UTEŽEVANJE

Sledi postopek uteževanja kategorij. Prvo bom prikazal hierarhično strukturo drevesa, ki sem ji ročno priredil uteži glede na moje osebne preference - torej kar si želim v telekomunikacijskem paketu.

Nastavitve uteži

Uteži sem nastavil tako, da sem se ročno sprehodil po drevesu in vedno nastavil vrednosti od 0 do 1 med vejami, ki vodijo v eno izpeljano kategorijo. Te vrednosti se morajo znotraj ene izpeljane kategorije sešteti v ena. Ta postopek se zaključi, ko pridem do korena. Prave uteži posameznih kategorij pa dobim tako, da zmnožim vse veje, ki vodijo do te kategorije.

```
In [81]: # PRVI NIVO
```

```
#cena
rednaCena = 0.3
cena2leti = 0.7
```

```
#internet
dol = 0.7
gor = 0.2
cloud = 0.1
```

```
#gsm
mobiM = 0.3
```

```
mobiS = 0.3
mobiPaketi = 0.4

#tv paketi
hb = 0.65
hdd = 0.3
extr = 0.05

# DRUGI NIVO

#televizija
programi = 0.3
zamik = 0.4
paketi = 0.3

#telefonija
gsm = 0.9
stacionarni = 0.1

# TRETJI NIVO
inter = 0.7
teve = 0.2
telefonija = 0.1

# CETRTI NIVO
telek = 0.4
c = 0.6

# Uteži so bile nastavljene glede na drevo. Prvo sem nastavil uteži od spodaj gor, nato pa zmnožil po vejah nazaj dol, da sem dobil dejanske vrednosti

imena = ["redna cena", "vezava dve leti", "internet dol hitrost", "internet gor hitrost", "število programov", "časovni zamik", "mobi minute", "mobi paketi", "mobi sms", "stacionarni telefon", "HBO", "HD", "oblak", "dodatni paket"]
utezi = [c*rednaCena, c*cena2leti, telek*inter*dol, telek*inter*gor, telek*teve*programi, telek*teve*zamik, telek*telefonija*gsm*mobiM, telek*telefonija*gsm*mobiPaketi, telek*telefonija*gsm*mobiS, telek*telefonija*stacionarni, telek*teve*paketi*hb, telek*teve*paketi*hdd, telek*inter*cloud, telek*teve*paketi*extr]

print("Uteži", utezi)
print("Vsota uteži mora biti ena:", sum(utezi))
```

Uteži [0.18, 0.42, 0.19599999999999998, 0.055999999999999994, 0.024000000000000004, 0.032000000000000001, 0.010800000000000002, 0.014400000000000005, 0.010800000000000002, 0.004000000000000001, 0.015600000000000003, 0.007200000000000001, 0.027999999999999997, 0.0012000000000000003]

Vsota uteži mora biti ena: 1.0

Uteži po kategorijah:

```
In [82]: for x in range(len(imena)):
          print(imena[x], "utežitev %.5f" % (utezi[x]))
```

redna cena utežitev 0.18000
vezava dve leti utežitev 0.42000
internet dol hitrost utežitev 0.19600
internet gor hitrost utežitev 0.05600

število programov utežitev 0.02400
časovni zamik utežitev 0.03200
mobi minute utežitev 0.01080
mobi paketi utežitev 0.01440
mobi sms utežitev 0.01080
stacionarni telefon utežitev 0.00400
HBO utežitev 0.01560
HD utežitev 0.00720
oblak utežitev 0.02800
dodatni paket utežitev 0.00120

Najbolj sem utežil obe ceni, ki skupaj predstavljata kar 0.6. Sledi internetna hitrost (dol) in (gor). Sledi časovni zamik, število programov in HBO. Take uteži sem postavil zato, ker smo finančno trenutno šibki in je glavni cilj nizka cena paketa. Sledi internet, saj je zaradi meni osebno vse slabše televizijske izkušnje in vse boljše multimedijske ponudbe na spletu, veliko bolj pomembna hitrost (dol) interneta kot pa televizijske storitve. Telefonija mi je najmanj pomembna, saj večinoma že imamo sklenjene pogodbe.

hierarhično drevo - utežene kategorije



Ocena variant

Sedaj potrebujemo še funkcijo za izračun utežene vsote. Sledi postopek ocenitve oziroma točkovanja variant.

Funkcija

```
In [83]: def weight_score(varianta, weight):
        score = 0
        for i in range(len(varianta)):
            score += varianta[i]*weight[i]
        return score

def ocene_variant(dataf, weight):
    tocke = []
    i = 0
    for varianta in dataf.index:
        score = weight_score(dataf.loc[varianta], weight)
        tocke.append((varianta, score))
        i+=1
    return tocke
```

Izračun

```
In [84]: tocke = ocene_variant(dfKorist, utezi)
        # UREDI OD MAX DO MIN
        tocke = sorted(tocke, key=lambda x: x[1], reverse=True)
```

Izpis

```
In [85]: for varianta, t in tocke:
         print("varianta:",varianta, "tocke:",t)

varianta: A1 kombo S tocke: 61.7422222222
varianta: Top Trio Start tocke: 59.8584474632
varianta: Oranžni Optimum tocke: 59.3258198182
varianta: Top Trio A tocke: 54.8855311807
varianta: A1 kombo M tocke: 52.2140296811
varianta: A1 kombo L tocke: 48.6959876932
varianta: T4 mega tocke: 46.945790199
varianta: Top Trio B tocke: 43.8588065991
varianta: T4 giga tocke: 42.4677700379
varianta: T3 tera tocke: 38.9297498769
varianta: Oranžni Diamant HBO tocke: 38.5733816146
varianta: Oranžni Diamant tocke: 37.1333816146
varianta: Top Trio C tocke: 36.826745641
varianta: T4 tera tocke: 35.9962347561
varianta: T3 mega tocke: 35.3814669462
varianta: T3 giga tocke: 32.0235073809
```

Glede na moje uteži vidimo, da prevladajo variante, ki imajo nižjo ceno. Zmagal je namreč A1 kombo S, sledi mu Top Trio Start od Telekoma. Na tretjem mestu je T2 s paketom Oranžni Optimum. Vsi trije so točkovno zelo skupaj.

Za alternativno varianto sem si izbral Oranžni Optimum, saj ima čez vse variante "povprečne" vrednosti. Prikaz:

```
In [88]: dfKorist.loc["A1 kombo S"]

Out[88]: Cena (redna mesecna)          100.000000
Cena vezava dve leti                  100.000000
Internet (dol)                        8.888889
Internet (gor)                        0.000000
TV programi                          0.000000
TV Zamik                             0.000000
Mobitel(enote/min)                   0.000000
Mobitel(enote/prenos)                0.000000
Mobitel(enote/sms)                   0.000000
Telefonija (enote/min)                0.000000
HBO (vključen)                       0.000000
HD programi                          0.000000
Oblak(GB)                            0.000000
Poljuben Dodaten Programski Paket    0.000000
Name: A1 kombo S, dtype: float64
```

```
In [87]: dfKorist.loc["Oranžni Optimum"]

Out[87]: Cena (redna mesecna)          46.875000
Cena vezava dve leti                  66.808040
Internet (dol)                        80.000000
Internet (gor)                        80.000000
TV programi                          50.000000
TV Zamik                             0.000000
```

Mobitel(enote/min)	10.000000
Mobitel(enote/prenos)	10.000000
Mobitel(enote/sms)	10.000000
Telefonija (enote/min)	0.000000
HBO (vključen)	0.000000
HD programi	24.390244
Oblak(GB)	33.333333
Poljuben Dodaten Programski Paket	0.000000
Name: Oranžni Optimum, dtype: float64	

SORT - primerjava variant

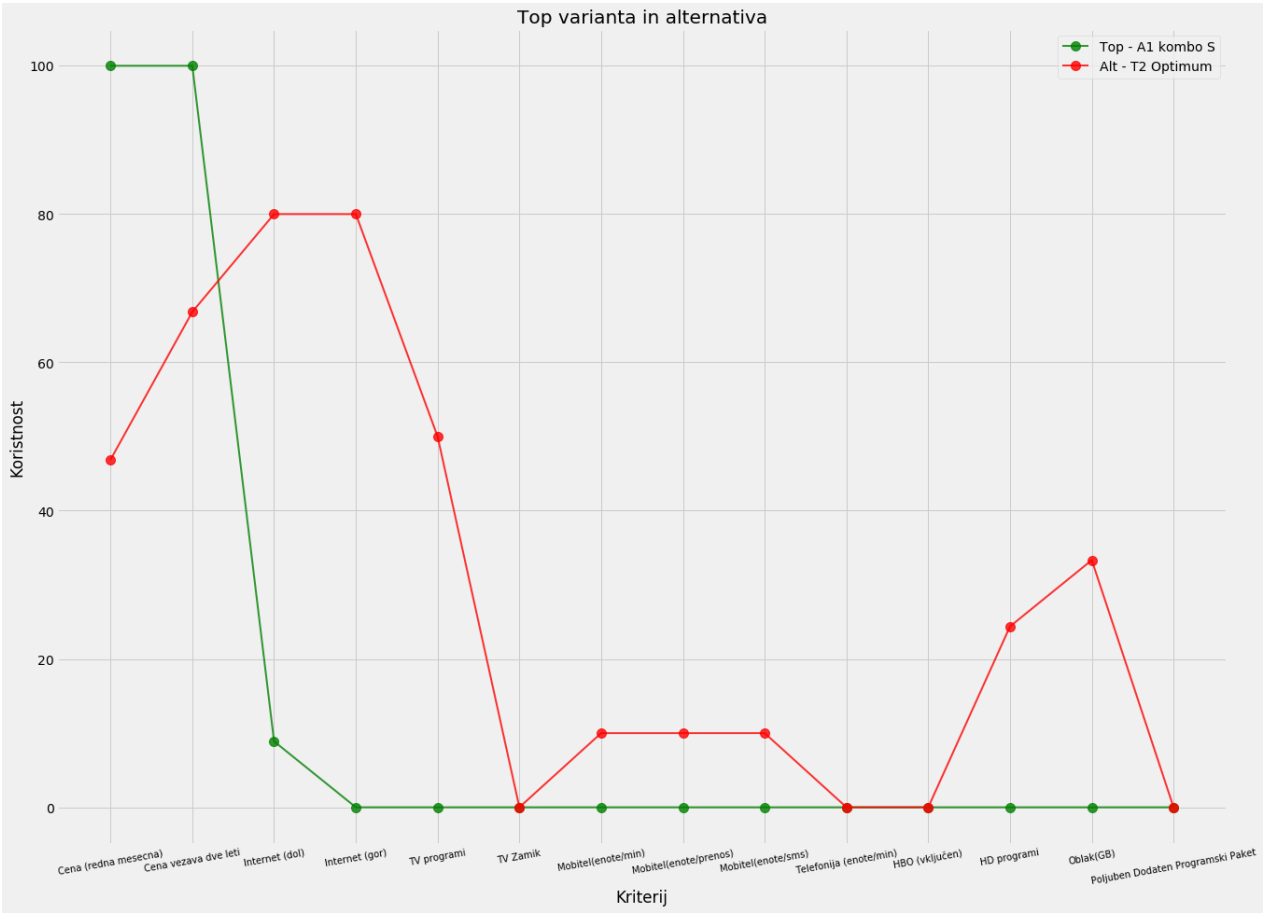
Sledijo primerjave variant z najboljšo in najslabšo varianto.

Grafični prikaz

```
In [91]: top = dfKorist.loc['A1 kombo S']
alt = dfKorist.loc['Oranžni Optimum']

fig, ax = plt.subplots(figsize=(20,15))

x1 = np.arange(len(top))
y1 = top
x2 = np.arange(len(alt))
y2 = alt
ax.plot(x1, y1, "g", alpha = 0.8, label="Top - A1 kombo S", linestyle="-", marker = "o", markersize=10, linewidth=2)
ax.plot(x2, y2, "r", alpha = 0.8, label="Alt - T2 Optimum", linestyle="-", marker = "o", markersize=10, linewidth=2)
ax.set_title("Top varianta in alternativa")
ax.set_xticks(x2)
ax.set_ylabel("Koristnost")
ax.set_xlabel("Kriterij")
ax.set_xticklabels(dfKorist.columns, rotation=10, size=10)
ax.legend(loc=1)
plt.show()
```



Komentar

- Top varianta: vidimo, da ima najboljša varianta zelo ugodno ceno (okoli 600-700 evrov, vezava za dve leti), a ima zelo slabo internetno hitrost, ostalih stvari pa nima ali pa so minimalne (recimo TV programi, le okoli 90).
- Alternativna varianta: nima tako ugodne redne cene (zanimvo je, da ima še kar ugodno ceno za dve leti - razlog je, da ima zelo dobro prvo letno akcijsko ceno) , a ima boljši internet in več televizijskih programov. Ima tudi telefonijo in prostor v oblaku.

Zmagala je seveda prva, ker mi je najbolj pomembna cena in šele nato telekomunikacije. Pri slednji izpeljani kategoriji mi je najbolj pomemben internet.

Primerjaj variante z najboljšo

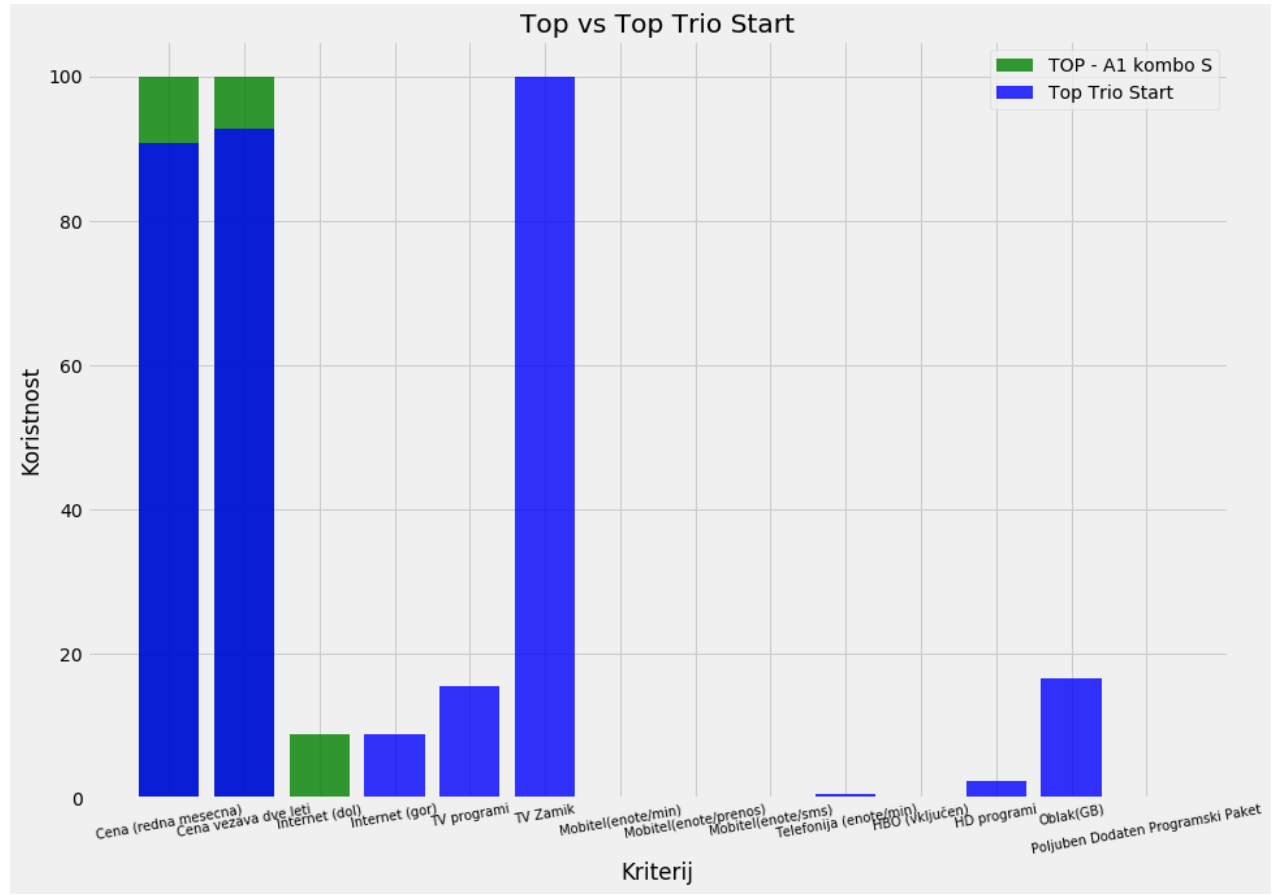
```
In [94]: top = dfKorist.loc['A1 kombo S']
x1 = np.arange(len(top))
y1 = top

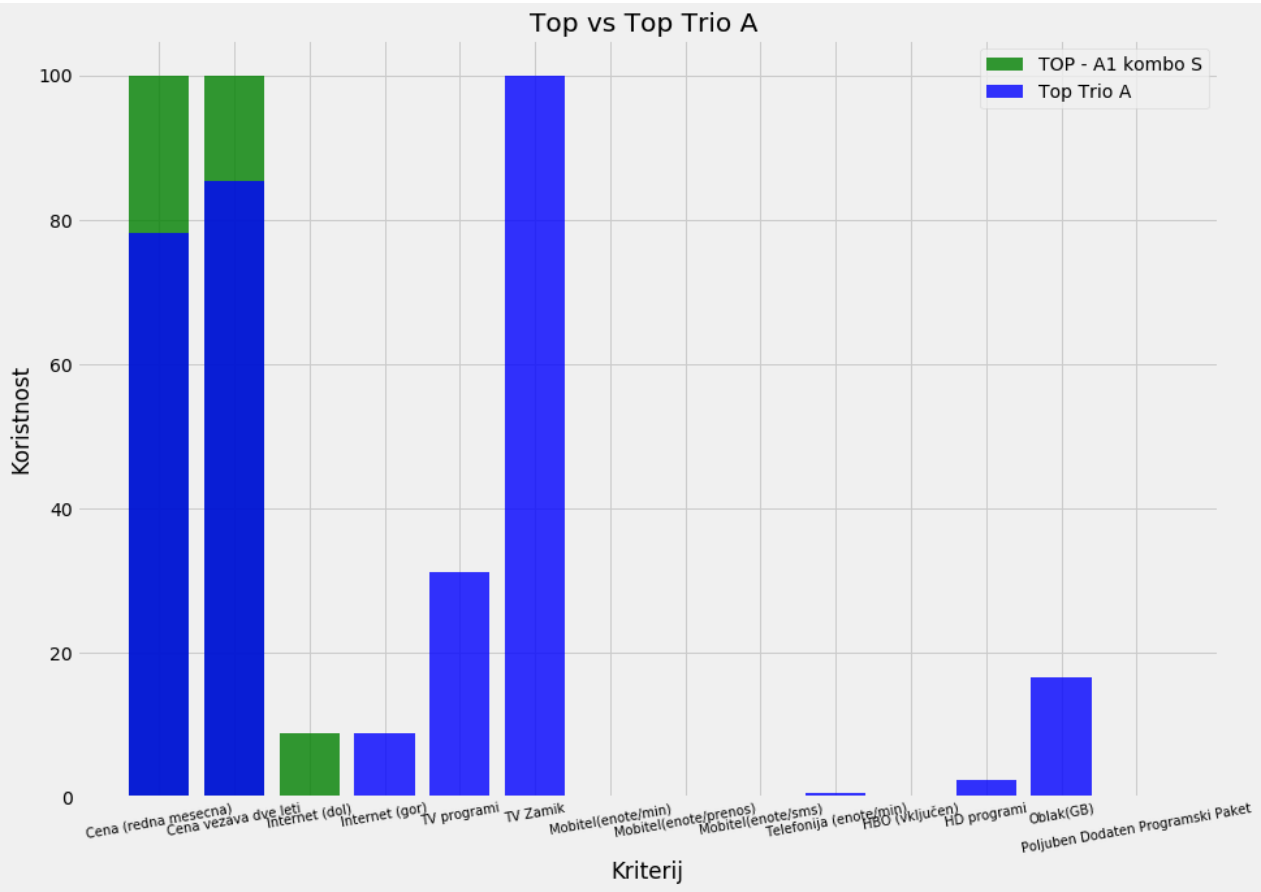
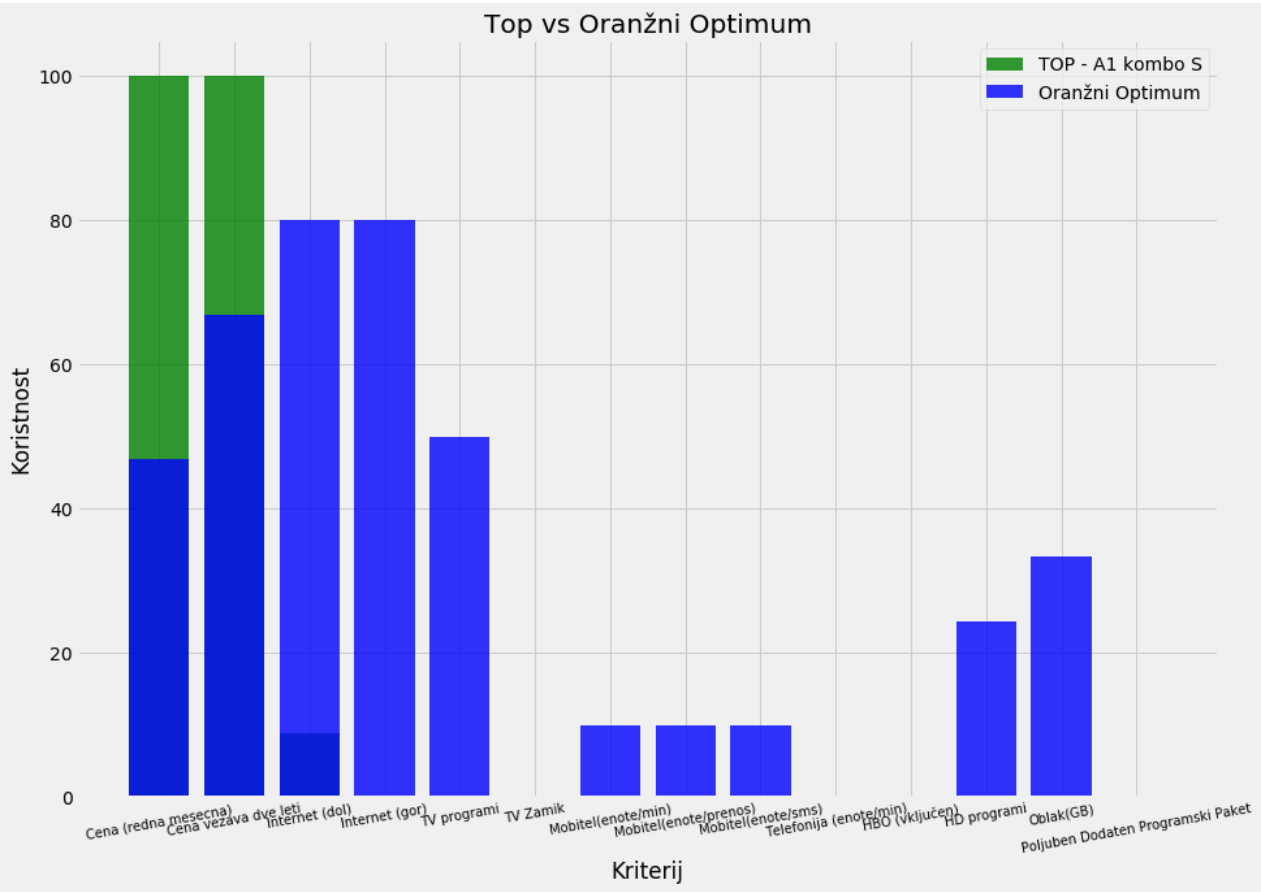
ostale_notTop = [x[0] for x in tocke[1:]]

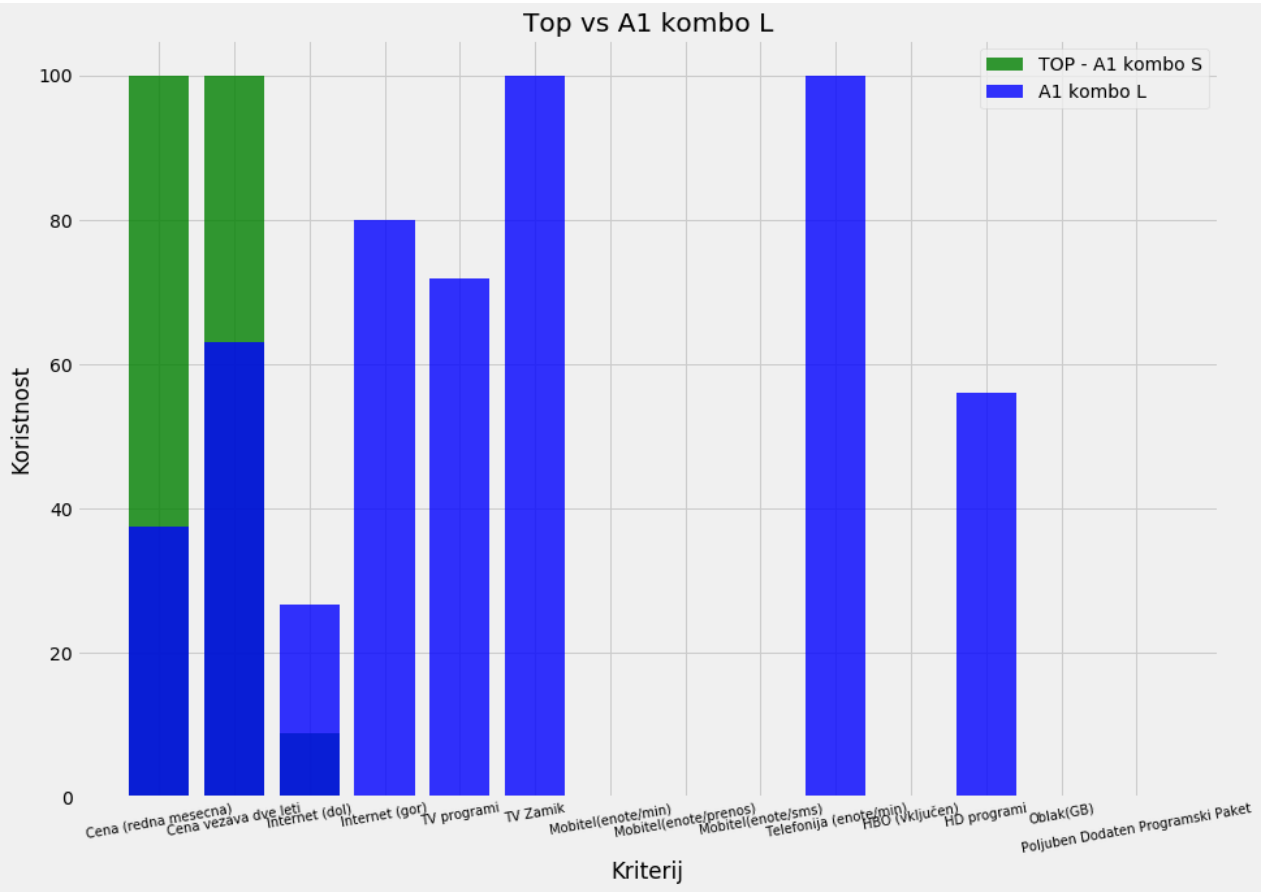
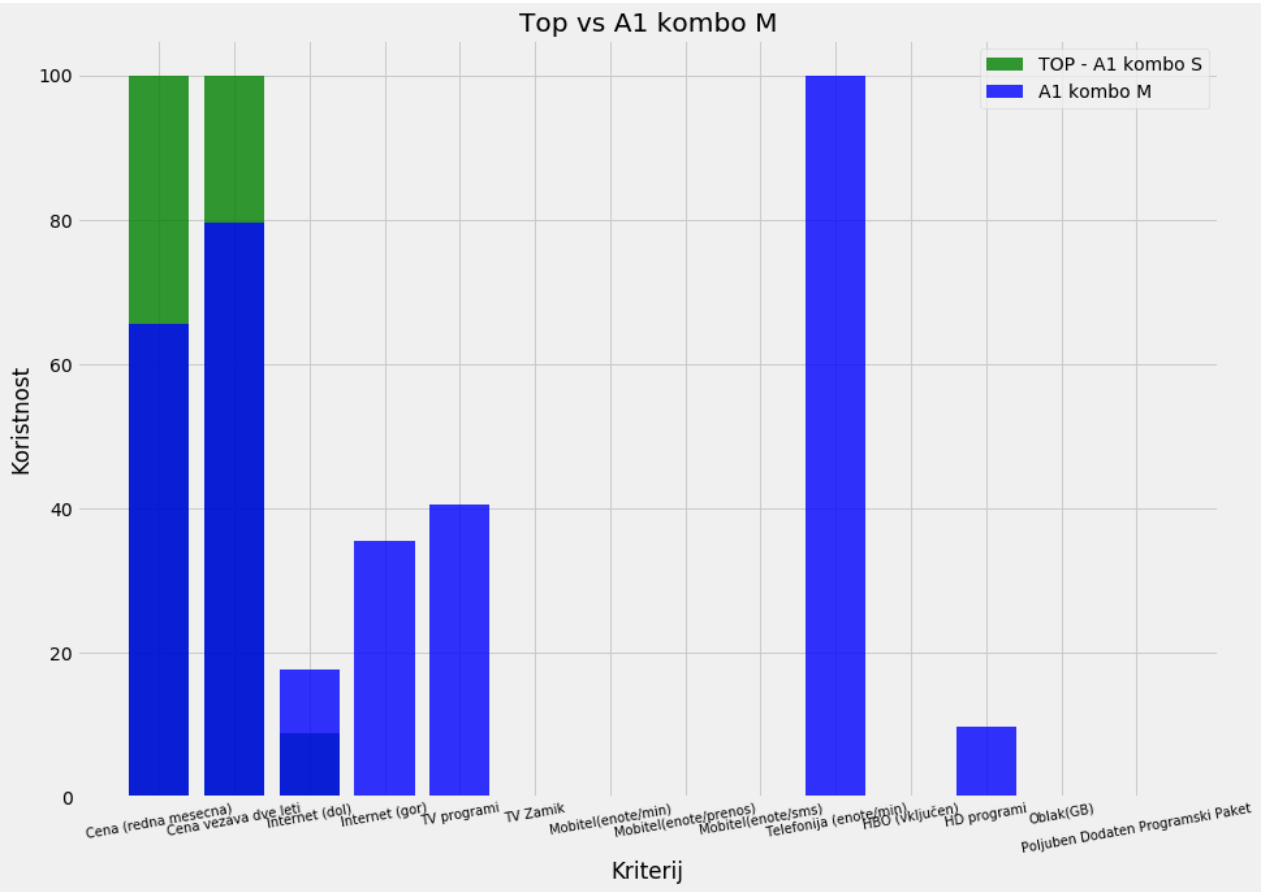
for varianta in ostale_notTop:
    fig, ax = plt.subplots(figsize=(14,10))
    ax.bar(x1, y1, color="green", alpha=0.8, label="TOP - A1 kombo S")
    y = dfKorist.loc[varianta]
    x = np.arange(len(y))
```

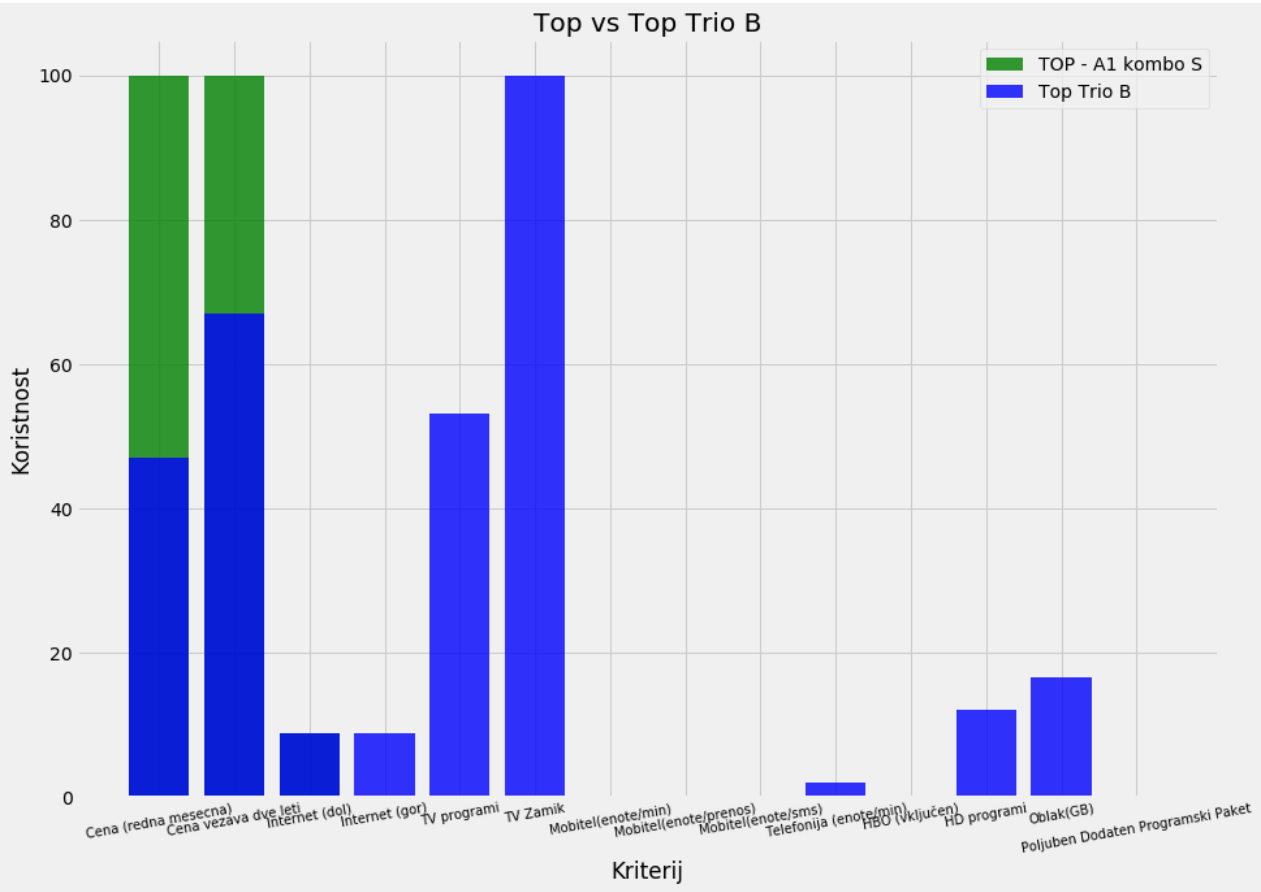
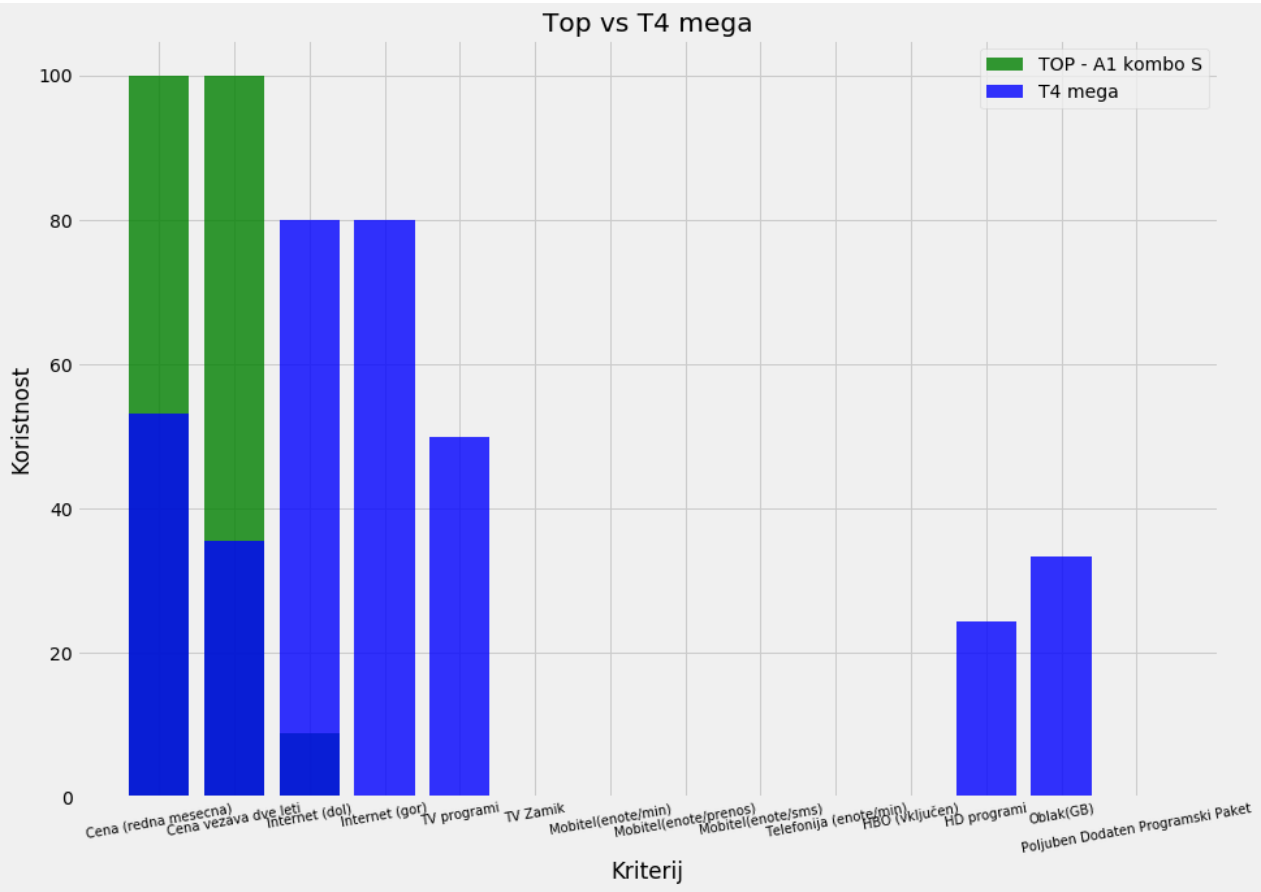


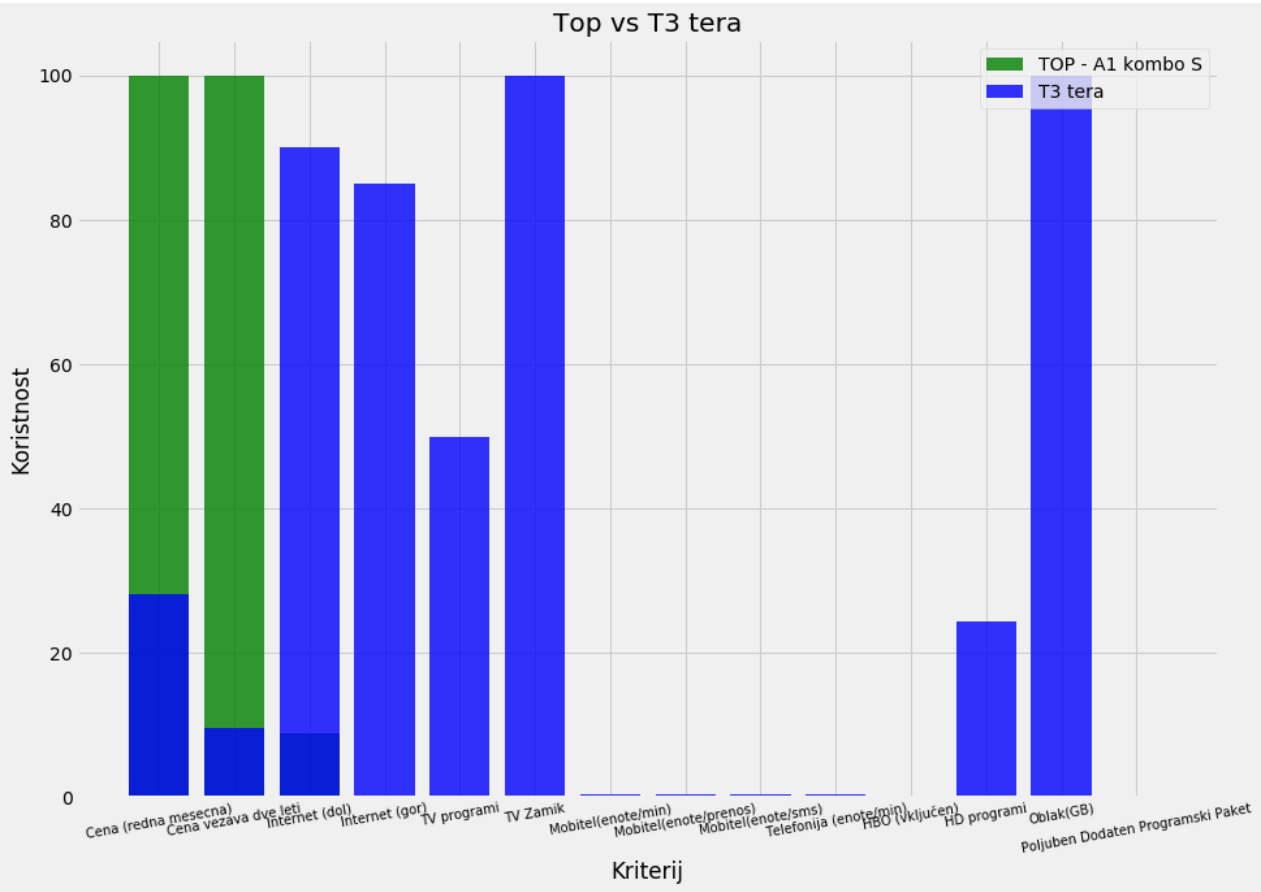
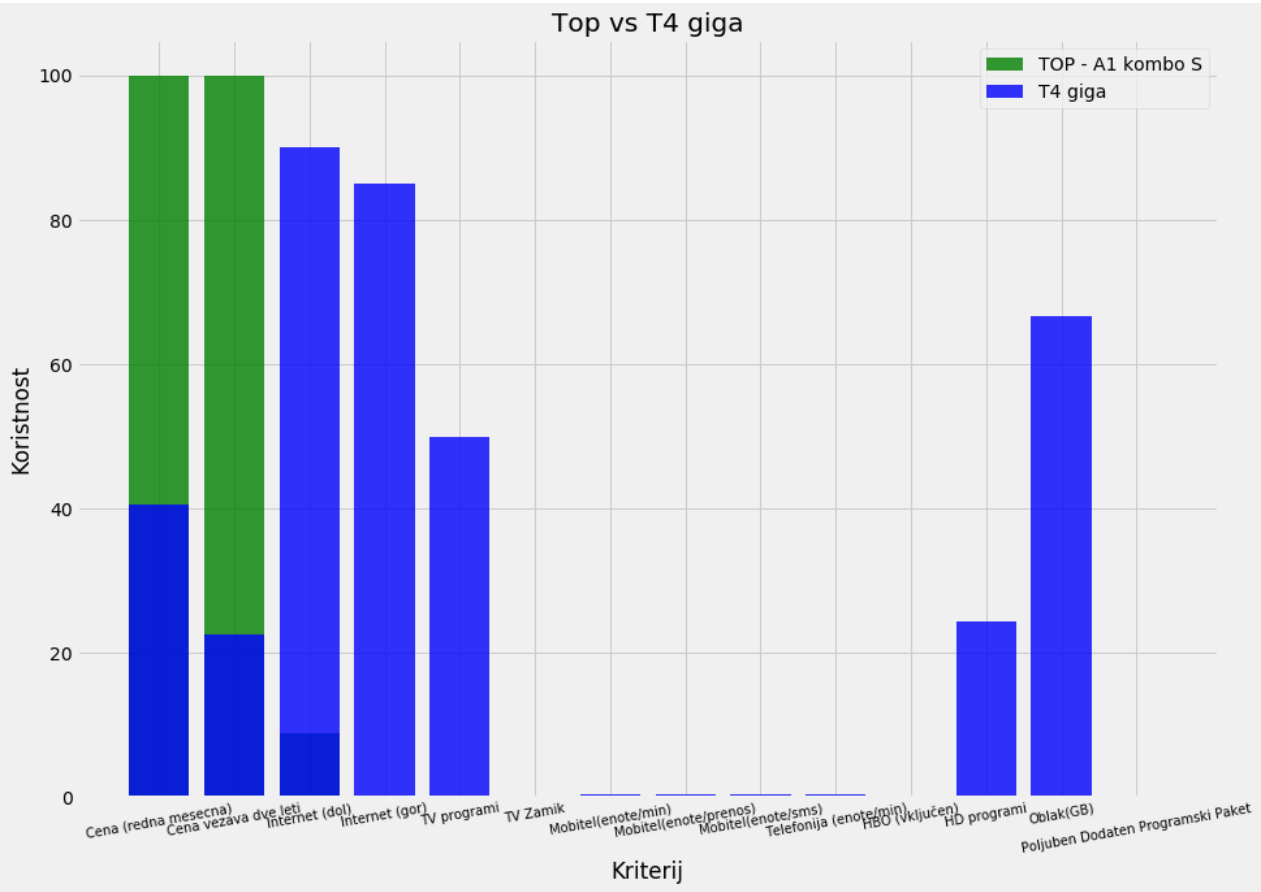
```
#ax.fill_between(x, y, y1, where=y>=y1, alpha=0.8, color=barva)
ax.bar(x, y, color="blue", alpha=0.8, label=varianta)
ax.set_title("Top vs %s" % (varianta))
ax.set_xticks(x)
ax.set_ylabel("Koristnost")
ax.set_xlabel("Kriterij")
ax.set_xticklabels(dfKorist.columns, rotation=10, size=10)
ax.legend(loc=1)
plt.show()
```

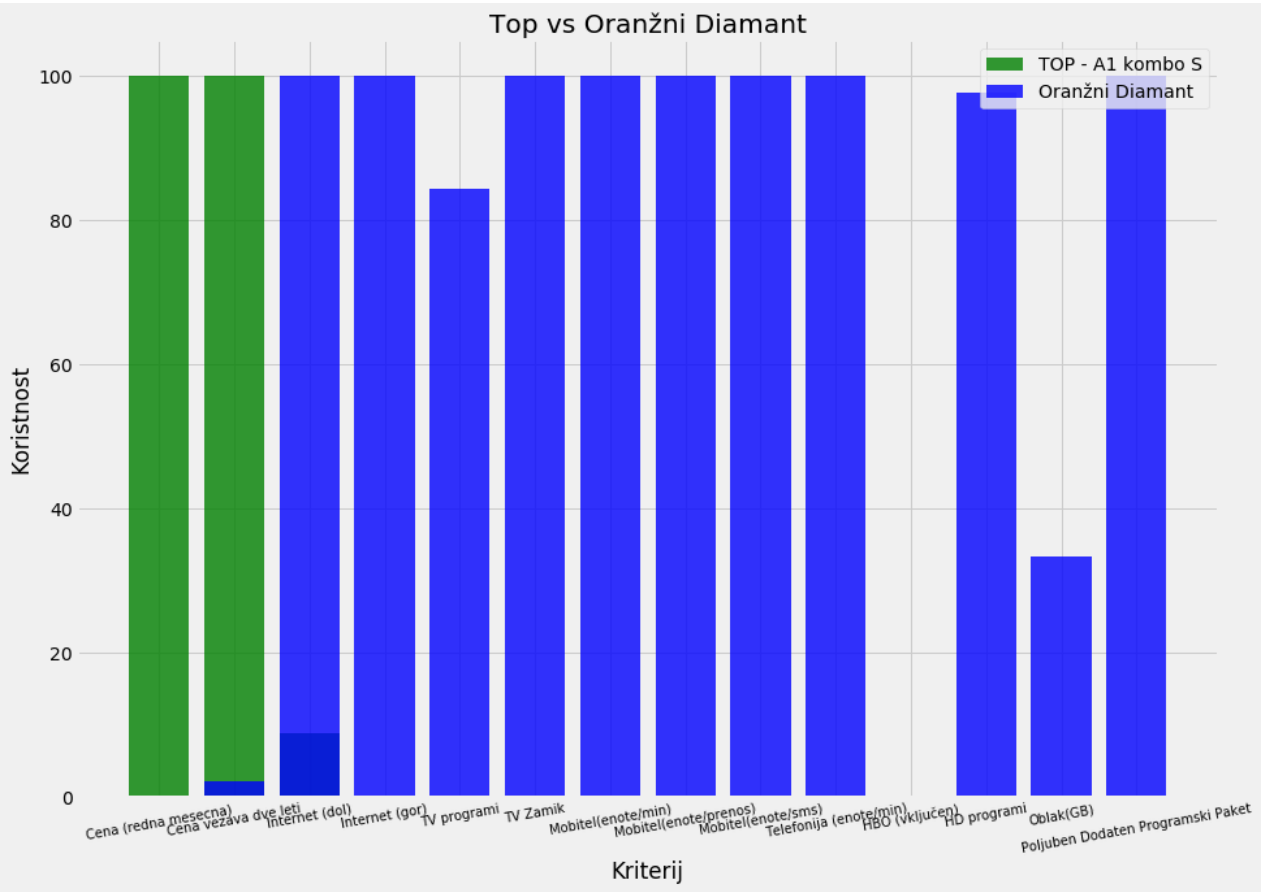
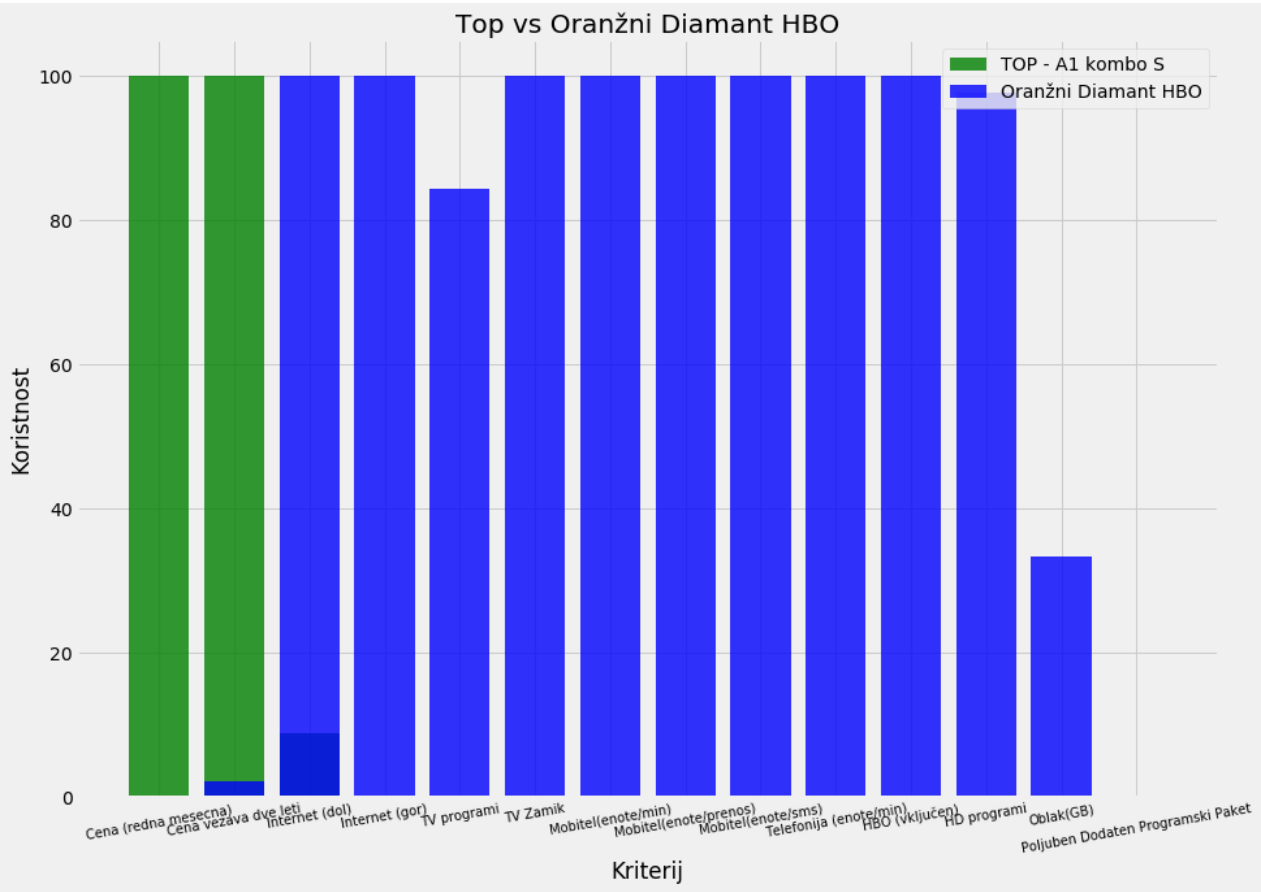


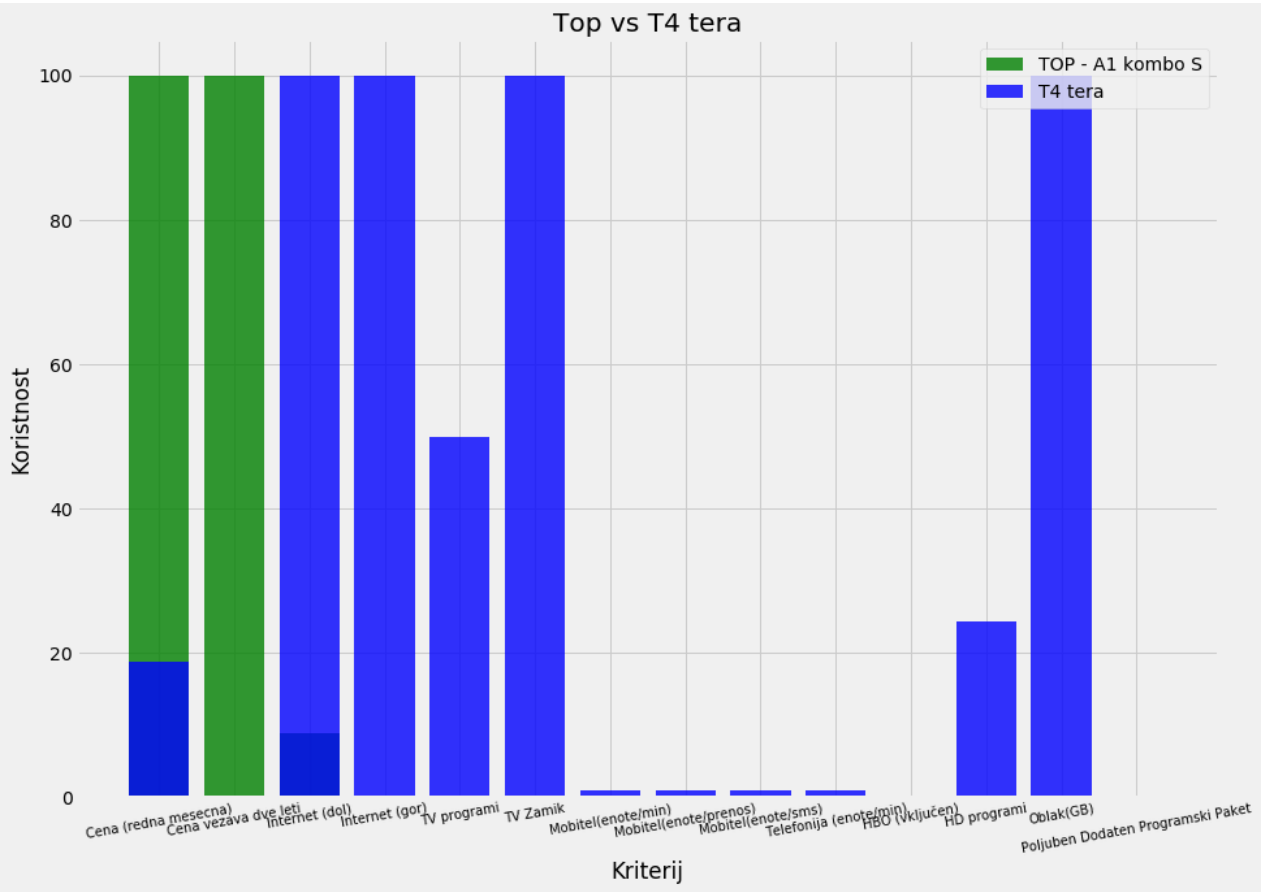
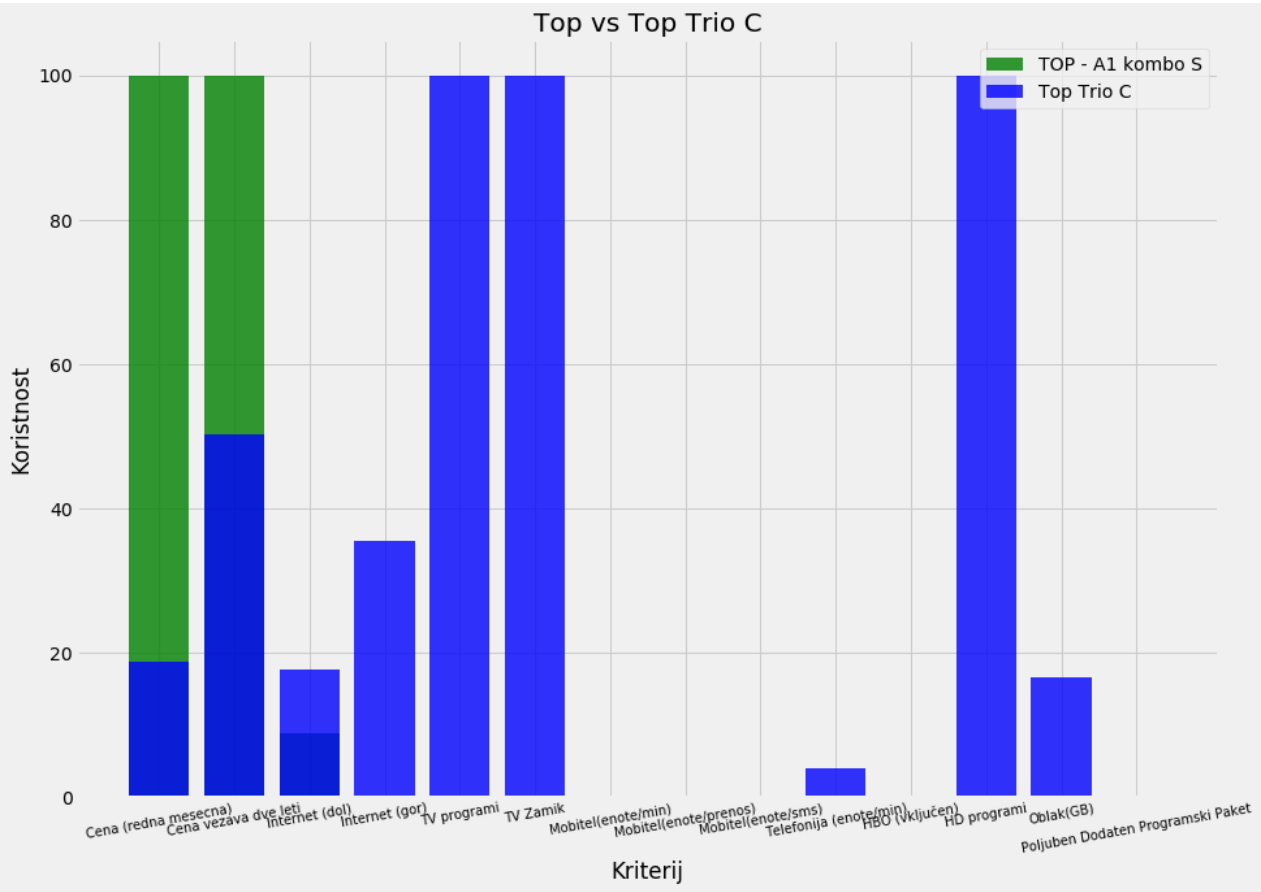


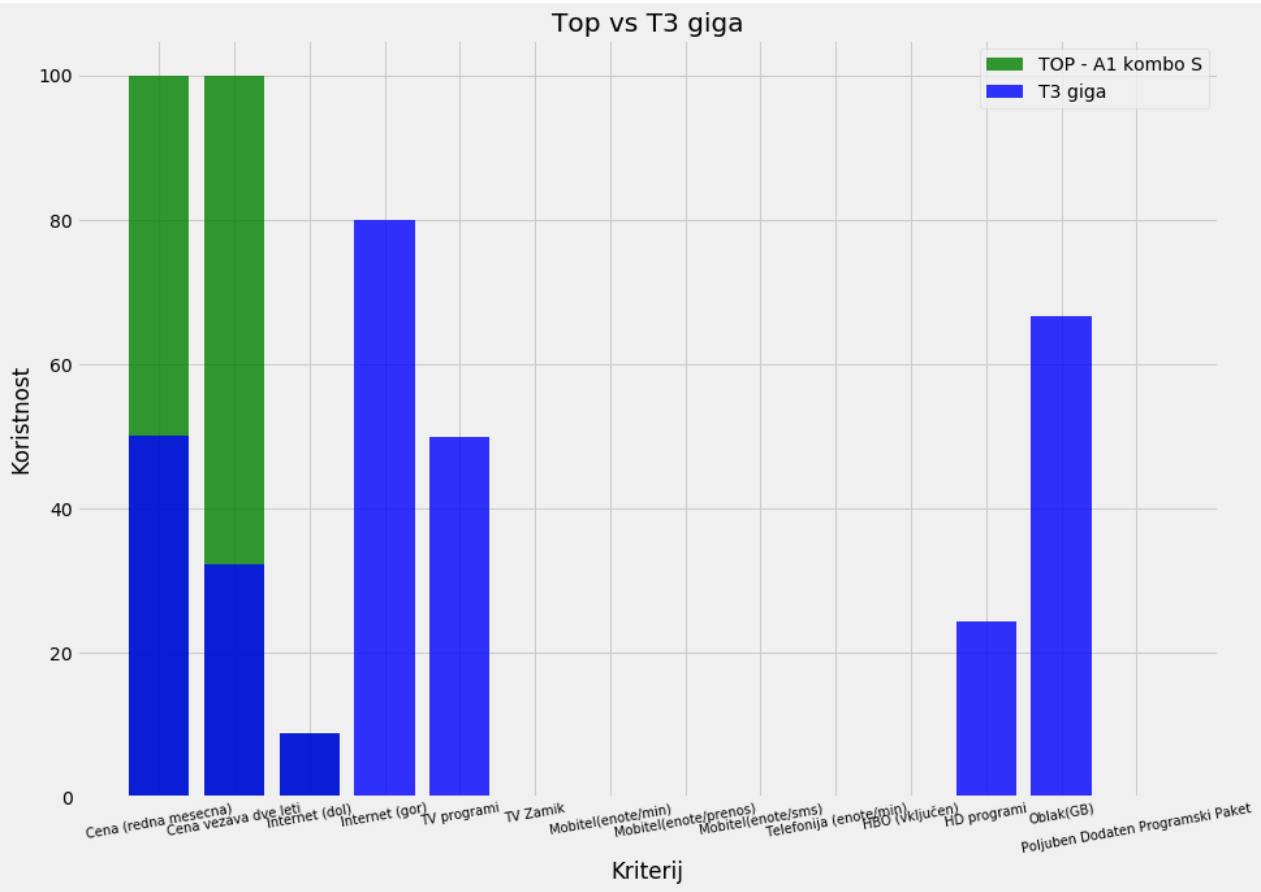
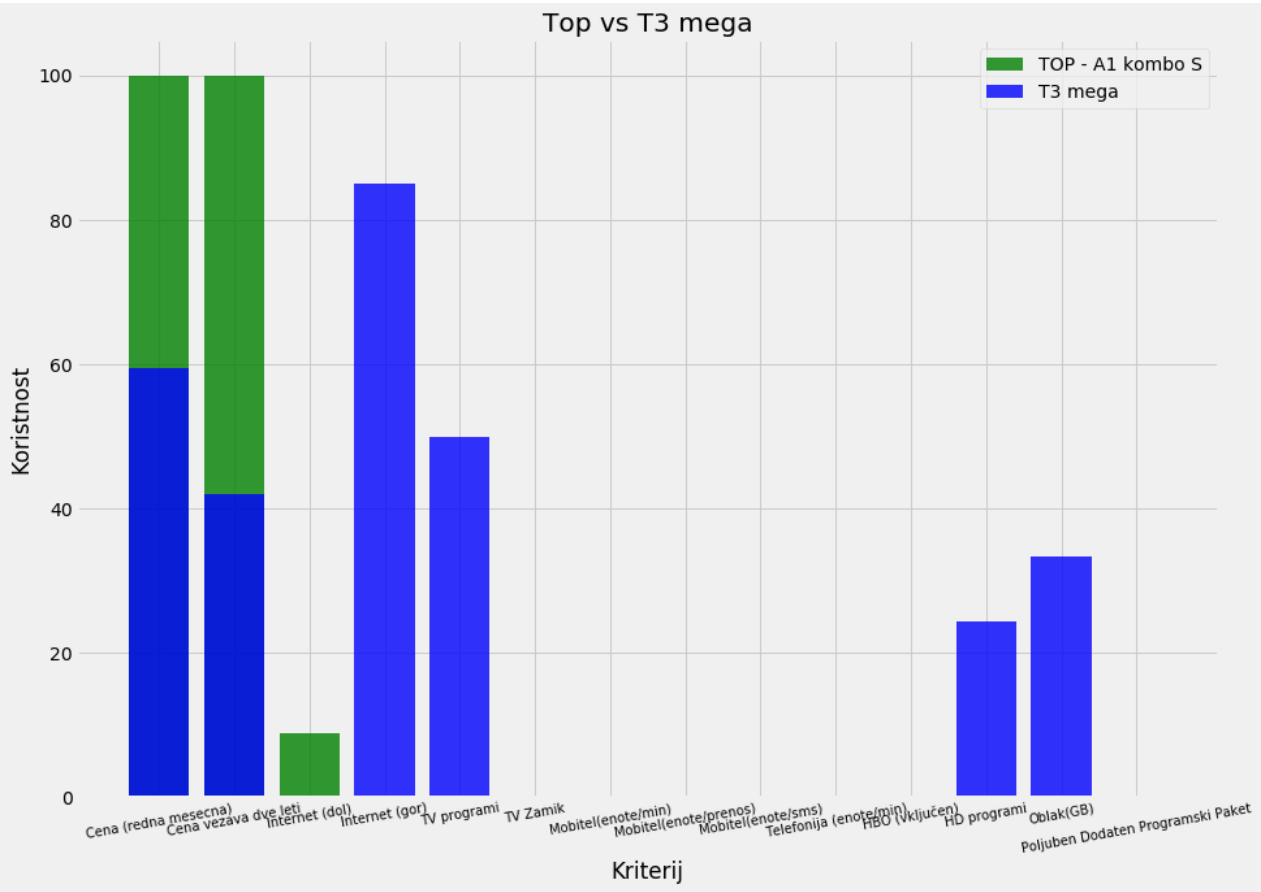












Komentarji Ne bom se razpisoval o posameznih primerjavah variante z najboljšo, saj večinoma lahko strnem v dve skupini:

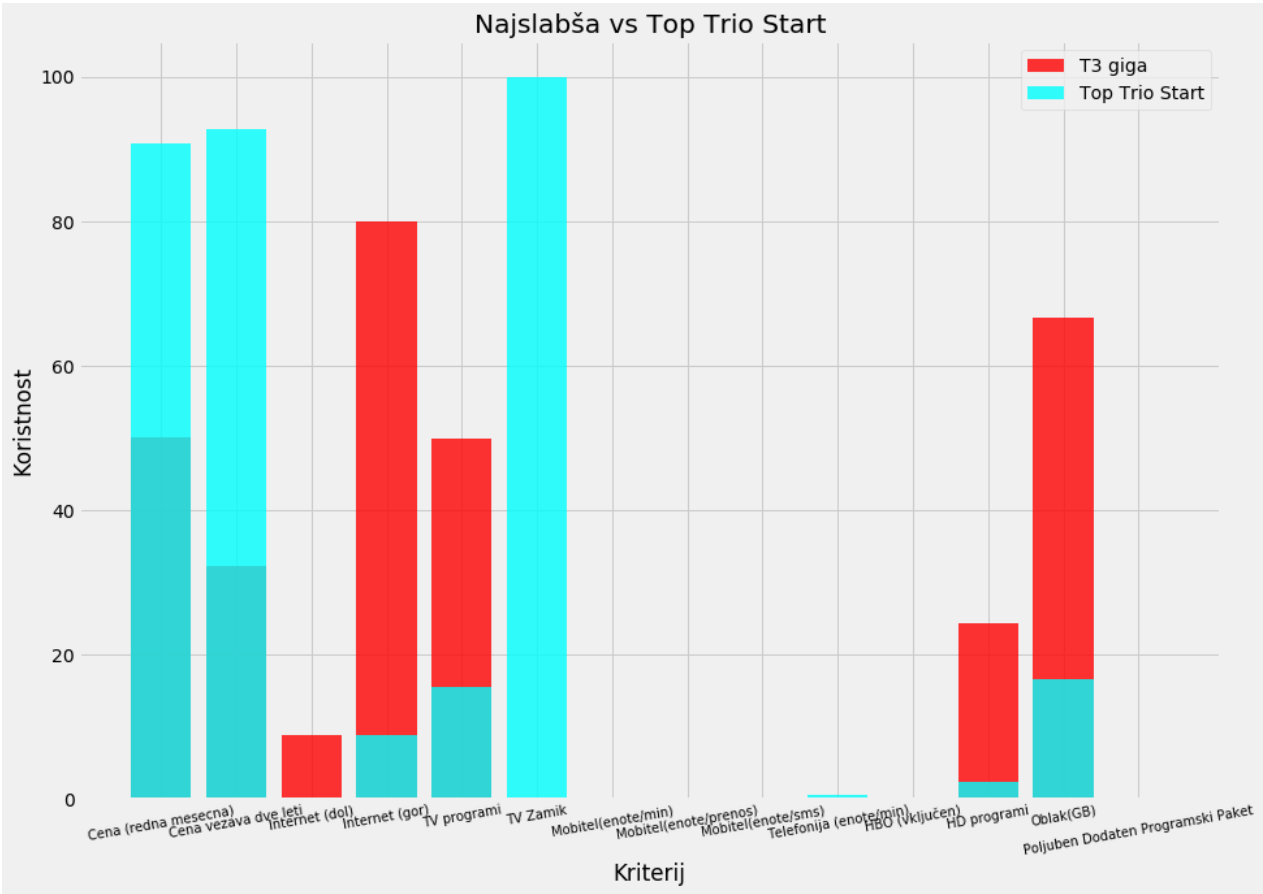
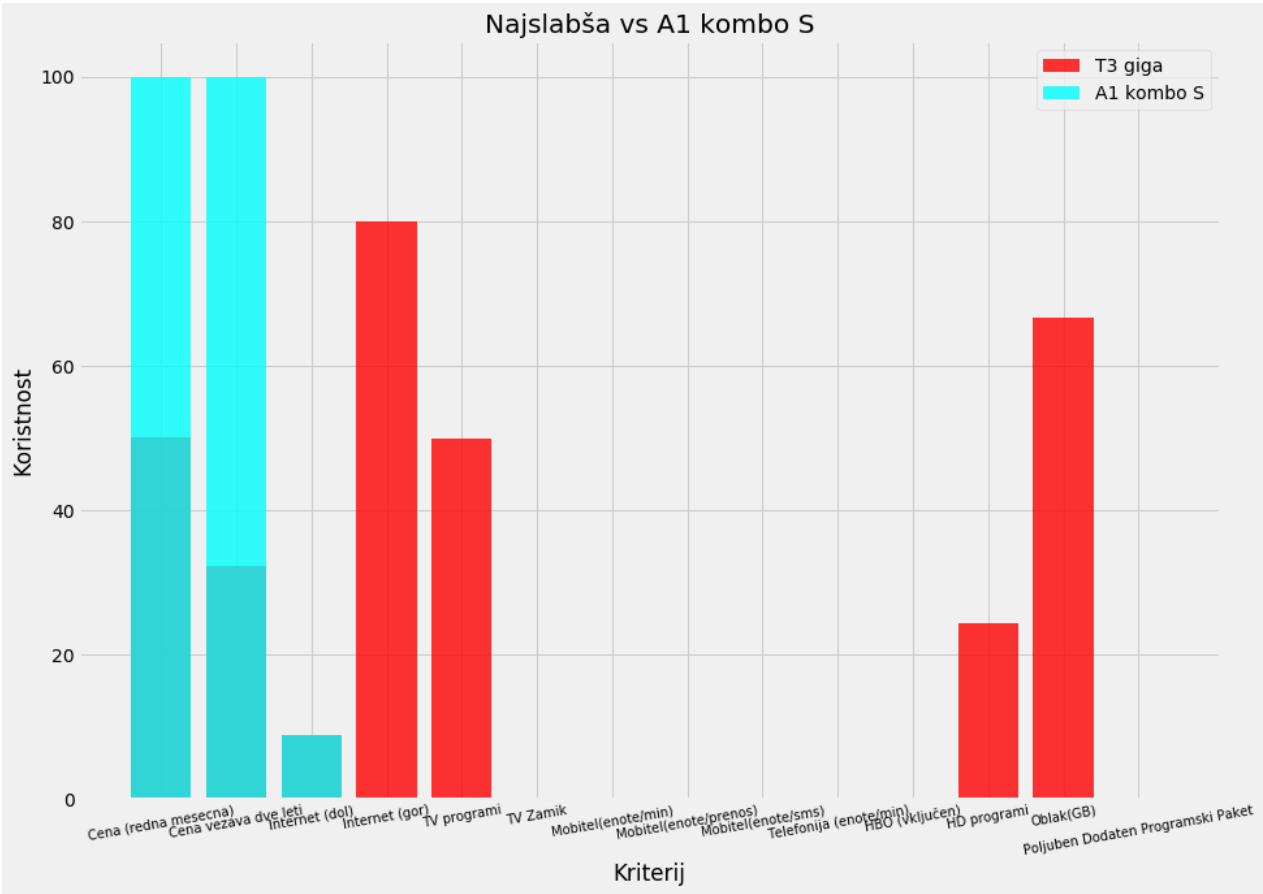
- najboljša je od vseh boljša po redni in pogodbeni ceni
- skoraj vse variante so boljše po ostalih kriterijih (tistih, ki nimajo povezave s ceno), le nekaj variant je še slabših po internetni (dol) hitrosti od najboljše izbire

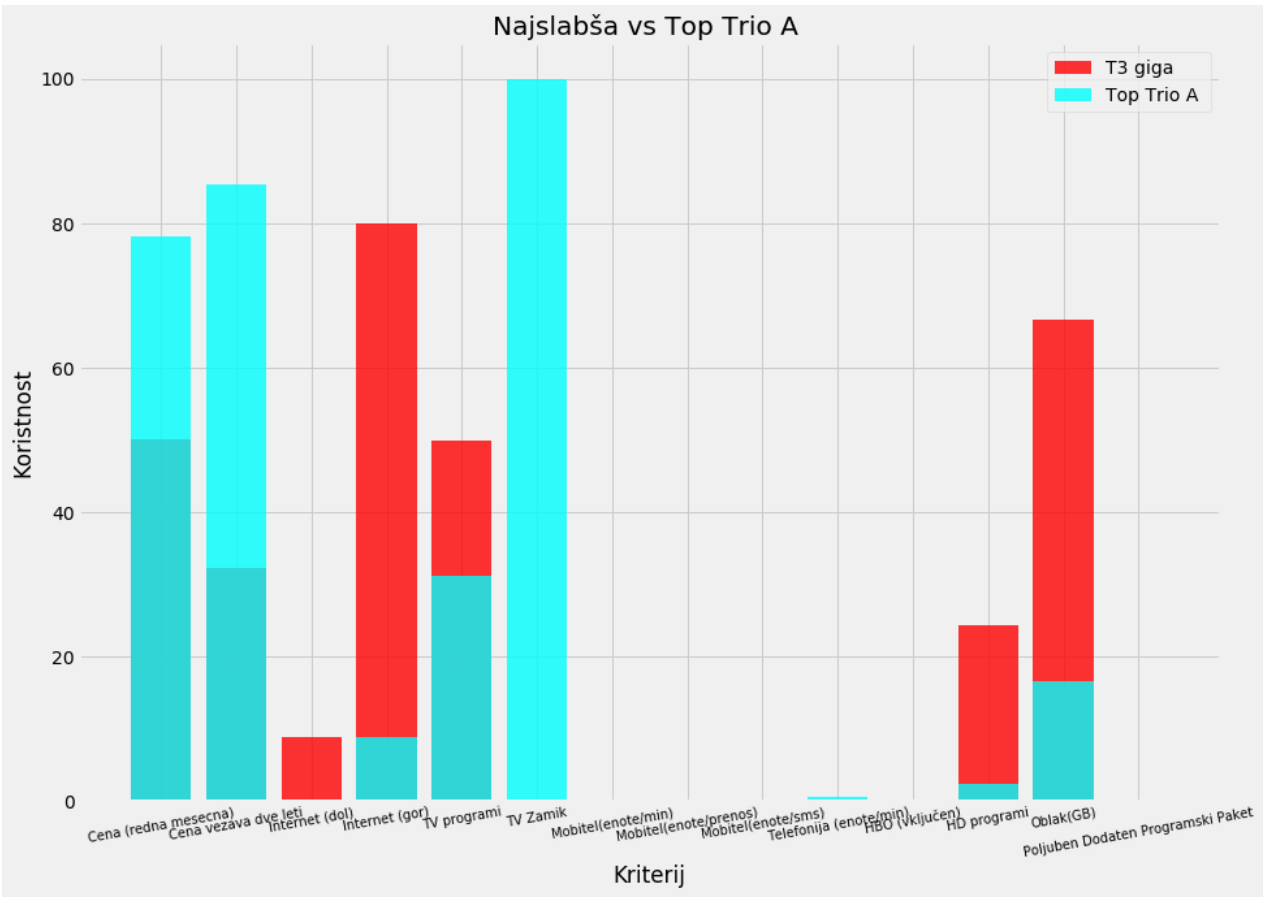
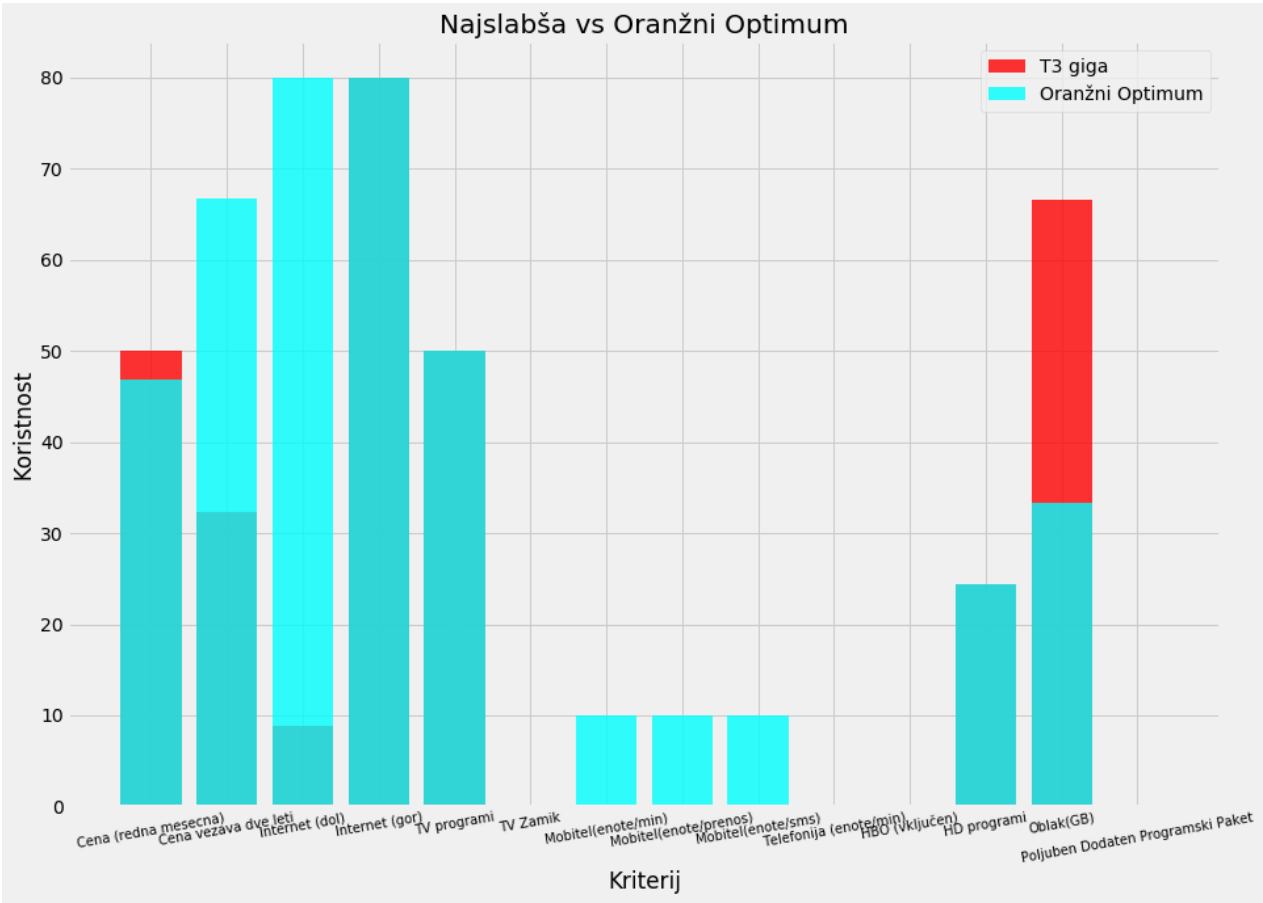
Najhitrejši zaključek je to, da nam je cena zelo pomembna in posledično zelo vpliva na najboljšo izbiro modela.

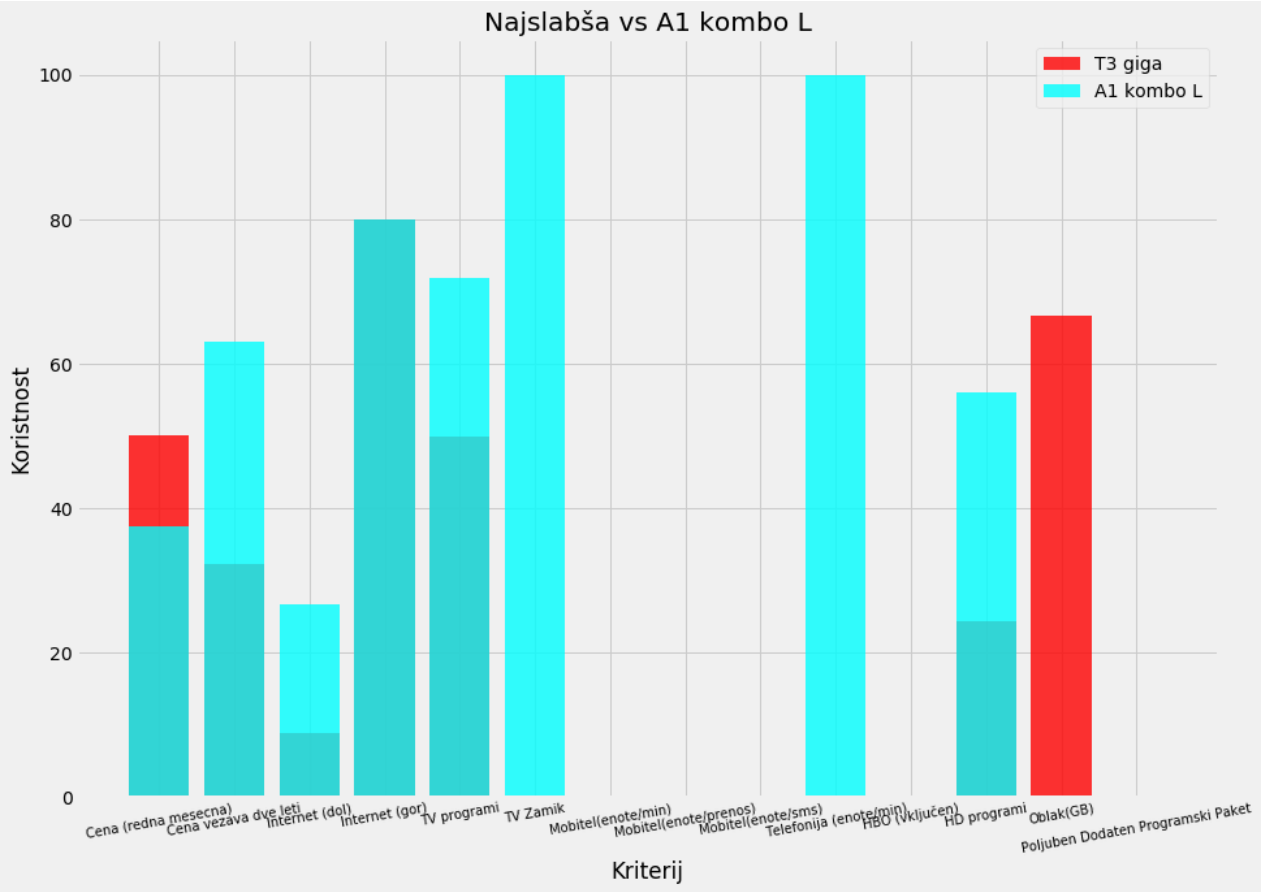
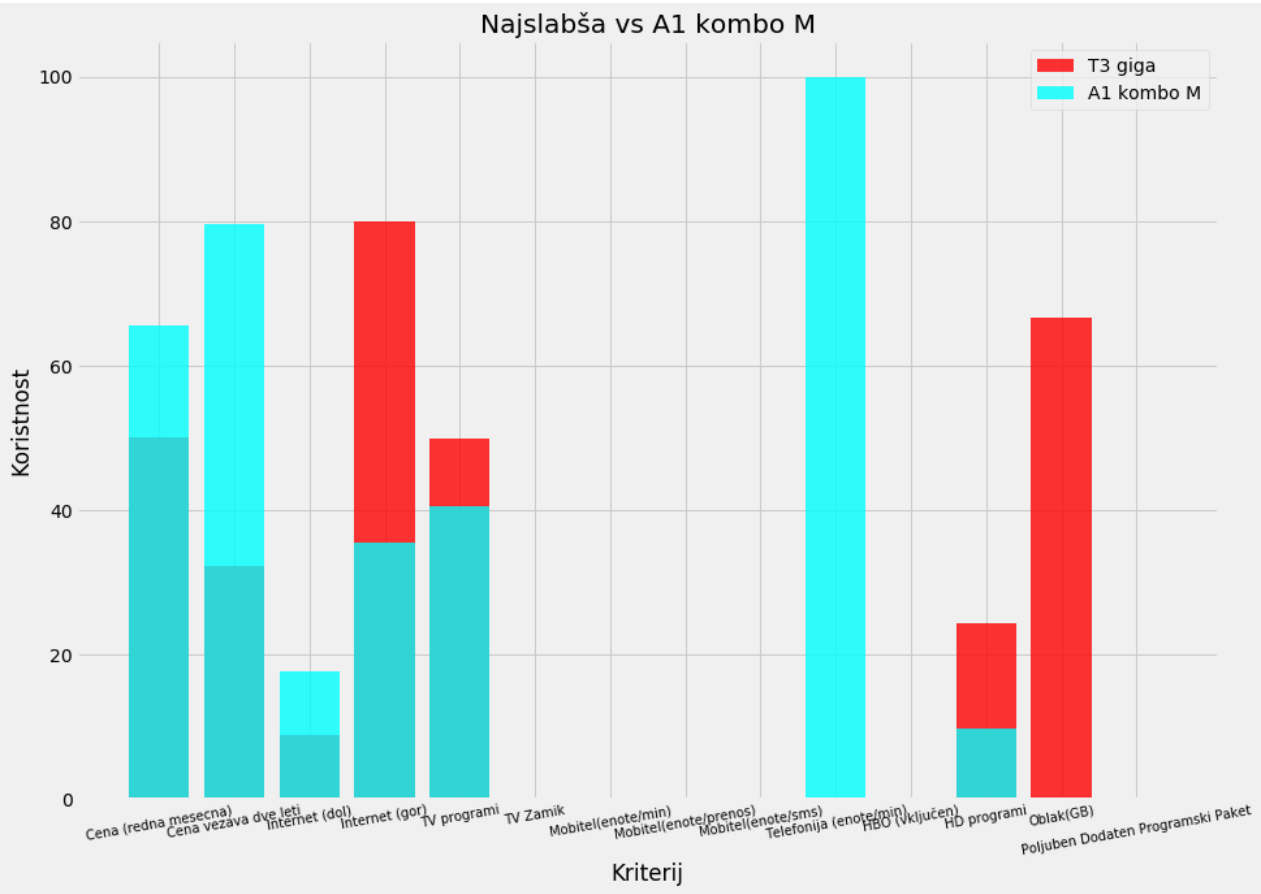
Primerjaj variante z najslabšo

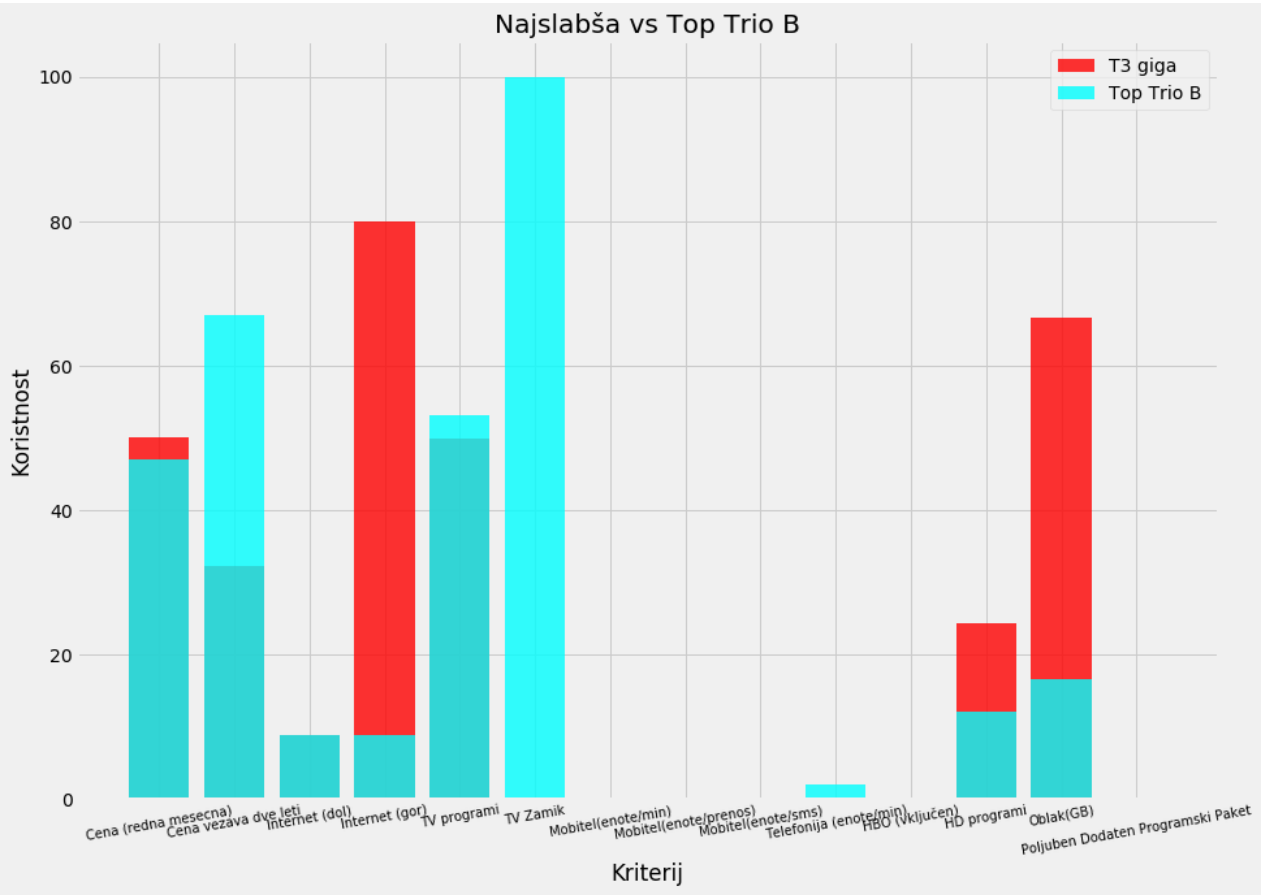
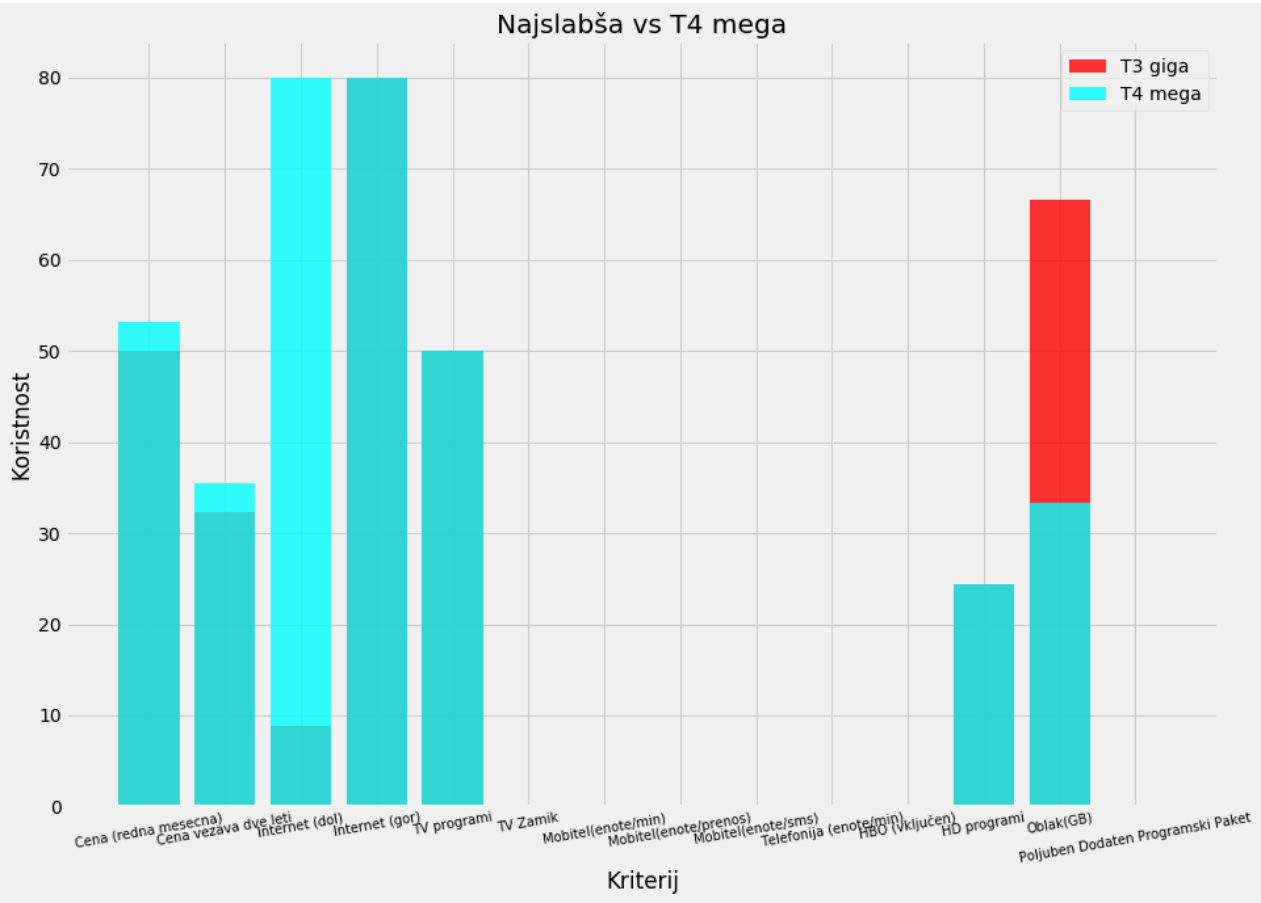
```
In [93]: najslabsa = tocke[-1][0]
ostale_notTop = [x[0] for x in tocke[:-1]]
y1 = dfKorist.loc[najslabsa]
x1 = np.arange((len(y1)))

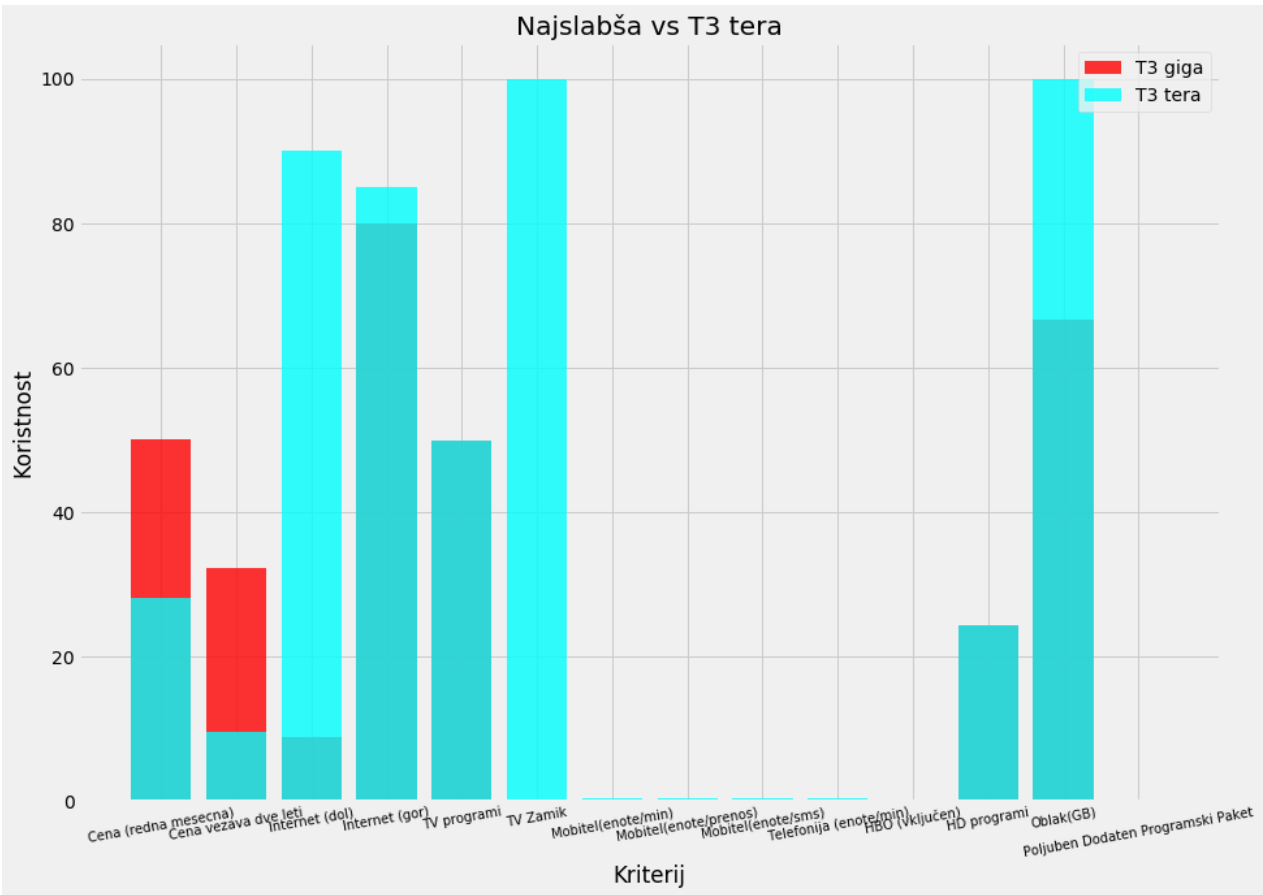
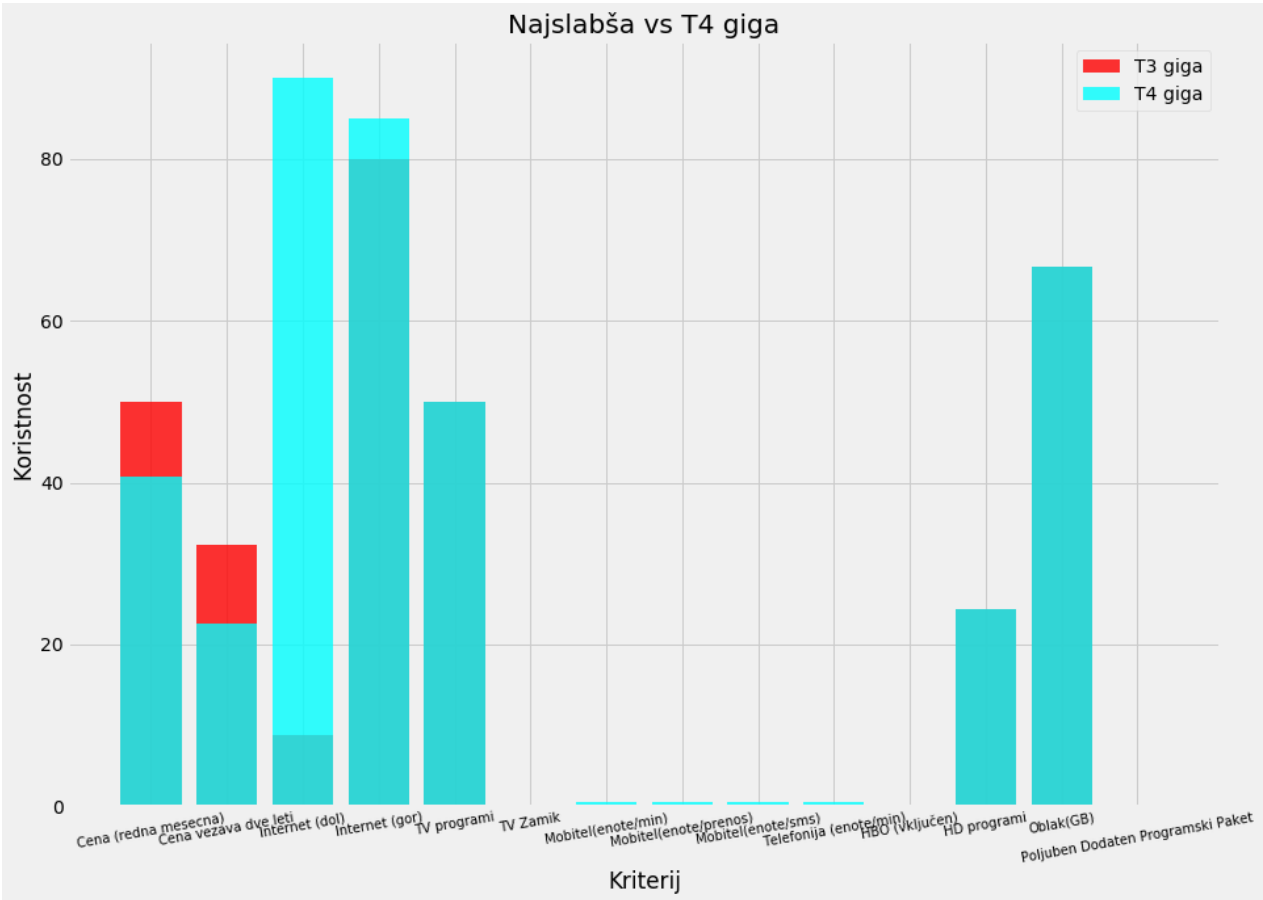
for varianta in ostale_notTop:
    fig, ax = plt.subplots(figsize=(14,10))
    ax.bar(x1, y1, color="red", alpha=0.8, label=najslabsa)
    y = dfKorist.loc[varianta]
    x = np.arange(len(y))
    #ax.fill_between(x, y, y1, where=y>=y1, alpha=0.8, color=barva)
    ax.bar(x, y, color="cyan", alpha=0.8, label=varianta)
    ax.set_title("Najslabša vs %s" % (varianta))
    ax.set_xticks(x)
    ax.set_ylabel("Koristnost")
    ax.set_xlabel("Kriterij")
    ax.set_xticklabels(dfKorist.columns, rotation=10, size=10)
    ax.legend(loc=1)
    plt.show()
```

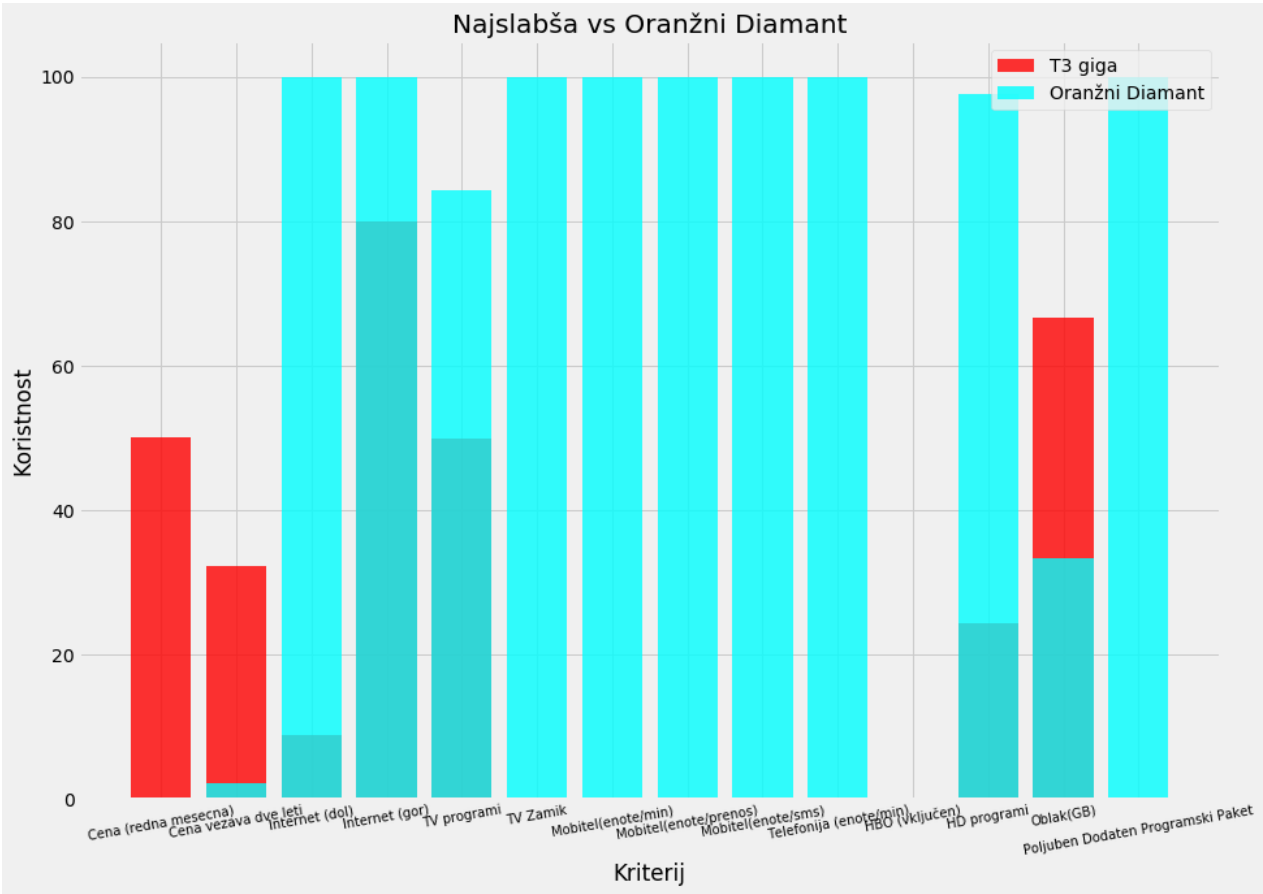
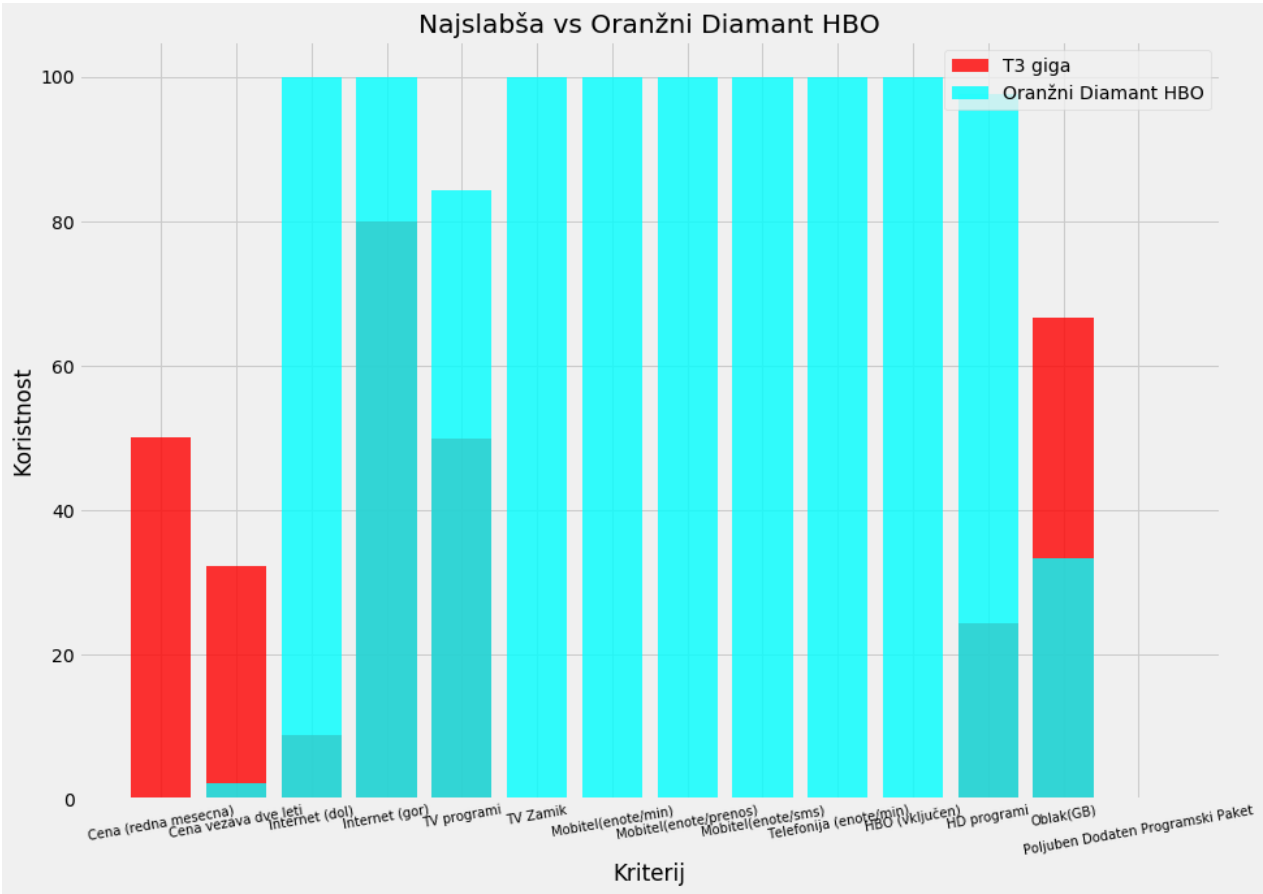


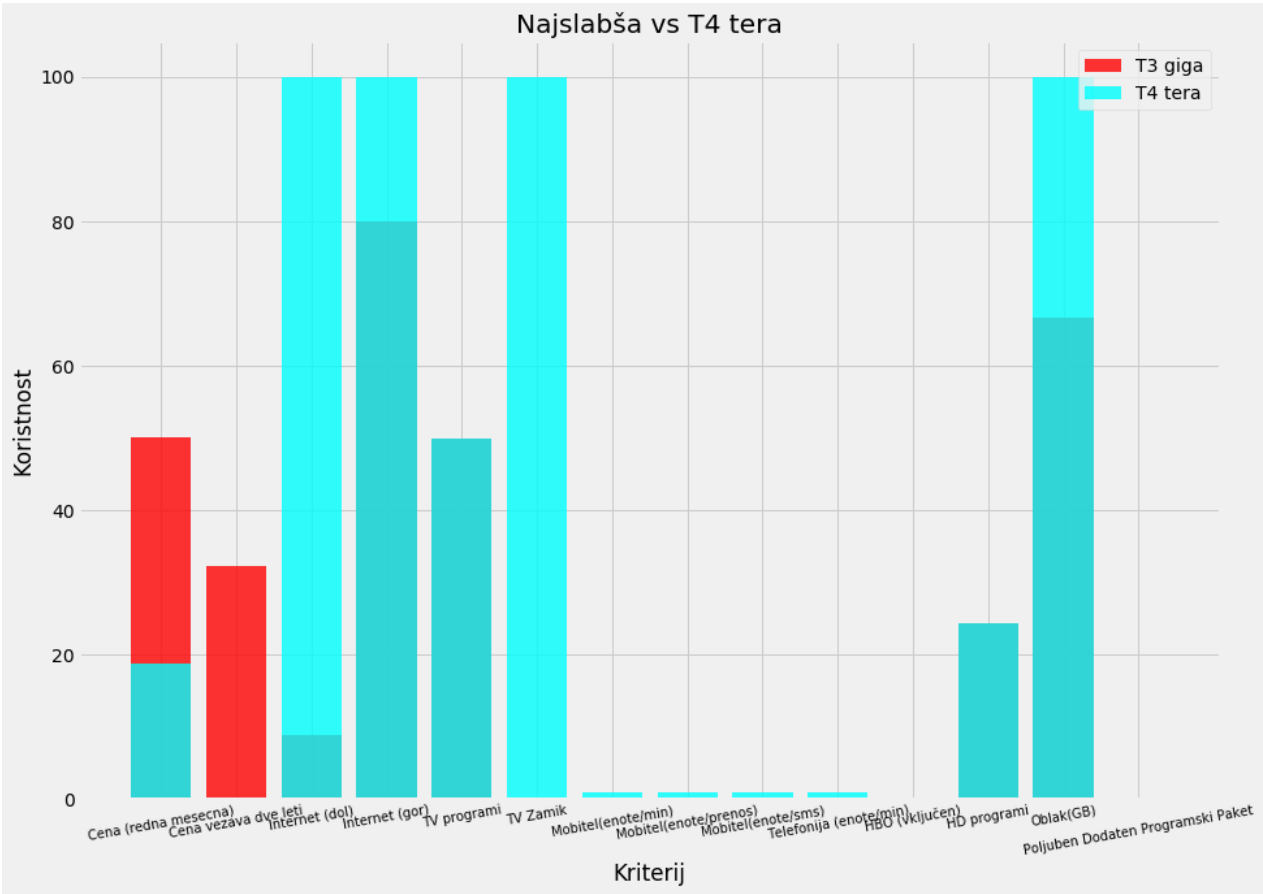
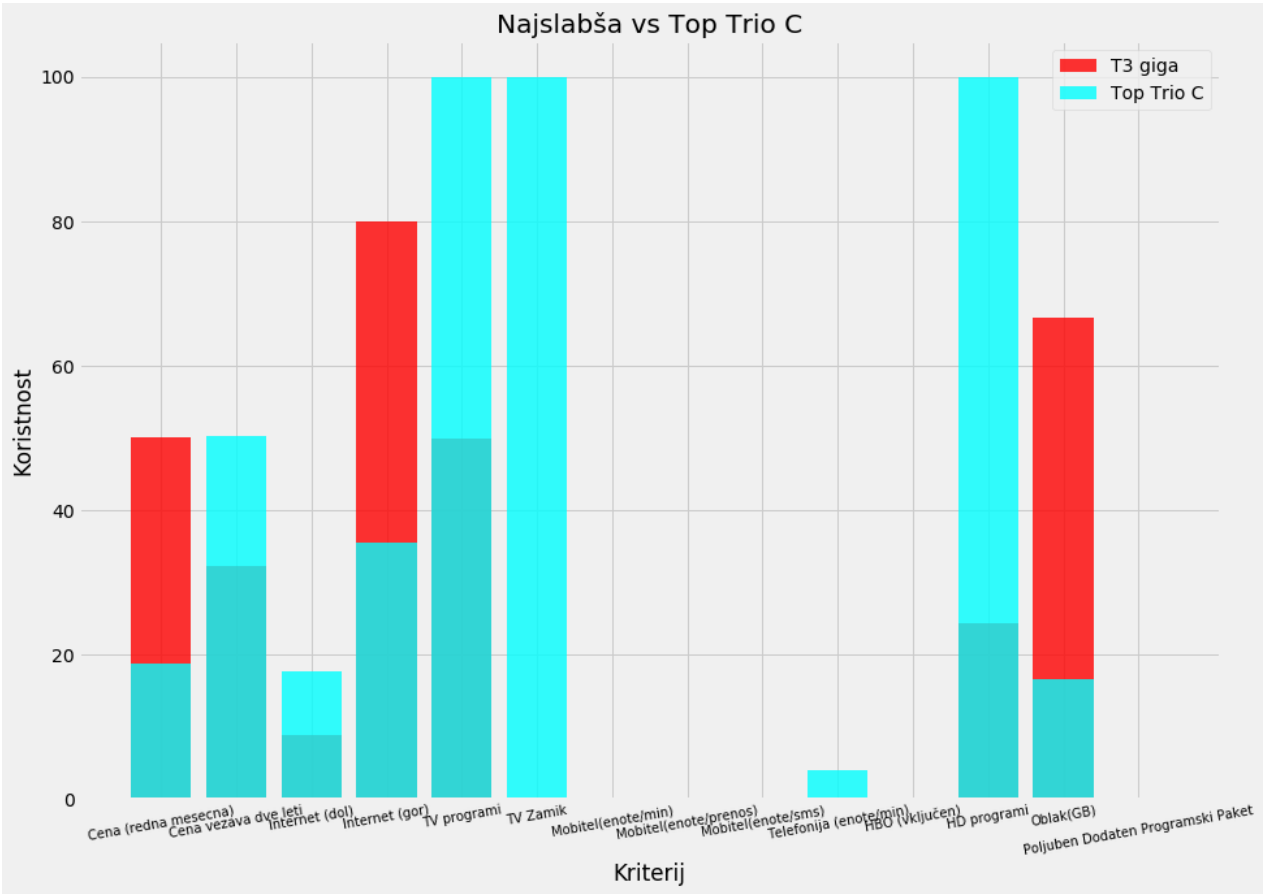


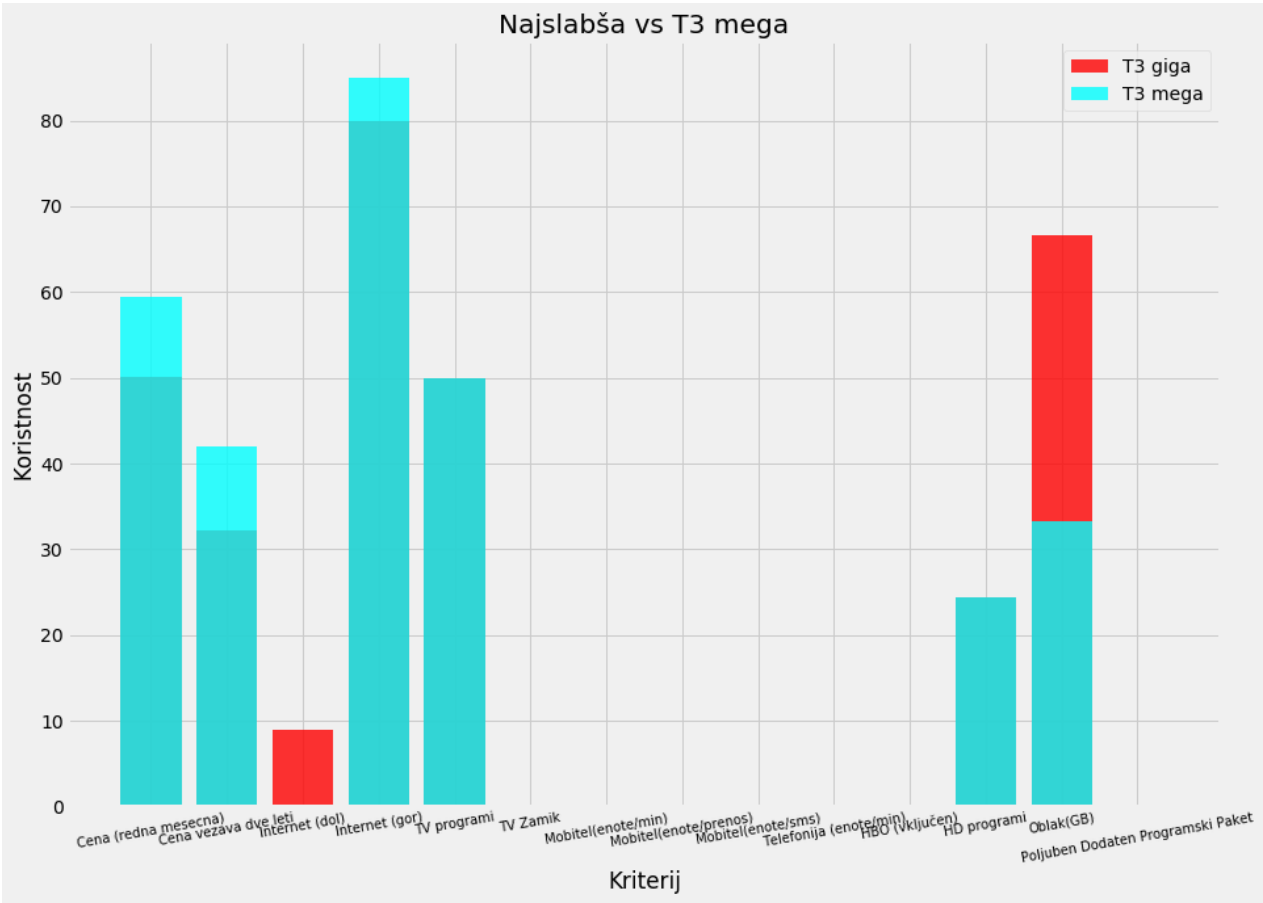












Komentarji

Najslabša varianta je večinoma slabša v ceni, če pa ni slabša v ceni ima ostale telekomunikacijske storitve opazno slabše od primerjane variante.

Moja alternativna varianta (Oranžni Optimum) je malenkost slabša v redni ceni, a je veliko boljša v ceni vezava dve leti, kar pomeni, da je očitno akcijska cena prvo leto res ugodna. Alternativna varianta je tudi veliko boljša v telekomunikacijskih storitvah z izjemo prostora, ki ga imamo na voljo v oblaku.

MAP

Map prikaz nam vizualizira odločitven model na podlagi le dveh kriterijev. To običajno uporabljamo na izpeljanih kriterijih. Izpeljana kriterija, ki jo uporabljam, sta:

- **cena**
- **telekomunikacije oziroma lastnosti paketa**

Spremenjena funkcija za uteženo glasovanje. Sedaj, če si cena (kategorija), se seštevaj v svojo spremenljivko, če si karkoli drugega pa v svojo.

```
In [95]: def weight_score2(varianta, weight):  
    cena = 0  
    lastnost = 0
```

```
for i in range(len(varianta)):
    # to je cena
    if i < 2:
        cena += varianta[i]*weight[i]
    else:
        lastnost += varianta[i]*weight[i]
return (cena, lastnost)
```

Nastavimo za vsako varianto njene koordinate - koordinata x je cena, koordinata y je lastnost/telekomunikacije:

```
In [96]: x_cena = []
y_lastnost = []
imena = dfKorist.index
for varianta in dfKorist.index:
    x,y = weight_score2(dfKorist.loc[varianta], utezi)
    x_cena.append(x)
    y_lastnost.append(y)
```

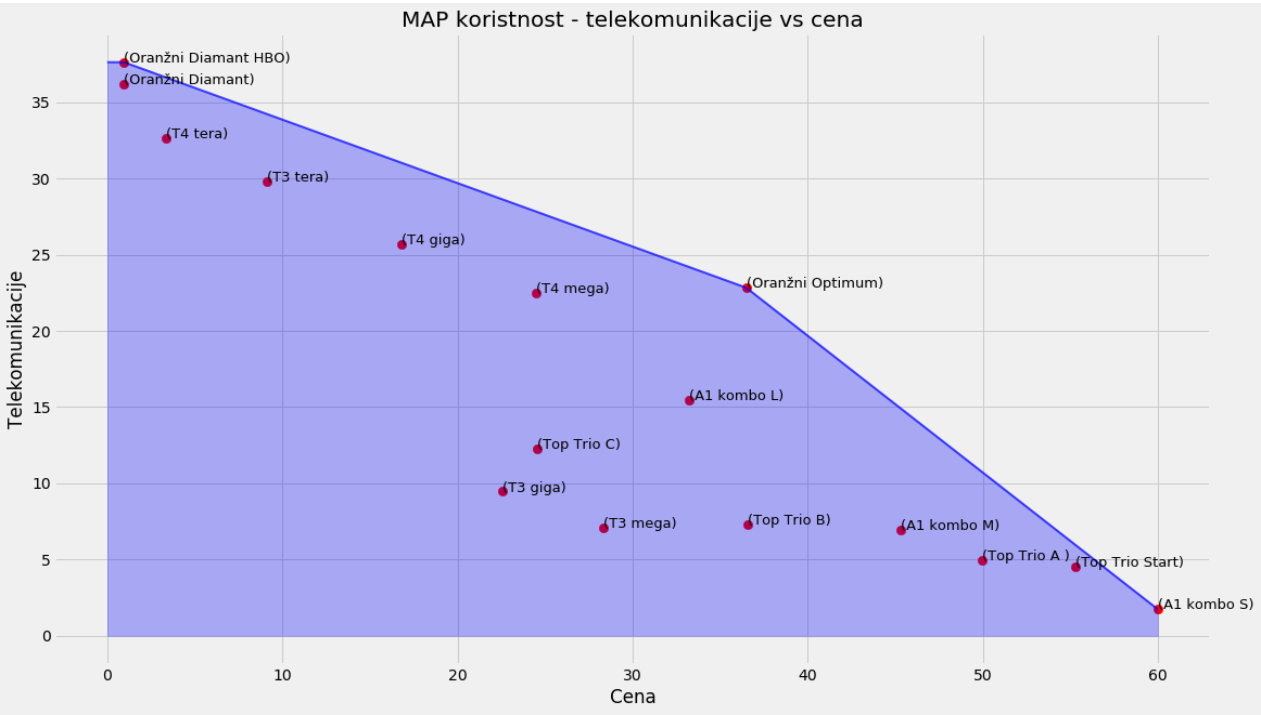
Grafično nariši MAP

```
In [101]: fig, ax = plt.subplots(figsize=(17,10))
# NARISI TOCKE
ax.scatter(x_cena, y_lastnost, color="red", s=70)
i = 0
for ime in imena:
    # DODAJ IMENA
    ax.annotate("(%s)"%ime, xy = (x_cena[i], y_lastnost[i]), textcoords='data', size=13)
    ax.line([0, y_lastnost[i]], [x_cena[i], y_lastnost[i]], color="k", linestyle="--", linewidth=0.5)
    i+=1

# narisi se ovojnico
# linija ki povezuje zunanje !!!!
linija_x = []
linija_y = []
for i in range(len(imena)):
    if imena[i] == "Oranžni Diamant HBO":
        linija_x.append(0)
        linija_y.append(y_lastnost[i])
        linija_x.append(x_cena[i])
        linija_y.append(y_lastnost[i])
    if imena[i] == "Oranžni Optimum" or imena[i] == "A1 kombo S":
        linija_x.append(x_cena[i])
        linija_y.append(y_lastnost[i])

# LINIJA
ax.plot(linija_x, linija_y, color="blue", alpha=0.7, linestyle="-", linewidth=2)
# POBARVAJ POD NJO
ax.fill_between(linija_x, linija_y, 0, color="blue", alpha=0.3)

ax.set_title("MAP koristnost - telekomunikacije vs cena")
ax.set_ylabel("Telekomunikacije")
ax.set_xlabel("Cena")
plt.show()
```



Komentar

Ovojnico napenjajo naslednje variante:

- Oranžni Diamant HBO
- Oranžni Optimum
- A1 kombo S Torej te so na optimalni liniji. Če bi hoteli dražji paket, bi bilo najbolj optimalno izbrati Diamant HBO, če bi želeli neko srednjo cenovno opcijo - Optimum, če pa najbolj cenovno - A1 kombo S.

Vidimo, da nekaterih variant sploh ni smiselno izbrati, so "manjvredne" v primerjavi z neko drugo varianto.

- **T3 giga vs Top Trio C** - Top Trio C je cenovno ugodnejši in boljši po lastnostih
- **T4 mega, T3 giga, Top Trio C, T3 mega, A1 kombo L, T4 mega vs Oranžni Optimum** - Optimum je cenejši in boljši po lastnostih od vseh teh
- **T3 giga vs T4 mega** - T4 mega je cenejši in boljši po lastnostih

Do teh manjvrednih variant pride, ker je cena sestavljena iz dveh cen - redne cene in cene za dve leti. Ko sem upravljal postopek izločitve, je lahko nekateri od teh paketov imel boljšo redno ceno, a slabšo ceno za dve leti. V skupni ceni pa je veliko poudarka na ceni za dve leti (razlog: več poudarka dajem na ceni za dve leti, ker je možnost ali da menjam paket po dveh letih ali menjam operaterja in se tako spet na novo vežem in koristim začetne akcijske cene).

Recimo tak primer je **T3 giga in Top Trio C**, ampak ne le to. Pri koristnosti ima tudi T3 boljšo internetno hitrost (gor), a ker je meni slednja nepomembna, to ne pripomore toliko k izboljšanju točk pri telekomunikacijah.

Občutljivost modela

Občutljivost modela nam pove, kako se spreminja naš izbor variant glede na izbiro uteži. Občutljivost modela sem testiral oziroma izrisal v prostoru izpeljanih kriterijev (cene), na x osi imamo uteži cene, na y osi pa točke oziroma koristnosti izračunane glede na novo težo cene.

Izračunaj koristnosti glede na spremembo uteži ceni Ne pozabi še izračunati zmagovalca in pogledati, če se je rezultat spremenil od prejšnjega. Če se je, si zapomni pri kateri uteži se je to zgodilo!

```
In [125]: ## SENZITIVNOST
## Za analizo občutljivosti preverimo občutljivost modela na spremembe uteži.
# Torej, direktno odgovarja na vprašanje: ali bi
# majhna sprememba uteži bistveno vplivala na naš odločitven model?

# nastavimo c in jo spreminjamo
variabilna_utez_cena = np.linspace(0, 1, 40)
slovar = {}
prejsnji_zmagovalec = -1 # za trackanje zmagovalcev
spremembe = 0
utezi_sprememba = []
for c in variabilna_utez_cena:
    # ostalo utez dobimo tako da 1 - c
    telek = 1 - c
    utezi_tmp = [c*rednaCena, c*cena2leti, telek*inter*dol, telek*inter*gor, telek*teve*programi, telek*teve*za
mik,
                telek*telefonija*gsm*mobiM, telek*telefonija*gsm*mobiPaketi, telek*telefonija*gsm*mobiS, telek*tel
efonija*stacionarni,
                telek*teve*paketi*hb, telek*teve*paketi*hdd, telek*inter*cloud, telek*teve*paketi*extr]

    # sedaj v vsaki iteraciji spremenimo utez ceni in nato popravimo vse utezi
    # v spodnji zanki gremo cez vse iteracije in jim izracunamo tocke glede na utezi
    maksimalna_varianta = -1 # le dve spremenljivki za keepanje track z maksimalno
    maksimalna_koristnost = -1
    for varianta in dfKorist.index:
        utility = weight_score(dfKorist.loc[varianta], utezi_tmp)
        #zapomni si maksimalno koristnost do sedaj
        if maksimalna_koristnost < utility:
            maksimalna_koristnost = utility
            maksimalna_varianta = varianta
        if varianta not in slovar:
            slovar[varianta] = [(c, utility)]
        else:
            slovar[varianta].append((c, utility))

    #da v prvi iteraciji pristejemo spremembo in nastavimo prejsnjega zmagovalca
    #prejsnji_zmagovalec == -1
    if prejsnji_zmagovalec == -1 or prejsnji_zmagovalec != maksimalna_varianta:
        prejsnji_zmagovalec = maksimalna_varianta
    # povecaj spremembe v vrhu
    spremembe += 1
    utezi_sprememba.append(c)

print("Število različnih zmagovalcev:", spremembe, "\nTi zmagovalci 'nastopijo' pri utežeh cene:", utezi_sprememba)
```

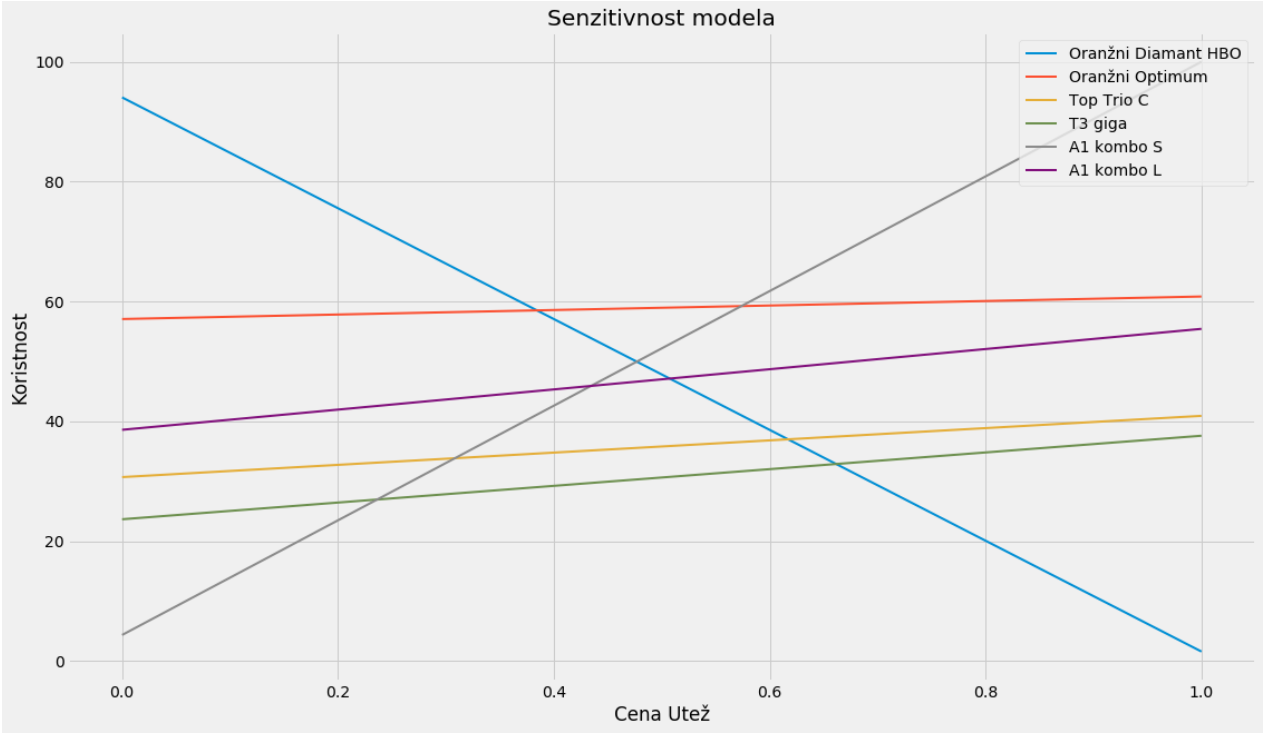
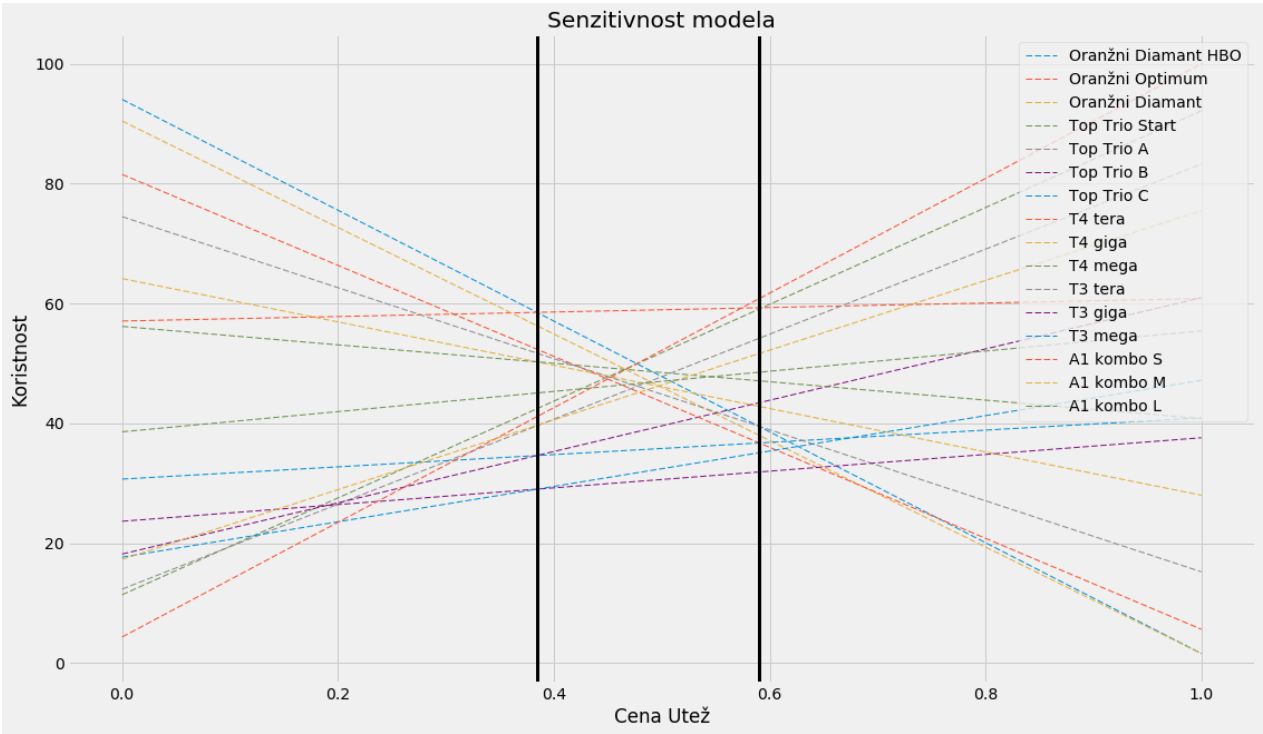
Število različnih zmagovalcev: 3

Ti zmagovalci 'nastopijo' pri utežeh cene: [0.0, 0.38461538461538458, 0.58974358974358976]

Grafično

```
In [130]: # NARISI GRAF SENZITIVNOSTI
fig, ax = plt.subplots(figsize=(17,10))
for varianta in slovar:
    xi = [xy[0] for xy in slovar[varianta]]
    yi = [xy[1] for xy in slovar[varianta]]
    ax.plot(xi, yi, linestyle="--", linewidth=1, label=varianta)
    ax.set_xlabel("Cena Utež")
    ax.set_ylabel("Koristnost")
    ax.set_title("Senzitivnost modela")
    #prve črte kjer nastopi prvi zmagovalec ne izrišem, zaradi lepšega!
    for x in utezi_sprememba[1:]:
        ax.axvline(x, color="k", linestyle="-", linewidth=3, alpha=0.7)
    #ax.set_xticks(xi)
    ax.legend(loc=1)
plt.show()

# IN SE MALENKOST BOLJ REDEK
fig, ax = plt.subplots(figsize=(17,10))
for varianta in slovar:
    if varianta == "T3 giga" or varianta == "Top Trio C" or varianta == "Oranžni Diamant HBO" or varianta
    == "A1 kombo S" or varianta == "Oranžni Optimum" or varianta == "A1 kombo L":
        xi = [xy[0] for xy in slovar[varianta]]
        yi = [xy[1] for xy in slovar[varianta]]
        ax.plot(xi, yi, linestyle="-", linewidth=2, label=varianta)
        ax.set_xlabel("Cena Utež")
        ax.set_ylabel("Koristnost")
        ax.set_title("Senzitivnost modela")
        #ax.set_xticks(xi)
        ax.legend(loc=1)
plt.show()
```



Komentar

Kar vidim je, da se mi najboljši zmagovalec spremeni glede na to, kakšno utež izberem za izpeljano kategorijo ceno.

- Če je ta utež **od 0 do 0.38**, je zmagovalec Oranžni Diamant HBO, ki ima od vseh paketov najboljše telekomunikacije za moje nastavitve, a je tudi najdražji.
- Utež med **0.38 in 0.59**: V tem območju je zmagovalna moja alternativna varianta, to je Oranžni Optimum. Ta paket ima dobro kombinacijo vseh kategorij, dobre telekomunikacije in spodobno ceno.

- Utež **od 0.59 do 1**: Če damo ceni tako visoko utež, bo zmagovalec A1 kombo S, ki je izmed vseh paketov najcenejši, a ima zelo slabe telekomunikacije - malo programov, počasne internetne hitrosti, brez telefona, itd.

Kaj če ?

Kaj če se redna cena alternativne variante - Oranžni Optimum, zniža? Bi potem zmagal v točkovanju? Za koliko bi se morala redna cena znižati, da bi zmagal?

Postopek Sprehodim se po ceni od trenutne cene pa do cene 0 evra. Potrebno je posodobiti tudi ceno za dve leti, saj je v to ceno všteta 12 mesečna akcijska cena in 12 mesečna redna cena. Ker sem akcijsko zavrgel, moram novo pogodbeno ceno izračunati na drugačen način. Preprosto vzamem prejšnjo pogodbeno ceno in od nje odštejem razliko zmanjšanja redne cene pomnožene z 12. V prvi iteraciji je razlika 0, ker grem čez z default vrednostmi. V drugi iteraciji pa bo dodatna spremenljivka hranila v sebi vrednost prejšnje redne cene. Nato od te dodatne spremenljivke odštejem novo redno ceno in dobim razliko. Novo ceno vsakič znova prevedem v koristnost in editiram vrstico oziroma varianto Oranžni optimum (posodobim vrednosti cene v vektorju koristnosti).

```
In [116]: """
varianta: A1 kombo S tocke: 61.7422222222
varianta: Top Trio Start tocke: 59.8584474632
varianta: Oranžni Optimum tocke: 59.3258198182
"""

# Koliko mora biti redna cena nizja, da se nam splaca?
# treba je dodati, da če manjšamo redno ceno, se manjša tudi dvo letna cena, saj je polovico nje redna cena
# popravljaj tako, da vsako zmanjšanje v redni ceni - > dvoletna cena - 12*zmanjšanje_redne_cene

#to so koristnosti !!
vrsticaOOT = dfKorist.loc['Oranžni Optimum']
#redna cena, start, konec, koliko tock
price_movement = np.linspace(vrsticaOOT[0], 0, 30)
prejsnja_redna = df.loc['Oranžni Optimum', "Cena (redna mesecna)"]
dveLeti_cena = df.loc['Oranžni Optimum', "Cena vezava dve leti"]
for price in price_movement:
    redna_cena = price
    dveLeti_cena = dveLeti_cena - 12*(prejsnja_redna-redna_cena)

#nastavim prejsnjo redno na prejsno vrednost
prejsnja_redna = redna_cena
#updatejam vrstico, da jo posljem v izracun
tmp_vrstica = vrsticaOOT.copy()
#moramo cene pretvoriti v koristnosti
tmp_vrstica[0] = cena_redna(redna_cena)
tmp_vrstica[1] = cena_vsa(dveLeti_cena)
tocke = weight_score(tmp_vrstica, utezi)

if tocke > 61.7422222222:
    print("TOČKE", tocke, ". Za koliko bi se morala zmanjšati redna cena, "
          "da bi bil Oranžni Optimum najboljša izbira:", df.loc['Oranžni Optimum', "Cena (redna mesečna)"] - redna_cena,
          "cena akcija:", redna_cena, ", cena regular:", df.loc['Oranžni Optimum', "Cena (redna mesecna)"]],
```

break

"Toliksna bi bila celotna cena za dve leti: ", dveLeti_cena)

TOČKE 63.4905096232 . Za toliko bi se morala zmanjšati redna cena, da bi bil Oranžni Optimum najboljša izbi
ra: 3.115 cena akcija: 46.875 , cena regular: 49.99 Toliksna bi bila celotna cena za dve leti: 802.38

Komentar Izkaže se, da če bi bila redna cena paketa za 3.1 evra cenejša, bi bil to najbolj optimalen
paket po trenutno nastavljenih utežeh. Za paket bi morali letno odšteti 802.38 evra, mesečno pa prvo
leto 19.99, drugo leto pa 46.88 evra.