# IOrespuestas_2

May 3, 2022

## 1 Actividad con Datos Textuales

[522]:
```
from nltk.stem import WordNetLemmatizer
from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import matplotlib.pyplot as plt

import pandas as pd
pd.options.mode.chained_assignment = None  # default='warn'

df = pd.read_csv("C:/Users/User/Documents/Facu/9216/IOrespuestas.csv")
df
```

[522]:

| | Marca temporal | Dirección de correo electrónico | Padrón sin números | \ |
|---|---|---|---|---|
| 0 | 30/03/2022 17:49:38 | pcortif@fi.uba.ar | 100723 | |
| 1 | 30/03/2022 18:00:36 | mcirolini@fi.uba.ar | 97739 | |
| 2 | 30/03/2022 18:30:40 | rparedes@fi.uba.ar | 97920 | |
| 3 | 30/03/2022 19:20:14 | mledesma@fi.uba.ar | 102908 | |
| 4 | 30/03/2022 19:20:32 | tandrada@fi.uba.ar | 100586 | |
| .. | … | … | … | |
| 116 | 30/03/2022 21:52:35 | jrenovales@fi.uba.ar | 103787 | |
| 117 | 30/03/2022 23:07:20 | jovillamil@fi.uba.ar | 99825 | |
| 118 | 31/03/2022 12:00:41 | npesaresi@fi.uba.ar | 104911 | |
| 119 | 1/04/2022 17:30:14 | mjung@fi.uba.ar | 102939 | |
| 120 | 2/04/2022 9:55:29 | fcolotto@fi.uba.ar | 101455 | |

| | ¿qué es la Investigación Operativa? |
|---|---|
| 0 | Se basa en  la utilización de métodos analític… |
| 1 | Optimización y diseño de operaciones, control … |
| 2 | Aplicaciones de métodos operativos a la vida real |
| 3 | La optimización de las operaciones mediante el… |
| 4 | Una forma interdisciplinaria de resolver probl… |
| .. | … |
| 116 | Es el area que se encarga de aplicar distintos… |
| 117 | NaN |
| 118 | Aplicación de la ciencia moderna a problemas c… |

```
119  No pude completar la encuesta con el celular e…
120  El estudio de las operaciones. Como distribuir…

[121 rows x 4 columns]
```

[523]:
```
# SELLECIONO LAS COLUMNS QUE QUIERO Y LES CAMBIO EL NOMBRE

df2=df[['Padrón sin números', '¿qué es la Investigación Operativa?']]
df2.columns = ['padrón', 'texto']
df2
```

[523]:
```
     padrón                                              texto
0    100723  Se basa en  la utilización de métodos analític…
1     97739  Optimización y diseño de operaciones, control …
2     97920  Aplicaciones de métodos operativos a la vida real
3    102908  La optimización de las operaciones mediante el…
4    100586  Una forma interdisciplinaria de resolver probl…
..      …                                                  …
116  103787  Es el area que se encarga de aplicar distintos…
117   99825                                               NaN
118  104911  Aplicación de la ciencia moderna a problemas c…
119  102939  No pude completar la encuesta con el celular e…
120  101455  El estudio de las operaciones. Como distribuir…

[121 rows x 2 columns]
```

[524]:
```
#EJEMPLO

df2['texto'][0]
```

[524]:
```
'Se basa en  la utilización de métodos analíticos para ayudar a tomar mejores
decisiones'
```

## 2 Defino las funciones

[525]:
```
# TOKENIZO

def tokens(texto):
    return [ w for w in word_tokenize(str(texto).lower()) if w.isalpha()]
```

[526]:
```
tokens(df2['texto'][0])
```

[526]:
```
['se',
 'basa',
 'en',
 'la',
 'utilización',
```

```
 'de',
 'métodos',
 'analíticos',
 'para',
 'ayudar',
 'a',
 'tomar',
 'mejores',
 'decisiones']
```

[527]:
```
# REMUEVO LOS STOP WORDS EN ESPAÑOL

def no_stops(tokens):
    return [t for t in tokens if t not in stopwords.words('spanish')]
```

[528]:
```
no_stops(tokens(df2['texto'][0]))
```

[528]:
```
['basa',
 'utilización',
 'métodos',
 'analíticos',
 'ayudar',
 'tomar',
 'mejores',
 'decisiones']
```

[529]:
```
# LEMANTIZO

wordnet_lematizer = WordNetLemmatizer()
spanish_stemmer = SnowballStemmer('spanish')


def lematizador(texto):
    #return [wordnet_lematizer.lemmatize(t) for t in texto]
     return [ spanish_stemmer.stem(t) for t in texto ]
```

[530]:
```
lematizador(no_stops(tokens(df2['texto'][0])))
```

[530]:
```
['bas', 'utiliz', 'metod', 'analit', 'ayud', 'tom', 'mejor', 'decision']
```

[531]:
```
# JUNTO A TODAS LAS FUNCIONES EN UNA SOLA

funciones = [tokens,no_stops,lematizador]

def preparar(texto):
    tokens = texto
```

```
    for transformar in funciones:
        tokens = transformar(tokens)
    return tokens
```

[532]:
```
# APLICO LA FUNCION A LA COLUMNA TEXTO

df2['texto'] = df2['texto'].apply(preparar)
```

[533]:
```
df2
```

[533]:

| | padrón | texto |
|---|---|---|
| 0 | 100723 | [bas, utiliz, metod, analit, ayud, tom, mejor,… |
| 1 | 97739 | [optimiz, diseñ, oper, control, proces, indust… |
| 2 | 97920 | [aplic, metod, oper, vid, real] |
| 3 | 102908 | [optimiz, oper, mediant, uso, program] |
| 4 | 100586 | [form, interdisciplinari, resolv, problem, mul… |
| .. | … | … |
| 116 | 103787 | [are, encarg, aplic, distint, metod, oper, ayu… |
| 117 | 99825 | [nan] |
| 118 | 104911 | [aplic, cienci, modern, problem, complej, apar… |
| 119 | 102939 | [pud, complet, encuest, celul, dia, clas, dij,… |
| 120 | 101455 | [estudi, oper, distribu, recurs, maximiz, bene… |

[121 rows x 2 columns]

## 3 Análisis

[534]:
```
# DEFINICION DE WIKIPEDIA

definicion = "Es una disciplina que se ocupa de la aplicación de métodos␣
 ↪analíticos avanzados para ayudar a tomar mejores decisiones empleando␣
 ↪técnicas de otras ciencias matemáticas, como modelado matemático, análisis␣
 ↪estadístico y optimización, la investigación de operaciones llega a␣
 ↪soluciones óptimas o casi óptimas para problemas complejos de toma de␣
 ↪decisiones"
```

[535]:
```
# APLICO LA FUNCION PREPARAR A LA DEFINICION

definicion_preparada = preparar(definicion)
definicion_preparada
```

[535]:
```
['disciplin',
 'ocup',
 'aplic',
 'metod',
 'analit',
 'avanz',
```

```
 'ayud',
 'tom',
 'mejor',
 'decision',
 'emple',
 'tecnic',
 'cienci',
 'matemat',
 'model',
 'matemat',
 'analisis',
 'estadist',
 'optimiz',
 'investig',
 'oper',
 'lleg',
 'solucion',
 'optim',
 'casi',
 'optim',
 'problem',
 'complej',
 'tom',
 'decision']
```

[536]:
```
print(len(definicion_preparada))
```

```
30
```

[537]:
```
# EJEMPLO

df2['texto'][0]
```

[537]:
```
['bas', 'utiliz', 'metod', 'analit', 'ayud', 'tom', 'mejor', 'decision']
```

[538]:
```
def resultado(definicion_preparada, text):
    valor= 0
    for s in definicion_preparada:
        for t in text:
            if t in s:
                valor += 1
    return valor
```

[539]:
```
resultado(definicion_preparada, df2['texto'][0])
```

[539]:
```
8
```

[540]:
```
# APLICO LA FUNCION QUE CALCULA EL RESULTADO A UNA NUEVA COLUMNA

df2['Nota'] = df2.apply(lambda x: resultado(definicion_preparada, x['texto']),␣
 ↪axis =1 )
```

[558]:
```
df3 = df2[['padrón','Nota']]
df3.head(20)
```

[558]:

| | padrón | Nota |
|---|---|---|
| 0 | 100723 | 8 |
| 1 | 97739 | 2 |
| 2 | 97920 | 3 |
| 3 | 102908 | 2 |
| 4 | 100586 | 1 |
| 5 | 97920 | 2 |
| 6 | 102309 | 6 |
| 7 | 102120 | 2 |
| 8 | 101100 | 0 |
| 9 | 103803 | 11 |
| 10 | 104210 | 2 |
| 11 | 102408 | 2 |
| 12 | 103875 | 0 |
| 13 | 98667 | 0 |
| 14 | 102277 | 1 |
| 15 | 103858 | 2 |
| 16 | 102346 | 2 |
| 17 | 104102 | 1 |
| 18 | 90845 | 1 |
| 19 | 104122 | 4 |

[559]:
```
len(df3['Nota'])
```

[559]: 121

[560]:
```
df3.groupby('Nota')['Nota'].count().sort_values(ascending=False)
```

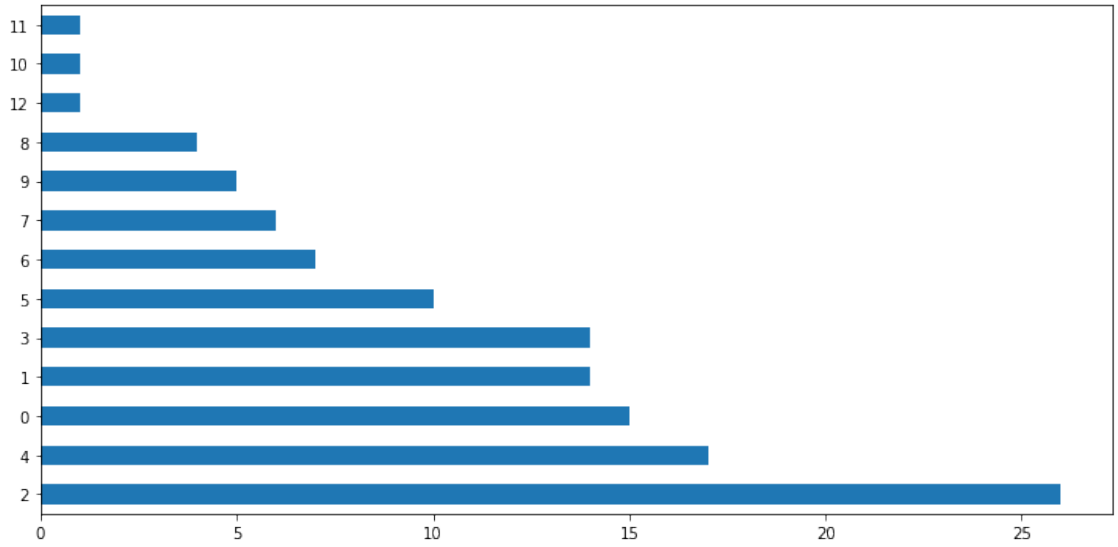
[560]:

| Nota | |
|---|---|
| 2 | 26 |
| 4 | 17 |
| 0 | 15 |
| 1 | 14 |
| 3 | 14 |
| 5 | 10 |
| 6 | 7 |
| 7 | 6 |
| 9 | 5 |
| 8 | 4 |

```
10     1
11     1
12     1
Name: Nota, dtype: int64
```

[562]:
```
fig,axis = plt.subplots(nrows=1,ncols=1, figsize=(10,5))
df3['Nota'].value_counts(ascending=True).plot(kind='barh',ax=axis)
axis.invert_yaxis()

fig.tight_layout()
```




## 4 Poner la nota

[563]:
```
df4=df3
```

[564]:
```
# Si tiene más de 5 palabras que coincidan aprueban

df4['Nota'] = df4['Nota'].apply(lambda x: 'Aprobado' if x > 4 else␣
 ↪'Desaprobado' )
```

[565]:
```
df4.head(20)
```

[565]:

| | padrón | Nota |
|---|---|---|
| 0 | 100723 | Aprobado |
| 1 | 97739 | Desaprobado |
| 2 | 97920 | Desaprobado |
| 3 | 102908 | Desaprobado |
| 4 | 100586 | Desaprobado |

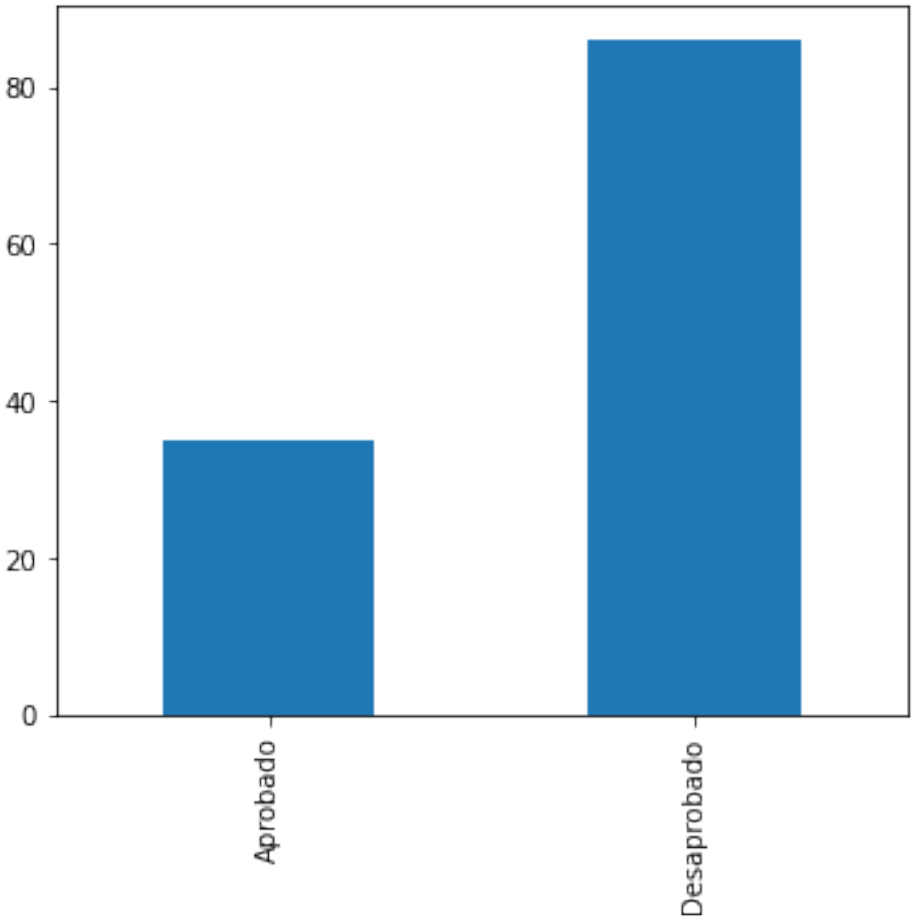
```
5    97920  Desaprobado
6   102309     Aprobado
7   102120  Desaprobado
8   101100  Desaprobado
9   103803     Aprobado
10  104210  Desaprobado
11  102408  Desaprobado
12  103875  Desaprobado
13   98667  Desaprobado
14  102277  Desaprobado
15  103858  Desaprobado
16  102346  Desaprobado
17  104102  Desaprobado
18   90845  Desaprobado
19  104122  Desaprobado
```

[566]:
```
df4.groupby('Nota')['Nota'].count().sort_values(ascending=False)
```

[566]:
```
Nota
Desaprobado    86
Aprobado       35
Name: Nota, dtype: int64
```

[569]:
```
fig,axis = plt.subplots(nrows=1,ncols=1, figsize=(5,5))
df3['Nota'].value_counts(ascending=True).plot(kind='bar',ax=axis)


fig.tight_layout()
```


80
60
40
20
0
Aprobado
Desaprobado

[ ]: