Tugas Praktikum Topik Dalam Pengenalan Pola

Membaca artikel yang sudah diupload pada newlms terkait dengan naïve bayes. Silahkan pilih artikel yang Anda sukai. Telusuri data yang terdapat pada artsikel tersebut. Coba untuk melakukan kembali pada salah satu data yang memang tersedia dan buat report dan terapkan dengan mengguankan Naïve Bayes.

Nama Mahasiswa : Rangga Pebrianto

NIM : G6601231006

Laporan Keseluruhan:

Eksperimen ini mengeksplorasi algoritma machine learning, bernama Multinomial Naive Bayes Classifier, dengan predikto sebagai variabel boolean yaitu 0 dan 1 untuk mendeteksi berita palsu. penelitian sebelumnya pada artikel yang saya pilih dengan judul "Performance of bernoulli's naive bayes classifier in the detection of fake news" telah diterapkan pada penelitian sebelumnya menggunakan Gaussian Naive Bayes dengan akurasi 72%, dan Bernoulli Naive Bayes 83%, dan percobaan yang dilakukan saat ini saya menggunakan Multinomial Naive Bayes dengan akurasi cukup baik yaitu 90%.


```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory
import re
import os
for dirname, _, filenames in os.walk('/content/drive'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.

from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer, HashingVectorizer

from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

from sklearn.feature_extraction.text import CountVectorizer

import matplotlib.pyplot as plt
```

```
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Agro-Maritim Presisi/Dr. Yeni Herdiyeni S.Si., M.Kom./Topik dalam Agro-Mar
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Tugas/Tugas 1.pdf
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Tugas/Hasil LKP_Modul Praktikum Pengenalan Pola
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Tugas/TUGAS PRAKTIKUM PENGOLAHAN DASAR TEKS - Co
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Tugas/TUGAS PRAKTIKUM PENGOLAHAN DASAR CITRA - C
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Tugas/Hasil_LKP_Modul_Praktikum_Pengenalan_Pola_
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Tugas/TUGAS_PRAKTIKUM_PENGOLAHAN_DASAR_TEKS.ipyn
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Tugas/TUGAS_PRAKTIKUM_PENGOLAHAN_DASAR_CITRA.ipy
/content/drive/MyDrive/Bissmillah Semester 1/Topik dalam Pengenalan Pola/Praktikum/Mini Project/Beras.zip
/content/drive/MyDrive/Bissmillah Semester 1/Filsafat Sains/Prof. Dr. Ir. Asep Saefuddin M.Sc./Tugas Filsafat Sains a.n. Rangga Peb
/content/drive/MyDrive/Bissmillah Semester 1/Bahasa Inggris untuk Doktor/Sertifikat Bahasa Inggris.pdf
/content/drive/MyDrive/Bissmillah Semester 1/Bahasa Inggris untuk Doktor/Lolos Placement Test dari Klaim sertifikat kemampuan bahas
/content/drive/MyDrive/Colab Notebooks/rangga.jpg
/content/drive/MyDrive/Colab Notebooks/ipb.png
/content/drive/MyDrive/Colab Notebooks/Data.csv
/content/drive/MyDrive/Colab Notebooks/Data.txt
/content/drive/MyDrive/Colab Notebooks/web_traffic.tsv
/content/drive/MyDrive/Colab Notebooks/5.png
/content/drive/MyDrive/Colab Notebooks/babon.jpg
/content/drive/MyDrive/Colab Notebooks/TUGAS PRAKTIKUM PENGOLAHAN DASAR TEKS.ipynb
/content/drive/MyDrive/Colab Notebooks/TUGAS PRAKTIKUM PENGOLAHAN DASAR CITRA
/content/drive/MyDrive/Colab Notebooks/Hasil LKP_Modul Praktikum Pengenalan Pola Pertemuan 1 & 2
/content/drive/MyDrive/Colab Notebooks/tips.csv
/content/drive/MyDrive/Colab Notebooks/LKP_3_Data_Visualization.ipynb
/content/drive/MyDrive/Colab Notebooks/kashmiri-prediction-with-convolutional-neural-network.ipynb
/content/drive/MyDrive/Colab Notebooks/prediction-with-convolutional-neural-network.ipynb
/content/drive/MyDrive/Colab Notebooks/Untitled0.ipynb
/content/drive/MyDrive/Colab Notebooks/Untitled1.ipynb
```

```
/content/drive/MyDrive/Colab Notebooks/LKP 4
/content/drive/MyDrive/Colab Notebooks/fake-news-classifier-with-naive-bayes.ipynb
/content/drive/MyDrive/Tugas LKP 4/Modul Praktikum Pengenalan Pola Pertemuan - 4 (Python) (1).docx
/content/drive/MyDrive/Tugas LKP 4/Kumpulan Artikel Naive Bayes-20230913.zip
/content/drive/MyDrive/Tugas LKP 4/Iris.csv
/content/drive/MyDrive/Tugas LKP 4/NaiveBayes.xlsx
/content/drive/MyDrive/Tugas LKP 4/submit.csv
/content/drive/MyDrive/Tugas LKP 4/test.csv
/content/drive/MyDrive/Tugas LKP 4/train.csv
```

```
df =pd.read_csv('../content/drive/My Drive/Tugas LKP 4/train.csv')
```

```
df.head()
```

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get	Consortiumnews.com	Why the Truth Might Get You Fired October	1

```
x= df.drop('label',axis=1)
```

```
x.head(2)
```

	id	title	author	text
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...
1	1	FLYNN: Hillary Clinton, Big	Daniel J.	Ever get the feeling your life

```
y = df['label']
```

```
df.shape
```

```
(20800, 5)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      20800 non-null  int64 
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

Check any Null Values in the dataframe

```
df.isnull().sum()
```

```
id          0
title     558
author   1957
text       39
label       0
dtype: int64
```

```
df=df.dropna()
```

```
df.head()
```

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1

```
df['title'][3]
```

```
'15 Civilians Killed In Single US Airstrike Have Been Identified'
```

```
messeges =df.copy()


messeges.reset_index(inplace=True)


messeges.head()
```

	index	id	title	author	text	label
0	0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0

```
# Install NLTK stopwords (jalankan hanya sekali)
import nltk
nltk.download('stopwords')

# Import library
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

corpus = []
for i in range(0, len(messeges)):
    review = re.sub('[^a-zA-Z]', ' ', messeges['title'][i])
    review = review.lower()
    review = review.split()

    review = [ps.stem(word) for word in review if not word in stopwords.words('english')]
    review = ' '.join(review)
    corpus.append(review)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
corpus[6]
```

Counter Vectorization

Bag of Words

```
cv = CountVectorizer(max_features=5000,ngram_range=(1,3))
X = cv.fit_transform(corpus).toarray()


# show resulting vocabulary; the numbers are not counts, they are the position in the sparse vector.
cv.vocabulary_
```

```
 activ' : 40,
 'leak email': 2430,
 'art': 234,
 'paul': 3164,
 'perfect': 3205,
 'paul ryan': 3167,
 'right breitbart': 3679,
 'player': 3258,
 'stop': 4165,
 'peopl': 3194,
 'commit': 834,
 'suicid': 4216,
 'japan': 2237,
 'california': 554,
 'tale': 4288,
 'divers': 1203,
 'sourc': 4057,
 'welcom': 4856,
 'envoy': 1413,
 'deni': 1118,
 'past': 3149,
 'inaugur': 2115,
 'day new': 1044,
 'day new york': 1045,
 'offer': 3015,
 'clue': 781,
 'blind': 427,
 'predict': 3342,
 'bad news': 300,
 'news trump': 2923,
 'total': 4432,
 'snowden': 4026,
 'realiti': 3532,
 'winner': 4902,
 'nsa': 2965,
 'somalia': 4043,
 'escal': 1432,
 'shadow': 3917,
 'war new': 4791,
 'war new york': 4792,
 'free': 1720,
 'care': 597,
 'bless': 426,
 ...}
```

```
X.shape
```

```
(18285, 5000)
```

```
y=messeges['label']
```

```
## Divide the dataset into Train and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=0)
```

```
feature_names = cv.get_feature_names_out()
print(feature_names[:20])
```

```
['abandon' 'abc' 'abc news' 'abduct' 'abe' 'abedin' 'abl' 'abort' 'abroad'
 'absolut' 'abstain' 'absurd' 'abus' 'abus new' 'abus new york' 'academi'
 'accept' 'access' 'access pipelin' 'access pipelin protest']
```

```
cv.get_params()
```

```
{'analyzer': 'word',
 'binary': False,
 'decode_error': 'strict',
 'dtype': numpy.int64,
 'encoding': 'utf-8',
 'input': 'content',
 'lowercase': True,
 'max_df': 1.0,
 'max_features': 5000,
 'min_df': 1,
 'ngram_range': (1, 3),
 'preprocessor': None,
 'stop_words': None,
 'strip_accents': None,
 'token_pattern': '(?u)\\b\\w\\w+\\b',
 'tokenizer': None,
 'vocabulary': None}
```

```
count_df = pd.DataFrame(X_train, columns=cv.get_feature_names_out())
```

```
count_df.head()
```

	abandon	abc	abc news	abduct	abe	abedin	abl	abort	abroad	absolut	...	zero
0	0	0	0	0	0	0	0	0	0	0	...	0
1	0	0	0	0	0	0	0	0	0	0	...	0
2	0	0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	1	...	0

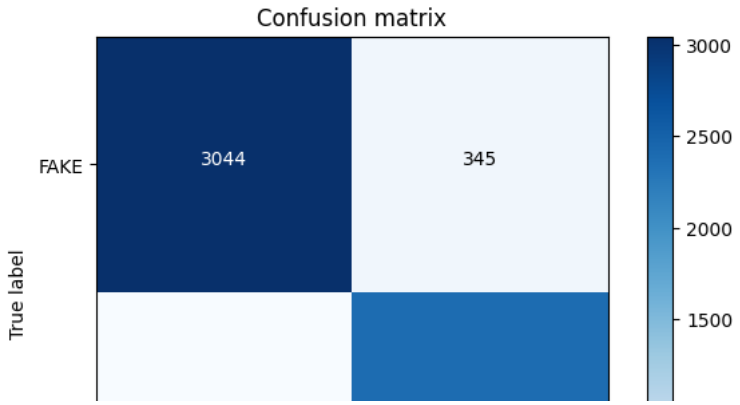
```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    See full source and example:
    http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

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

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

Mulinomial Naive Bayes Theorem

```
from sklearn.naive_bayes import MultinomialNB
classifier=MultinomialNB()
from sklearn import metrics
import numpy as np
import itertools
```

```
classifier.fit(X_train, y_train)
pred = classifier.predict(X_test)
score = metrics.accuracy_score(y_test, pred)
print("accuracy:   %0.3f" % score)
cm = metrics.confusion_matrix(y_test, pred)
plot_confusion_matrix(cm, classes=['FAKE', 'REAL'])
```

```
accuracy:   0.902
Confusion matrix, without normalization
```




```
classifier.fit(X_train, y_train)
pred = classifier.predict(X_test)
score = metrics.accuracy_score(y_test, pred)
score
```

```
0.9019055509527755
```

```
y_train.shape
```

```
(12250,)
```

Multinomial Classifier with Hyperparameter

```
classifier=MultinomialNB(alpha=0.1)
```

```
previous_score=0
for alpha in np.arange(0,1,0.1):
    sub_classifier=MultinomialNB(alpha=alpha)
    sub_classifier.fit(X_train,y_train)
    y_pred=sub_classifier.predict(X_test)
    score = metrics.accuracy_score(y_test, y_pred)
    if score>previous_score:
        classifier=sub_classifier
    print("Alpha: {}, Score : {}".format(alpha,score))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/naive_bayes.py:629: FutureWarning: The default value for `force_alpha` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/naive_bayes.py:635: UserWarning: alpha too small will result in numeric errors, set
  warnings.warn(
Alpha: 0.0, Score : 0.8903065451532726
Alpha: 0.1, Score : 0.9020712510356255
Alpha: 0.2, Score : 0.9025683512841757
Alpha: 0.30000000000000004, Score : 0.9024026512013256
Alpha: 0.4, Score : 0.9017398508699255
Alpha: 0.5, Score : 0.9015741507870754
Alpha: 0.6000000000000001, Score : 0.9022369511184756
Alpha: 0.7000000000000001, Score : 0.9025683512841757
Alpha: 0.8, Score : 0.9015741507870754
Alpha: 0.9, Score : 0.9017398508699255
```

```
## Get Features names
feature_names = cv.get_feature_names_out()
```

```
import numpy as np

# Mendapatkan bobot fitur dari model Multinomial Naive Bayes
feature_names = cv.get_feature_names_out()
coefficients = np.exp(classifier.feature_log_prob_[1])  # Ganti [1] dengan kelas yang sesuai

# Urutkan bobot fitur berdasarkan nilai tertinggi
sorted_coefficients = sorted(list(zip(feature_names, coefficients)), key=lambda x: x[1], reverse=True)

# Tampilkan 20 fitur dengan bobot tertinggi
top_20_features = sorted_coefficients[:20]
print(top_20_features)
```

```
[('trump', 0.018312907193949624), ('hillari', 0.013734110893434785), ('clinton', 0.012321745865415295), ('elect', 0.007446808510638
```

```
# Probabilitas posterior untuk setiap fitur dalam kelas pertama
probabilities = classifier.feature_log_prob_[0]

# Membuat daftar berisi pasangan fitur dan probabilitasnya
feature_probabilities = list(zip(feature_names, probabilities))

# Mengurutkan berdasarkan probabilitasnya (most real)
most_real_features = sorted(feature_probabilities, key=lambda x: x[1], reverse=True)[:20]
print(most_real_features)
```

```
[('new', -2.9468577463990755), ('time', -2.994219848520549), ('york', -3.000566240637532), ('new york', -3.0008020674474167), ('new
```

```
# Probabilitas log untuk setiap fitur dalam kelas pertama (fake)
log_probabilities = classifier.feature_log_prob_[0]

# Membuat daftar berisi pasangan fitur dan probabilitas log-nya
feature_log_probabilities = list(zip(feature_names, log_probabilities))

# Mengurutkan berdasarkan probabilitas log (most fake)
most_fake_features = sorted(feature_log_probabilities, key=lambda x: x[1])[:5000]
print(most_fake_features)
```

```
[('access pipelin protest', -11.458457546147459), ('acknowledg emf', -11.458457546147459), ('acknowledg emf damag', -11.45845754614
```

```
train=pd.read_csv('../content/drive/My Drive/Tugas LKP 4/train.csv')
test=pd.read_csv('../content/drive/My Drive/Tugas LKP 4/test.csv')
test.info()
test['label']='t'
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5200 entries, 0 to 5199
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      5200 non-null   int64 
 1   title   5078 non-null   object
 2   author  4697 non-null   object
 3   text    5193 non-null   object
dtypes: int64(1), object(3)
memory usage: 162.6+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      20800 non-null  int64 
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

```
from sklearn.feature_extraction.text import TfidfTransformer

test=test.fillna(' ')
train=train.fillna(' ')
test['total']=test['title']+' '+test['author']+test['text']
train['total']=train['title']+' '+train['author']+train['text']

#tfidf
transformer = TfidfTransformer(smooth_idf=False)
count_vectorizer = CountVectorizer(ngram_range=(1, 2))
counts = count_vectorizer.fit_transform(train['total'].values)
tfidf = transformer.fit_transform(counts)
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import TfidfTransformer

#data prep
test=test.fillna(' ')
train=train.fillna(' ')
test['total']=test['title']+' '+test['author']+test['text']
train['total']=train['title']+' '+train['author']+train['text']
```

```
#tfidf
transformer = TfidfTransformer(smooth_idf=False)
count_vectorizer = CountVectorizer(ngram_range=(1, 2))
counts = count_vectorizer.fit_transform(train['total'].values)
tfidf = transformer.fit_transform(counts)


targets = train['label'].values
test_counts = count_vectorizer.transform(test['total'].values)
test_tfidf = transformer.fit_transform(test_counts)

#split in samples
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(tfidf, targets, random_state=0)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/feature_extraction/text.py:1682: RuntimeWarning: divide by zero encountered in divi
  idf = np.log(n_samples / df) + 1
```

```
from sklearn.ensemble import (RandomForestClassifier, ExtraTreesClassifier,
                              AdaBoostClassifier)

Extr = ExtraTreesClassifier(n_estimators=5,n_jobs=4)
Extr.fit(X_train, y_train)
print('Accuracy of ExtrTrees classifier on training set: {:.2f}'
     .format(Extr.score(X_train, y_train)))
print('Accuracy of Extratrees classifier on test set: {:.2f}'
     .format(Extr.score(X_test, y_test)))
```

```
Accuracy of ExtrTrees classifier on training set: 1.00
Accuracy of Extratrees classifier on test set: 0.85
```

```
from sklearn.naive_bayes import MultinomialNB

NB = MultinomialNB()
NB.fit(X_train, y_train)
print('Accuracy of NB  classifier on training set: {:.2f}'
     .format(NB.score(X_train, y_train)))
print('Accuracy of NB classifier on test set: {:.2f}'
     .format(NB.score(X_test, y_test)))
```

```
Accuracy of NB  classifier on training set: 0.88
Accuracy of NB classifier on test set: 0.78
```

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(C=1e5)
logreg.fit(X_train, y_train)
print('Accuracy of Lasso classifier on training set: {:.2f}'
     .format(logreg.score(X_train, y_train)))
print('Accuracy of Lasso classifier on test set: {:.2f}'
     .format(logreg.score(X_test, y_test)))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy of Lasso classifier on training set: 1.00
Accuracy of Lasso classifier on test set: 0.98
```

```
targets = train['label'].values
logreg = LogisticRegression()
logreg.fit(counts, targets)

example_counts = count_vectorizer.transform(test['total'].values)
predictions = logreg.predict(example_counts)
pred=pd.DataFrame(predictions,columns=['label'])
pred['id']=test['id']
pred.groupby('label').count()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: Con
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

1 to 2 of 2 entries Filter

label:
to

id:
to

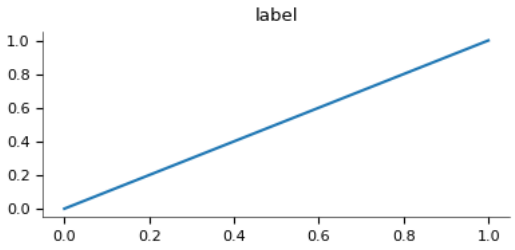
Search by all fields:

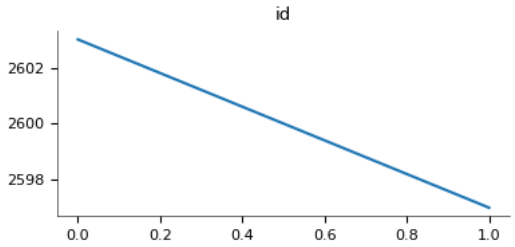
label	id
0	2603
1	2597

Show 25 per page

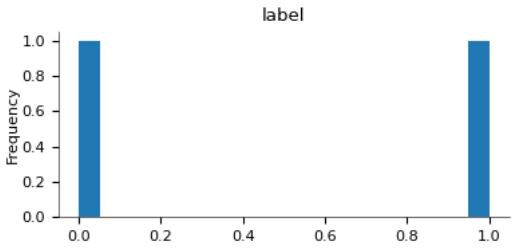
Like what you see? Visit the data table notebook to learn more about interactive tables.

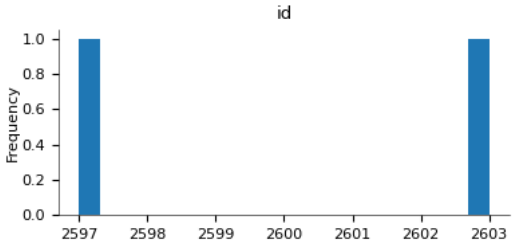
Values





Distributions





2-d distributions

```
pred.to_csv('countvect5.csv', index=False)
```

✓ 0s completed at 5:58 PM