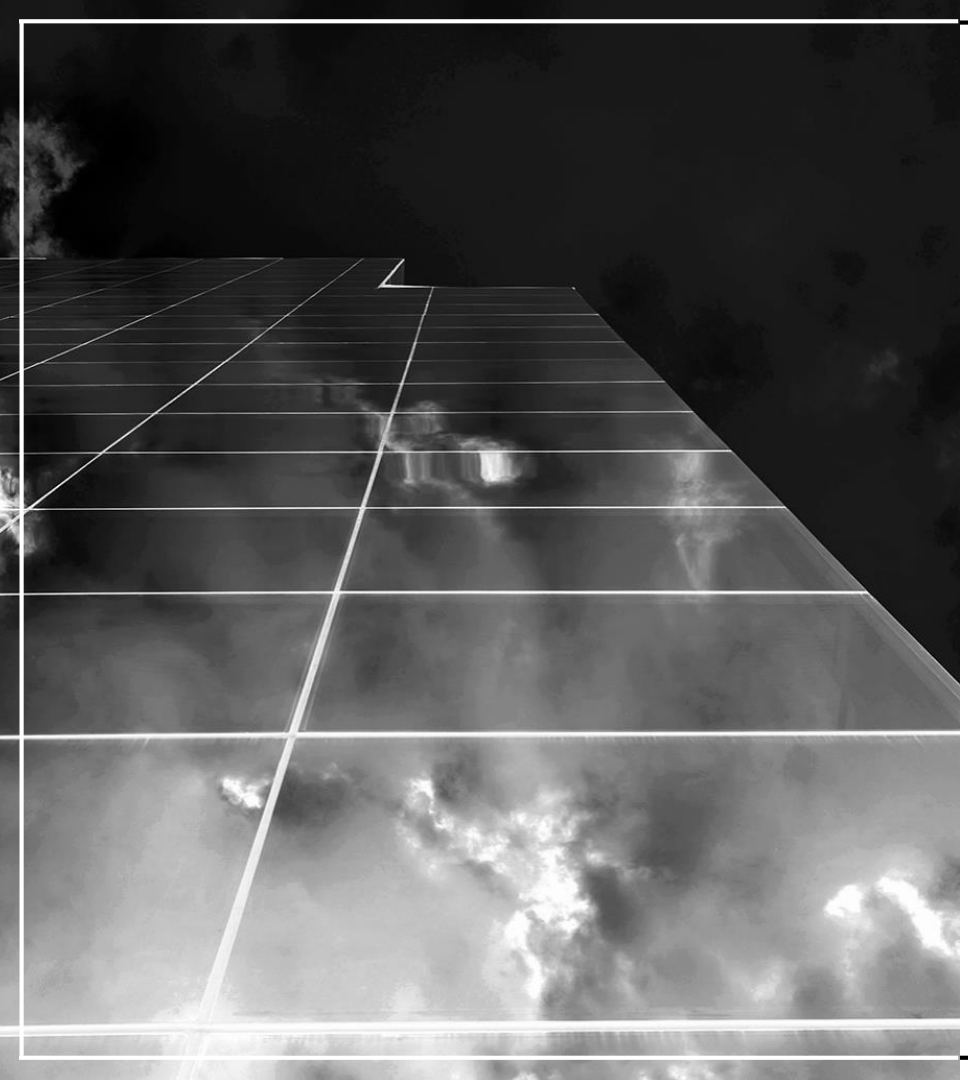
# DAŽNIAUSIAI TEKSTE PASITAIKANČIŲ ŽODŽIŲ PAIEŠKA

PROJEKTINIS DARBAS

# TURINYS

# 01 UŽDUOTIS



# PROJEKTINIO DARBO UŽDUOTIS

Sukurti programą, kuri rastų tekste dažniausiai pasitaikančius žodžius, išskyrus nereikšmingus žodžius. (angl. stop words).

# 02 GRUPĖS SUDĖTIS

# KOMANDA

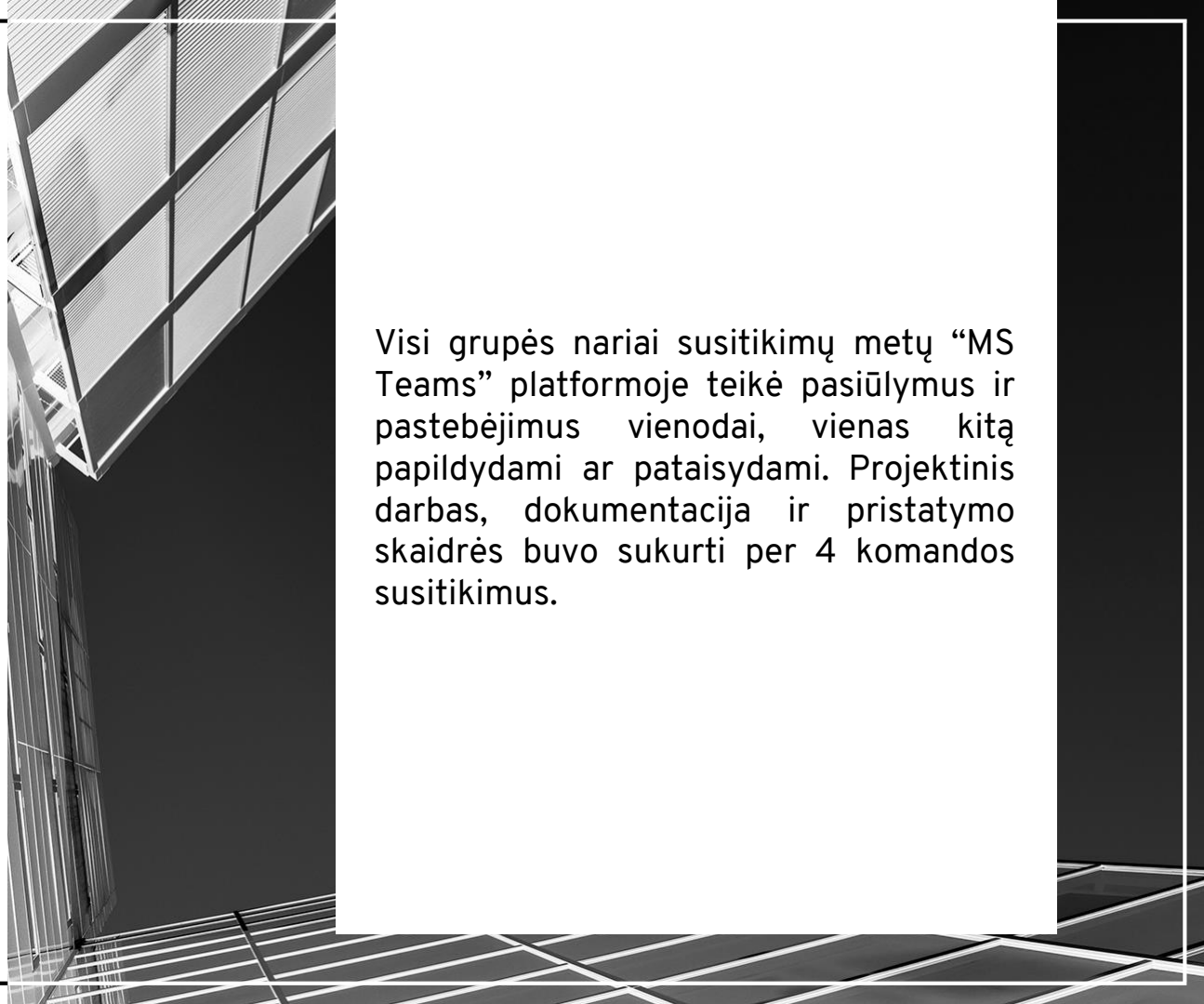
OSKARAS DIRAITIS

GYTAUTĖ BARZDŽIŪTĖ

TOMAS KLEVAS

DAINIUS MASEVIČIUS

# GRUPĖS SUDĖTIS IR PASISKIRSTYMAS PAREIGOMIS



Visi grupės nariai susitikimų metų “MS Teams” platformoje teikė pasiūlymus ir pastebėjimus vienodai, vienas kitą papildydami ar pataisydami. Projektinis darbas, dokumentacija ir pristatymo skaidrės buvo sukurti per 4 komandos susitikimus.

# 03 UŽDUOTIES REALIZAVIMAS

# TEKSTINIAI FAILAI

## STOPWORDS.TXT

“Stopwords” failo turinys yra toks: the, of, and, in, a, by, its, was, to, at, as, it, has, but, not. Tai nereikšmingi žodžiai, kurie nėra skaičiuojami kaip dažniausiai pasikartojantys žodžiai tekste. “Text” failo turinys - tai tekstas, kuris yra nagrinėjamas.

## TEXT.TXT

“Text” failo turinys - tai tekstas, kuris yra nagrinėjamas.

# NAUDOJAMOS BIBLIOTEKOS

## COLLECTIONS

Importuojamos talpyklos, kuriose galima saugoti duomenis, jų kolekcijas (pavyzdžiui list, dict, set, tuple).

## PANDAS

Leidžia pateikti duomenis tokiu būdu, kuris yra tinkamas duomenų analizei

## MATHPLOTLIB.PYPLOT

Leidžia naudotis MATLAB programos privalumais: leidžia sukurti figūras, linijas, etiketes ir panašiai.

# PROGRAMOS KODAS (1)

Pirma, nuskaitomas failas, su nurodyta koduote (žr. pav. 2). UTF-8 koduotė - viena kintamo ilgio simbolių koduočių, kuria galima užrašyti bet kokį Unikodo simbolį.

```
# Read input file
# The encoding is specified here also
file = open('text.txt', encoding="utf8")
text = file.read()
```

Nustatomi žodžiai, kurie nebus skaičiuojami kaip dažniausiai pasikartojantys. Tai vadinamieji “nereikšmingi žodžiai”. Vartotojas gali pats papildyti nereikšmingų žodžių sąrašą. Tai rodo programos lankstumą, tačiau atneša ir šiokių tokių trūkumų

```
# Stopwords
stopwords = set(line.strip() for line in open('stopwords.txt'))
stopwords = stopwords.union(set(['mr','mrs','one','two','said']))
```

# PROGRAMOS KODAS (2)

Sukuriamas naudojamo teksto žodynas: jei žodis tame tekste yra naujas, jis įrašomas į žodyną, jeigu jau egzistuoja, padidinamas jo kiekis.

```
# Instantiate a dictionary, and for every word in the file,
# Add to the dictionary if it doesn't exist. If it does, increase the count.
wordcount = {}
```

Norint eliminuoti dublikatus (pavyzdžiui Vilnius, Vilnius ir Vilnius:) pašalinami simboliai, sumažinamos didžiosios raidės. Sukuriamas ankstesniame žingsnyje minėtas teksto žodynas

```
# To eliminate duplicates, remember to split by punctuation, and use case demiliters.
for word in text.lower().split():
    word = word.replace(".","")
    word = word.replace(",","")
    word = word.replace(":","")
    word = word.replace("\"","")
    word = word.replace("!","")
    word = word.replace("â€œ","")
    word = word.replace("â€˜","")
    word = word.replace("*","")
    if word not in stopwords:
        if word not in wordcount:
            wordcount[word] = 1
        else:
            wordcount[word] += 1
```

# PROGRAMOS KODAS (3)

Vartotojui leidžiama pasirinkti kiek dažniausiai pasikartojančių žodžių spausdinti. Atspausdinami rezultatai.

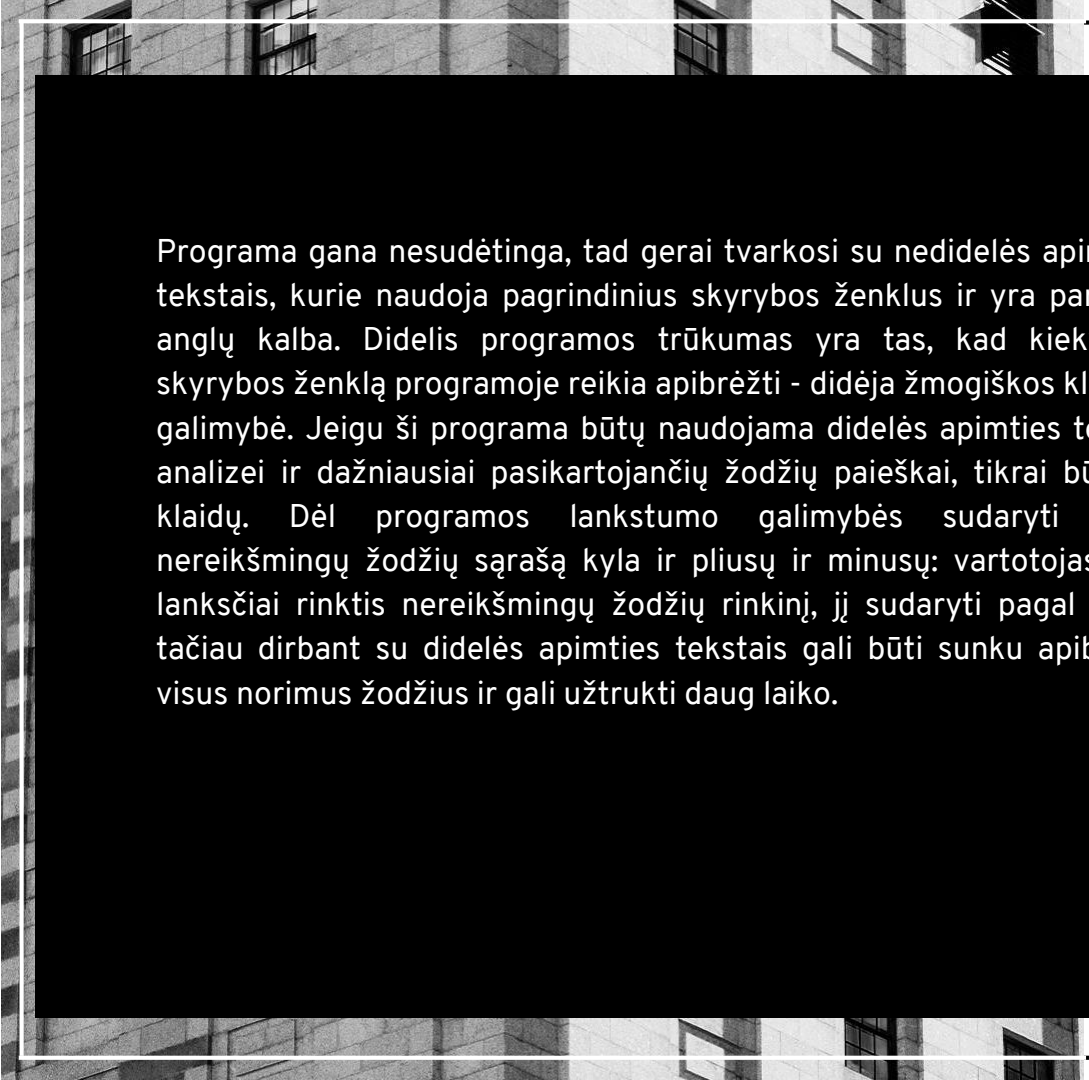
```
# Print most common word
n_print = int(input("How many most common words to print: "))
print("\nOK. The {} most common words are as follows\n".format(n_print))
word_counter = collections.Counter(wordcount)
for word, count in word_counter.most_common(n_print):
    print(word, ": ", count)
```

```
How many most common words to print: 10

OK. The 10 most common words are as follows

university :  20
vilnius :  8
lithuania :  7
famous :  5
history :  4
soviet :  4
higher :  3
cultural :  3
vilnensis :  3
from :  3
```

# PROGRAMINĖ ĮRANGA

Programos kūrimui, testavimui, rezultatų atvaizdavimui - projekto kūrimui pasirinkta naudoti Jupyter NoteBook. Tai patogus, greitas darbo įrankis, leidžiantis ne tik matyti tekstinius, tačiau ir vizualizuotus rezultatus.

# 04 TESTAVIMO IŠVADOS



Programa gana nesudėtinga, tad gerai tvarkosi su nedidelės apimties tekstais, kurie naudoja pagrindinius skyrybos ženklus ir yra parašyti anglų kalba. Didelis programos trūkumas yra tas, kad kiekvieną skyrybos ženklą programoje reikia apibrėžti - didėja žmogiškos klaidos galimybė. Jeigu ši programa būtų naudojama didelės apimties tekstų analizei ir dažniausiai pasikartojančių žodžių paieškai, tikrai būtų ir klaidų. Dėl programos lankstumo galimybės sudaryti savo nereikšmingų žodžių sąrašą kyla ir pliusų ir minusų: vartotojas gali lanksčiai rinktis nereikšmingų žodžių rinkinį, jį sudaryti pagal save, tačiau dirbant su didelės apimties tekstais gali būti sunku apibrėžti visus norimus žodžius ir gali užtrukti daug laiko.

# TESTAVIMO IŠVADOS

# 05 REZULTATŲ VIZUALIZAVIMAS

# REZULTATŲ VIZUALIZAVIMAS

Programoje realizuota galimybė pasirinkti išvesties kiekį - kiek dažniausiai pasikartojančių žodžių atspausdinti. Pagal tai sudaroma diagrama, turinti 5 skirtingas spalvas. Jeigu atvaizduojami daugiau negu 5 stulpeliai, spalvos paeiliui kartojamos.

```
# Create a data frame of the most common words
# Draw a bar chart
lst = word_counter.most_common(n_print)
df = pd.DataFrame(lst, columns = ['Word', 'Count'])
df.plot.bar(x='Word',y='Count', color=['C0', 'C1', 'C2', 'C3', 'C4'])
```