S2 - HOPE kNN_v2

November 30, 2020

0.1 Import data from DB.

```
[1]: import pandas as pd
     import numpy as np
```

```
[2]: dfOrg = pd.read_csv('hope_dataset_cleaned.csv')

     print(dfOrg.shape[0])
```

```
1243
```

```
[3]: dfOrg.head(10)
```

```
[3]:   pedido.data.attributes.age pedido.data.attributes.diagnostic_main  \
     0                       75.0                     FISTULA PERITONEAL
     1                       75.0                     FISTULA PERITONEAL
     2                       75.0                     FISTULA PERITONEAL
     3                       75.0                     FISTULA PERITONEAL
     4                       75.0                     FISTULA PERITONEAL
     5                       75.0                     FISTULA PERITONEAL
     6                       75.0                     FISTULA PERITONEAL
     7                       75.0                     FISTULA PERITONEAL
     8                       75.0                     FISTULA PERITONEAL
     9                       75.0                     FISTULA PERITONEAL

       pedido.data.attributes.gender  articulo  respuesta.articlesRevisedYear  \
     0                          male  27395425                           2018
     1                          male  28560554                           2018
     2                          male  28641726                           2017
     3                          male  26245344                           2016
     4                          male  28942543                           2018
     5                          male  24782153                           2014
     6                          male  28002229                           2018
     7                          male  27505109                           2017
     8                          male  24850546                           2015
     9                          male  29371050                           2019

       respuesta.articlesRevisedMonth  \
0                                1
1                                4
2                               12
3                               12
4                                6
5                                6
6                                9
7                                4
8                                1
9                                4

                              respuesta.pubmed_keys  utilidad
0  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       1.0
1  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
2  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
3  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
4  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
5  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
6  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
7  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
8  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
9  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
```

Remove “articulo” and “gender” to remove attributes without value

[4]:
```
dfOrg = dfOrg.drop([
    'pedido.data.attributes.gender',
    'articulo'
], axis=1)

dfOrg.head(10)
```

[4]:
```
   pedido.data.attributes.age pedido.data.attributes.diagnostic_main  \
0                        75.0                     FISTULA PERITONEAL
1                        75.0                     FISTULA PERITONEAL
2                        75.0                     FISTULA PERITONEAL
3                        75.0                     FISTULA PERITONEAL
4                        75.0                     FISTULA PERITONEAL
5                        75.0                     FISTULA PERITONEAL
6                        75.0                     FISTULA PERITONEAL
7                        75.0                     FISTULA PERITONEAL
8                        75.0                     FISTULA PERITONEAL
9                        75.0                     FISTULA PERITONEAL

   respuesta.articlesRevisedYear  respuesta.articlesRevisedMonth  \
0                           2018                               1
1                           2018                               4
```

```
2                           2017                              12
3                           2016                              12
4                           2018                               6
5                           2014                               6
6                           2018                               9
7                           2017                               4
8                           2015                               1
9                           2019                               4

                               respuesta.pubmed_keys  utilidad
0  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       1.0
1  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
2  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
3  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
4  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
5  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
6  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
7  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
8  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
9  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
```

Expand pubmed_keys attribute

[5]:

```
dfOrg['respuesta.pubmed_keys'] = dfOrg['respuesta.pubmed_keys'].apply(lambda x :
 ↪ str(x).split(','))

dfOrg = dfOrg.explode('respuesta.pubmed_keys').reset_index(drop=True)

dfOrg.head(10)
```

[5]:

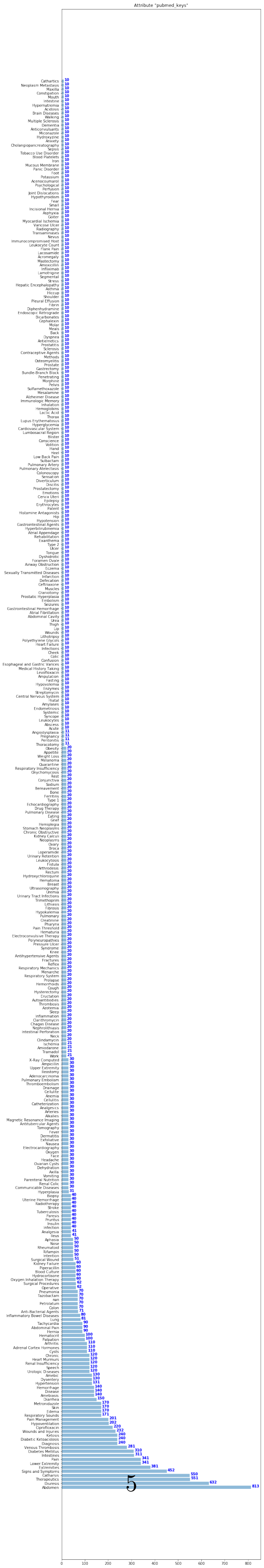
```
   pedido.data.attributes.age pedido.data.attributes.diagnostic_main  \
0                        75.0                    FISTULA PERITONEAL
1                        75.0                    FISTULA PERITONEAL
2                        75.0                    FISTULA PERITONEAL
3                        75.0                    FISTULA PERITONEAL
4                        75.0                    FISTULA PERITONEAL
5                        75.0                    FISTULA PERITONEAL
6                        75.0                    FISTULA PERITONEAL
7                        75.0                    FISTULA PERITONEAL
8                        75.0                    FISTULA PERITONEAL
9                        75.0                    FISTULA PERITONEAL

   respuesta.articlesRevisedYear  respuesta.articlesRevisedMonth  \
0                           2018                               1
1                           2018                               1
2                           2018                               1
3                           2018                               1
```

```
4                           2018                               1
5                           2018                               1
6                           2018                               1
7                           2018                               1
8                           2018                               1
9                           2018                               1

  respuesta.pubmed_keys  utilidad
0               Abdomen       1.0
1        Adenocarcinoma       1.0
2           Antiemetics       1.0
3         Blood Culture       1.0
4             Catharsis       1.0
5              Diuresis       1.0
6               Fistula       1.0
7           Gastrectomy       1.0
8     Incisional Hernia       1.0
9            Intestines       1.0
```

[6]:

```
import matplotlib.pyplot as plt

categoriesORGPubMedKeys = dfOrg['respuesta.pubmed_keys'].value_counts()

print("total: " + str(categoriesORGPubMedKeys.size))

y_values = np.arange(len(categoriesORGPubMedKeys.index))

plt.figure(figsize=(10,80))
plt.barh(y_values, categoriesORGPubMedKeys.values, align='center', alpha=0.5)
plt.yticks(y_values, categoriesORGPubMedKeys.index)

for i, v in enumerate(categoriesORGPubMedKeys.values):
    plt.text(v + 3, i, str(v), color='blue', fontweight='bold', fontsize=10)

plt.title('Attribute "pubmed_keys"')

plt.show()
```

```
total: 353
```


Attribute "pubmed_keys"
Cathartics 10
Neoplasm Metastasis 10
Maxilla 10
Constipation 10
Mouth 10
Intestine 10
Hypernatremia 10
Acidosis 10
Brain Diseases 10
Walking 10
Multiple Sclerosis 10
Dementia 10
Anticonvulsants 10
Miconazole 10
Hydroxyzine 10
Anxiety 10
Cholangiopancreatography 10
Sepsis 10
Tobacco Use Disorder 10
Blood Platelets 10
Iron 10
Mucous Membrane 10
Panic Disorder 10
Foot 10
Potassium 10
Acenocoumarol 10
Psychological 10
Perfusion 10
Joint Dislocations 10
Hypothyroidism 10
Fear 10
Small 10
Incisional Hernia 10
Asphyxia 10
Goiter 10
Myocardial Ischemia 10
Varicose Ulcer 10
Radiography 10
Transaminases 10
Nevus 10
Immunocompromised Host 10
Leukocyte Count 10
Flank Pain 10
Lacosamide 10
Acromegaly 10
Mastectomy 10
Amoxicillin 10
Infliximab 10
Lamotrigine 10
Segmental 10
Stress 10
Hepatic Encephalopathy 10
Asthma 10
Hiccup 10
Shoulder 10
Pleural Effusion 10
Fibrin 10
Diphenhydramine 10
Endoscopic Retrograde 10
Bicarbonates 10
Cephalexin 10
Molar 10
Meals 10
Back 10
Dyspnea 10
Antiemetics 10
Prostatitis 10
Sclerosis 10
Contraceptive Agents 10
Methods 10
Osteomyelitis 10
Prostate 10
Gastrectomy 10
Bundle-Branch Block 10
Penetrating 10
Morphine 10
Pelvis 10
Sulfamethoxazole 10
Mesalamine 10
Alzheimer Disease 10
Immunologic Memory 10
Inhalation 10
Hemoglobins 10
Lactic Acid 10
Thorax 10
Lupus Erythematosus 10
Hyperglycemia 10
Cardiovascular System 10
Lumbosacral Region 10
Blister 10
Conscience 10
Volition 10
Hand 10
Heel 10
Low Back Pain 10
Sulbactam 10
Pulmonary Artery 10
Pulmonary Atelectasis 10
Colonoscopy 10
Sensation 10
Diverticulum 10
Discitis 10
Prostatectomy 10
Emotions 10
Cervix Uteri 10
Epilepsy 10
Erythrocytes 10
Patent 10
Histamine Antagonists 10
Hip 10
Hypotension 10
Gastrointestinal Agents 10
Hyperbilirubinemia 10
Atrial Appendage 10
Rehabilitation 10
Exanthema 10
Type 2 10
Ulcer 10
Tongue 10
Dyshidrotic 10
Foramen Ovale 10
Airway Obstruction 10
Eczema 10
Sexually Transmitted Diseases 10
Infarction 10
Defecation 10
Ceftriaxone 10
Muscles 10
Craniotomy 10
Prostatic Hyperplasia 10
Embolism 10
Seizures 10
Gastrointestinal Hemorrhage 10
Atrial Fibrillation 10
Abdominal Cavity 10
Urea 10
Thigh 10
Lip 10
Wounds 10
Lithotripsy 10
Polyethylene Glycols 10
Heart Failure 10
Infections 10
Cheek 10
Colic 10
Confusion 10
Esophageal and Gastric Varices 10
Medical History Taking 10
Levofloxacin 10
Amputation 10
Fasting 10
Hypovolemia 10
Enzymes 10
Streptomycin 10
Central Nervous System 10
Hiatal 10
Amylases 10
Endometriosis 10
Systemic 10
Syncope 10
Leukocytes 10
Abscess 10
Acute 10
Angiodysplasia 11
Pregnancy 11
Peritonitis 11
Thoracotomy 11
Obesity 20
Appetite 20
Weight Loss 20
Melanoma 20
Quarantine 20
Respiratory Insufficiency 20
Onychomycosis 20
Rest 20
Conjunctiva 20
Sodium 20
Bereavement 20
Bone 20
Ferritins 20
Type 1 20
Echocardiography 20
Drug Therapy 20
Pulmonary Disease 20
Eating 20
Grief 20
Hemiplegia 20
Stomach Neoplasms 20
Chronic Obstructive 20
Kidney Calculi 20
Neoplasms 20
Ovary 20
Broca 20
Loperamide 20
Urinary Retention 20
Leukocytosis 20
Fistula 20
Arthrodesis 20
Rectum 20
Hydroxychloroquine 20
Hematoma 20
Breast 20
Ultrasonography 20
Uremia 20
Urinary Tract Infections 20
Trimethoprim 20
Lithiasis 20
Fibrosis 20
Hypokalemia 20
Pulmonary 20
Creatinine 20
Pharynx 20
Pain Threshold 20
Hematuria 20
Electroconvulsive Therapy 20
Polyneuropathies 20
Pressure Ulcer 20
Syndrome 20
Knee 20
Antihypertensive Agents 20
Fractures 20
Reflex 20
Respiratory Mechanics 20
Menarche 20
Respiratory System 20
Prolapse 20
Hemorrhoids 20
Cough 20
Hysterectomy 20
Eructation 20
Autoantibodies 20
Thrombosis 20
Azotemia 20
Sleep 20
Inflammation 20
Clarithromycin 20
Chagas Disease 20
Nephrolithiasis 20
Intestinal Perforation 20
Neck 20
Clindamycin 20
Ischemia 21
Amiodarone 21
Tramadol 21
Work 21
X-Ray Computed 30
Ampicillin 30
Upper Extremity 30
Ileostomy 30
Adenocarcinoma 30
Pulmonary Embolism 30
Thromboembolism 30
Drainage 30
Cellulite 30
Anemia 30
Cellulitis 30
Catheterization 30
Analgesics 30
Arteries 30
Alkalies 30
Magnetic Resonance Imaging 30
Antitubercular Agents 30
Tomography 30
Fever 30
Dermatitis 30
Exfoliative 30
Nausea 30
Electrocardiography 30
Oxygen 30
Face 30
Headache 30
Ovarian Cysts 30
Dehydration 30
Axilla 30
Vomiting 30
Parenteral Nutrition 30
Renal Colic 30
Communicable Diseases 30
Hyperplasia 31
Biopsy 40
Uterine Hemorrhage 40
Radiotherapy 40
Stroke 40
Tuberculosis 40
Paresis 40
Pruritus 40
Insulin 40
Infection 40
Analgesia 41
Ileus 41
Aphasia 50
Nose 50
Rheumatoid 50
Rifampin 50
Intention 50
Surgical Wound 51
Kidney Failure 60
Piperacillin 60
Blood Culture 60
Hydrocortisone 60
Oxygen Inhalation Therapy 60
Surgical Procedures 62
Operative 62
Pneumonia 70
Tazobactam 70
nan 70
Petrolatum 70
Colon 70
Anti-Bacterial Agents 71
Inflammatory Bowel Diseases 80
Lung 81
Tachycardia 90
Abdominal Pain 90
Hernia 90
Hematocrit 100
Palpation 100
Arthritis 110
Adrenal Cortex Hormones 110
Cysts 110
Chronic 120
Heart Murmurs 120
Renal Insufficiency 120
Speech 120
Urologic Diseases 120
Amebic 130
Dysentery 130
Hypertension 131
Hemorrhage 140
Disease 140
Amebiasis 140
Diarrhea 150
Metronidazole 170
Skin 170
Edema 170
Respiratory Sounds 171
Pain Management 201
Hypoventilation 202
Ciprofloxacin 220
Wounds and Injuries 232
Ketosis 240
Diabetic Ketoacidosis 240
Diagnosis 240
Venous Thrombosis 281
Diabetes Mellitus 310
Intestines 311
Pain 341
Lower Extremity 341
Extremities 381
Signs and Symptoms 452
Catharsis 550
Therapeutics 551
Diuresis 632
Abdomen 813
0
100
200
300
400
500
600
700
800

0.2 Transform (factorice) from Categories to continuous atributes

Transform ‘pedido.data.attributes.diagnostic_main’ atribute

[7]:
```
dfKNN = dfOrg

categoriesORGDiagnosticMain = dfKNN['pedido.data.attributes.diagnostic_main'].
 ↪value_counts()

print("total: " + str(categoriesORGDiagnosticMain.size))

categoriesORGDiagnosticMain
```

```
total: 31
```

[7]:
```
INFECCION DE PARTES BLANDAS    3270
DOLOR ABDOMINAL                2137
CETOACIDOSIS DIABETICA         1430
REHABILITACION NEUROLOGICA     1050
INSUFICIENCIA RESPIRATORIA      910
FISTULA PERITONEAL              770
REACCION ALERGICA               660
DIFICULTAD RESPIRATORIA         550
INFECCION URINARIA              470
DISNEA                          430
SINDROME FEBRIL                 390
LEGRADO                         360
CEFALEA INTENSA                 320
NEUMONIA                        320
ACV.ISQUEMICO                   310
INSUFICIENCIA CARDIACA          310
TEP                             250
PROLAPSO                        200
METRORRAGIA                     170
DIABETES                        160
ANEMIA                          150
HEMORRAGIA DIGESTIVA            140
ABDOMEN AGUDO                   121
ARTRITIS SEPTICA                120
POLITRAUMATISMO                 110
TORACOTOMIA                     110
LUXACION COLUMNA CERVICAL       100
CA GASTRICO                      90
DOLOR                            90
ADENOMA DE PROSTATA              40
```

```
DERMOLIPECTOMIA                    40
Name: pedido.data.attributes.diagnostic_main, dtype: int64
```

[8]:

```
dataDiagnosticMain, categoriesDiagnosticMain = pd.factorize(dfKNN['pedido.data.
 ↪attributes.diagnostic_main'])

dfKNN['pedido.data.attributes.diagnostic_main'] = dataDiagnosticMain
```

Transform ‘respuesta.pubmed_keys’ atribute

[9]:

```
categoriesORGPubMedKeys = dfKNN['respuesta.pubmed_keys'].value_counts()

print("total: " + str(categoriesORGPubMedKeys.size))
```

```
total: 353
```

[10]:

```
dataPubMedKeys, categoriesPubMedKeys = pd.factorize(dfKNN['respuesta.
 ↪pubmed_keys'])

dfKNN['respuesta.pubmed_keys'] = dataPubMedKeys
```

[11]:

```
dfKNN.head(10)
```

[11]:

```
   pedido.data.attributes.age  pedido.data.attributes.diagnostic_main  \
0                        75.0                                        0
1                        75.0                                        0
2                        75.0                                        0
3                        75.0                                        0
4                        75.0                                        0
5                        75.0                                        0
6                        75.0                                        0
7                        75.0                                        0
8                        75.0                                        0
9                        75.0                                        0

   respuesta.articlesRevisedYear  respuesta.articlesRevisedMonth  \
0                           2018                               1
1                           2018                               1
2                           2018                               1
3                           2018                               1
4                           2018                               1
5                           2018                               1
6                           2018                               1
7                           2018                               1
8                           2018                               1
9                           2018                               1
```

```
   respuesta.pubmed_keys  utilidad
0                      0       1.0
1                      1       1.0
2                      2       1.0
3                      3       1.0
4                      4       1.0
5                      5       1.0
6                      6       1.0
7                      7       1.0
8                      8       1.0
9                      9       1.0
```

[12]:

```
print("age NaN => " + str(dfKNN[pd.isnull(dfKNN['pedido.data.attributes.age'])].shape[0]))
print("diagnostic_main NaN => " + str(dfKNN[pd.isnull(dfKNN['pedido.data.attributes.diagnostic_main'])].shape[0]))
print("articlesRevisedYear NaN => " + str(dfKNN[pd.isnull(dfKNN['respuesta.articlesRevisedYear'])].shape[0]))
print("articlesRevisedMonth NaN => " + str(dfKNN[pd.isnull(dfKNN['respuesta.articlesRevisedMonth'])].shape[0]))
print("pubmed_keys NaN => " + str(dfKNN[pd.isnull(dfKNN['respuesta.pubmed_keys'])].shape[0]))
print("utilidad NaN => " + str(dfKNN[pd.isnull(dfKNN['utilidad'])].shape[0]))
```

```
age NaN => 10
diagnostic_main NaN => 0
articlesRevisedYear NaN => 0
articlesRevisedMonth NaN => 0
pubmed_keys NaN => 0
utilidad NaN => 14758
```

Remove row with age eq NaN

[13]:

```
dfKNN = dfKNN[pd.notnull(dfKNN['pedido.data.attributes.age'])]
```

0.3 Separe data by utilidad is defined

[14]:

```
dfDataSetComplete = dfKNN[pd.notnull(dfKNN['utilidad'])]

print(dfDataSetComplete.shape[0])

dfDataSetToPredict = dfKNN[pd.isnull(dfKNN['utilidad'])]

print(dfDataSetToPredict.shape[0])
```

```
830
14748
```

```
[15]: dfDataSetComplete.head(10)
```

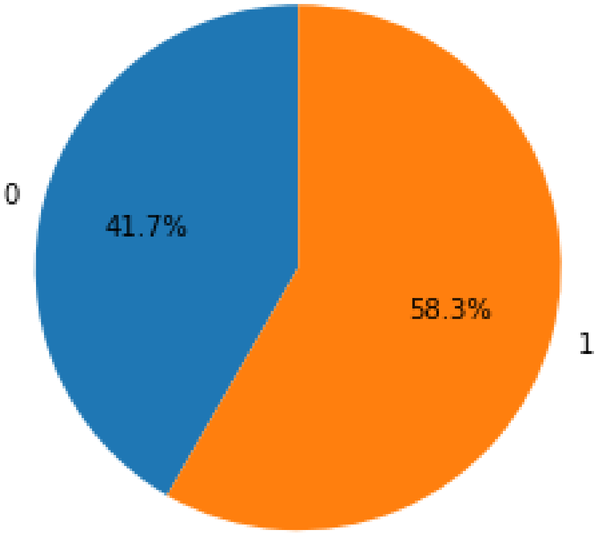
```
[15]:    pedido.data.attributes.age  pedido.data.attributes.diagnostic_main  \
      0                        75.0                                       0
      1                        75.0                                       0
      2                        75.0                                       0
      3                        75.0                                       0
      4                        75.0                                       0
      5                        75.0                                       0
      6                        75.0                                       0
      7                        75.0                                       0
      8                        75.0                                       0
      9                        75.0                                       0

         respuesta.articlesRevisedYear  respuesta.articlesRevisedMonth  \
      0                           2018                               1
      1                           2018                               1
      2                           2018                               1
      3                           2018                               1
      4                           2018                               1
      5                           2018                               1
      6                           2018                               1
      7                           2018                               1
      8                           2018                               1
      9                           2018                               1

         respuesta.pubmed_keys  utilidad
      0                      0       1.0
      1                      1       1.0
      2                      2       1.0
      3                      3       1.0
      4                      4       1.0
      5                      5       1.0
      6                      6       1.0
      7                      7       1.0
      8                      8       1.0
      9                      9       1.0
```

0.4 Check distribution of “utilidad” attribute

```
[16]: utilityValues = dfDataSetComplete['utilidad'].value_counts()

      print(utilityValues)

      import matplotlib.pyplot as plt

      labels = '0', '1'
```

```
sizes = [utilityValues.get(0.0), utilityValues.get(1.0)]

fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
ax1.axis('equal')

plt.show()
```

```
1.0    484
0.0    346
Name: utilidad, dtype: int64
```



0.5 k-NN

[17]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt
```

[18]:
```
X = dfDataSetComplete[['pedido.data.attributes.diagnostic_main',
       'respuesta.articlesRevisedYear',
       'respuesta.articlesRevisedMonth',
       'respuesta.pubmed_keys']].values

y = dfDataSetComplete['utilidad'].values
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

[19]:

```
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[20]:

```
k_range = range(1, 20)
accuracy_weights_uniform = []
error_weights_uniform = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors = k, weights='uniform', n_jobs=4)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    accuracy_weights_uniform.append(knn.score(X_test, y_test))
    error_weights_uniform.append(np.mean(y_pred != y_test))
```

[21]:

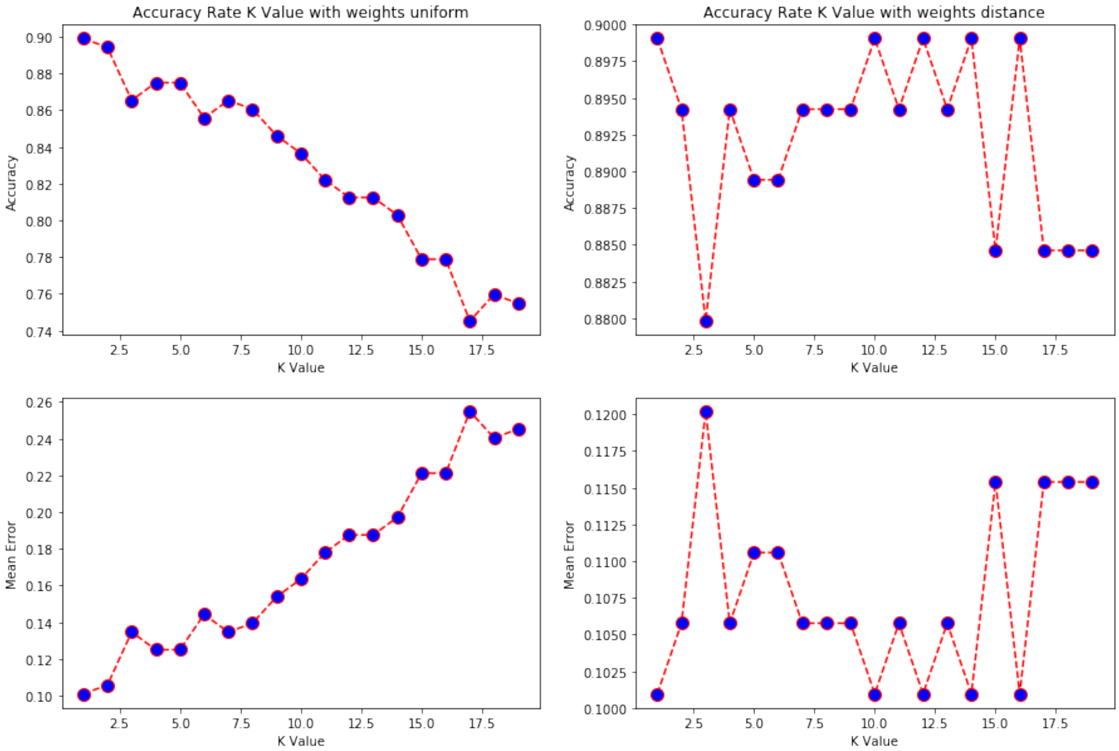
```
k_range = range(1, 20)
accuracy_weights_distance = []
error_weights_distance = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors = k, weights='distance', n_jobs=4)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    accuracy_weights_distance.append(knn.score(X_test, y_test))
    error_weights_distance.append(np.mean(y_pred != y_test))
```

[22]:

```
fig, axs = plt.subplots(2, 2,figsize=(15, 10))
axs[0, 0].plot(range(1, 20), accuracy_weights_uniform, color='red',␣
 ↪linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
axs[0, 0].set_title('Accuracy Rate K Value with weights uniform')
axs[0, 0].set_xlabel('K Value')
axs[0, 0].set_ylabel('Accuracy')
axs[0, 1].plot(range(1, 20), accuracy_weights_distance, color='red',␣
 ↪linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
axs[0, 1].set_title('Accuracy Rate K Value with weights distance')
axs[0, 1].set_xlabel('K Value')
axs[0, 1].set_ylabel('Accuracy')
axs[1, 0].plot(range(1, 20), error_weights_uniform, color='red',␣
 ↪linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
axs[1, 0].set_xlabel('K Value')
axs[1, 0].set_ylabel('Mean Error')
axs[1, 1].plot(range(1, 20), error_weights_distance, color='red',␣
 ↪linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
```

```
axs[1, 1].set_xlabel('K Value')
axs[1, 1].set_ylabel('Mean Error')
```

[22]:
```
Text(0, 0.5, 'Mean Error')
```



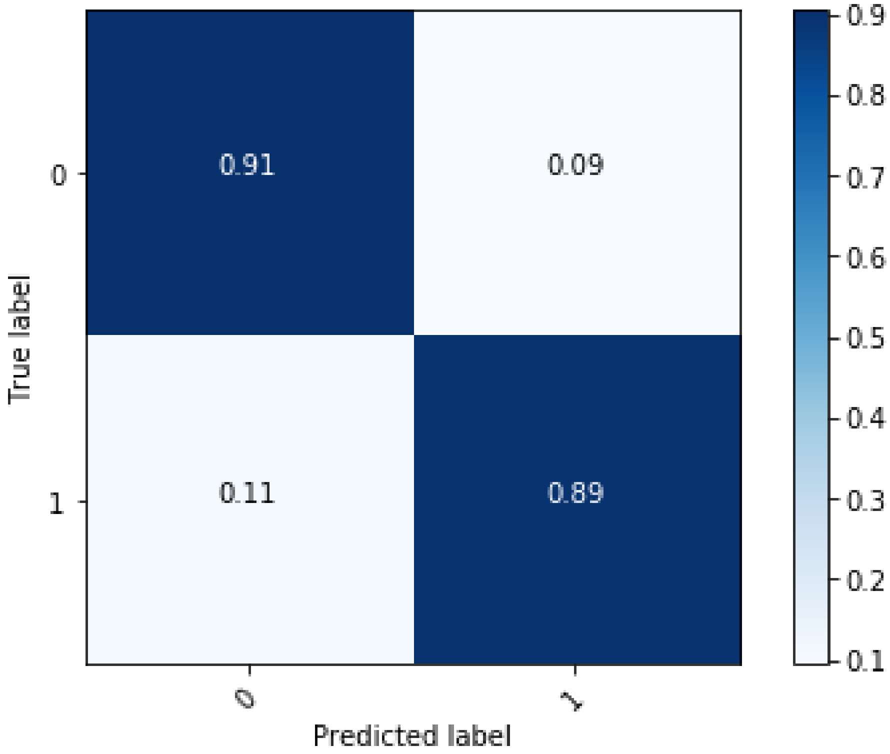
[23]:
```
n_neighbors = 1

knn = KNeighborsClassifier(n_neighbors, weights='uniform', n_jobs=4)
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(X_test, y_test)))
```

```
Accuracy of K-NN classifier on training set: 0.95
Accuracy of K-NN classifier on test set: 0.90
```

Show confusion matrix:

[24]:
```
import itertools
from sklearn.metrics import confusion_matrix

preds = knn.predict(X_test)
cnf_matrix = confusion_matrix(y_test, preds)
```

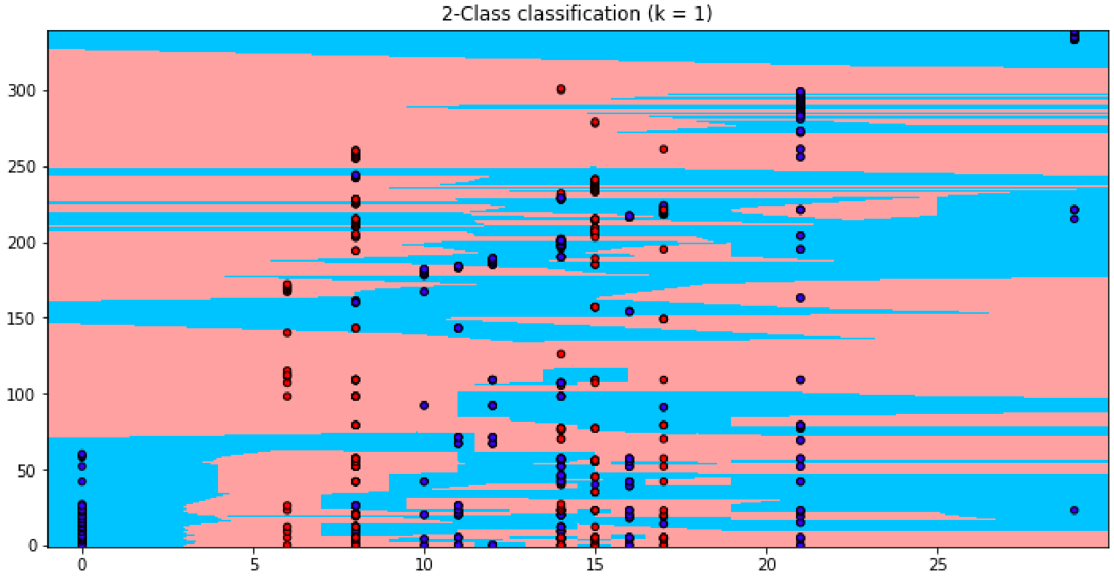
```
def plot_confusion_matrix(cm, classes):
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    cmap=plt.cm.Blues

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], ".2f"),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

n_classes=["0","1"]
plot_confusion_matrix(cnf_matrix, classes=n_classes)
```



0.6 Print the K-NN classification only with the attributes "diagnostic_main" and "pubmed_keys"

[25]:

```
X_plot = dfDataSetComplete[['pedido.data.attributes.diagnostic_main',
       'respuesta.pubmed_keys']].values
y_plot = dfDataSetComplete['utilidad'].values

h = .02  # step size in the mesh

# Create color maps
cmap_light = ListedColormap(['#ffa1a1', '#00c4ff'])
cmap_bold = ListedColormap(['#ff0000', '#3a00ff'])

# we create an instance of Neighbours Classifier and fit the data.
clf = KNeighborsClassifier(n_neighbors)
clf.fit(X_plot, y_plot)

# Plot the decision boundary. For that, we will assign a color to each
# point in the mesh [x_min, x_max]x[y_min, y_max].
x_min, x_max = X_plot[:, 0].min() - 1, X_plot[:, 0].max() + 1
y_min, y_max = X_plot[:, 1].min() - 1, X_plot[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))
Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])

# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(figsize=(12, 6))
plt.pcolormesh(xx, yy, Z, cmap=cmap_light)

# Plot also the training points
plt.scatter(X_plot[:, 0], X_plot[:, 1], c=y_plot, cmap=cmap_bold,
            edgecolor='k', s=20)
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.title("2-Class classification (k = 1)")

plt.show()
```


2-Class classification (k = 1)

0.7 Run Prediction

[26]:

```
def runPrediction(row):
    valuesrow = np.array([row.get(['pedido.data.attributes.diagnostic_main',
       'respuesta.articlesRevisedYear',
       'respuesta.articlesRevisedMonth',
       'respuesta.pubmed_keys']).values])
    return knn.predict(valuesrow)

dfDataSetToPredict.apply(runPrediction, axis=1)
```

[26]:

```
28         [0.0]
29         [1.0]
30         [1.0]
31         [1.0]
32         [1.0]
           …
15583      [1.0]
15584      [1.0]
15585      [1.0]
15586      [1.0]
15587      [1.0]
Length: 14748, dtype: object
```

[ ]: