

Fake News

Fake News Detector

Tri Wahyu Setiawan

Deskripsi Masalah

Fake news / Hoax / Berita palsu adalah informasi yang menyesatkan dan disampaikan atau disajikan seolah-olah benar.

Disinformasi ini dapat muncul dalam berbagai bentuk, termasuk artikel tertulis, gambar, video, atau unggahan media sosial, dan seringkali disebarkan melalui platform online dan jejaring media sosial.

Tujuan dibuat ini adalah :

1. Sensasionalisme : mencari perhatian
2. Manipulasi politik : menyebarkan informasi palsu lawan politik
3. Hoaks : memang digunakan untuk hiburan / lelucon
4. Mencari keuntungan : mendapatkan klik di medsos

Tujuan dari project ini adalah untuk membuat machine learning yang bisa memprediksi berita real dan berita fake/ hoaks dan bisa digunakan oleh masyarakat, sebagai media penyaring informasi.



Mengumpulkan Data

Sumber dataset :

<https://www.kaggle.com/code/rajatkumar30/fake-news-prediction-92-5-accuracy/notebook>

```
[ ] df = pd.read_csv('news.csv', encoding = 'latin-1')
df
```

	Unnamed: 0		title	text	label
0	8476		You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...		Google Pinterest Digg LinkedIn Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy		U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	â Kaydee King (@KaydeeKing) November 9, 2016...		FAKE
4	875	The Battle of New York: Why This Primary Matters		It's primary day in New York and front-runners...	REAL
...
6330	4490	State Department says it can't find emails fro...		The State Department told the Republican Natio...	REAL
6331	8062	The âPâ in PBS Should Stand for âPlutocr...		The âPâ in PBS Should Stand for âPlutocr...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...		Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...		ADDIS ABABA, Ethiopia âPresident Obama conve...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...		Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL



Menelaah Data

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 6335 entries, 0 to 6334  
Data columns (total 4 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   Unnamed: 0   6335 non-null   int64  
1   title        6335 non-null   object  
2   text         6335 non-null   object  
3   label        6335 non-null   object  
dtypes: int64(1), object(3)  
memory usage: 198.1+ KB
```

Dari syntax yang digunakan didapatkan :

kolom title, text dan label mempunyai tipe data object



Memvalidasi Data

```
df.shape
```

```
(6335, 4)
```

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

```
0
```

```
Unnamed: 0 0
```

```
title 0
```

```
text 0
```

```
label 0
```

Kesimpulan :

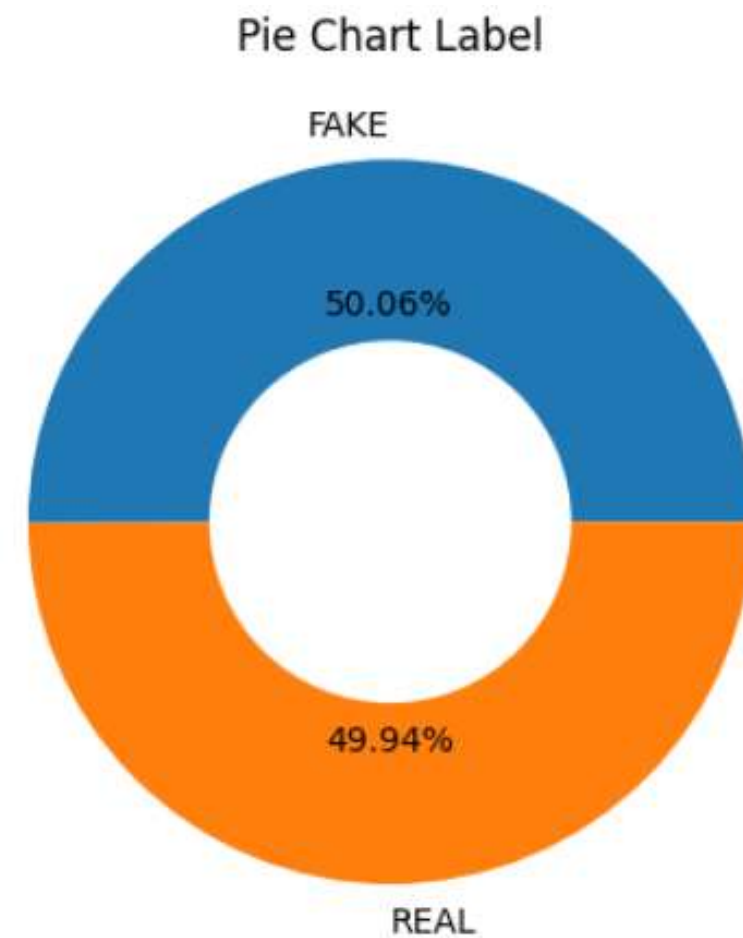
Data terdiri dari 4 kolom dan 6334 baris

Tidak ada data yang kosong



Hasil pengecekan struktur data :

1. Tidak ada data yang kosong
2. Tipe data dakolom title, text dan label sesuai (object)
3. Variable target (kolom label) *balance* antara data fake dan data real



Menentukan Objek Data

Tahapan :

1. Menggabungkan data dari kolom title dan text
2. Hanya menggunakan kolom label dan kolom text yang baru

```
] #menggabungkan data dari title dan text  
df['text2'] = df['title'] + ' ' + df['text']  
df['text2'].head(5)
```

```
#df yang digunakan hanya dari kolom label dan text2  
df = df[['label', 'text2']]  
df.head(5)
```

Output data yang dihasilkan :

	label	text2
0	FAKE	You Can Smell Hillary's Fear Daniel Greenfie...
1	FAKE	Watch The Exact Moment Paul Ryan Committed Pol...
2	REAL	Kerry to go to Paris in gesture of sympathy U....
3	FAKE	Bernie supporters on Twitter erupt in anger ag...
4	REAL	The Battle of New York: Why This Primary Matte...



Membersihkan Data

1. Clean Text

Tujuan :

Menghilangkan simbol, angka, tanda baca, dan mengganti huruf besar menjadi huruf kecil

```
import re
import nltk

#membuat fungsi
def clean (Text) :
    sms = re.sub('[^a-zA-Z]', ' ', Text) #menghilangkan semua yang non abjad dibatasi dengan spasi
    sms = sms.lower() #mengganti ke huruf kecil semua
    sms = sms.split()
    sms = ' '.join(sms)
    return sms
```

```
df['text_clean'] = df['text2'].apply(clean)
df
```

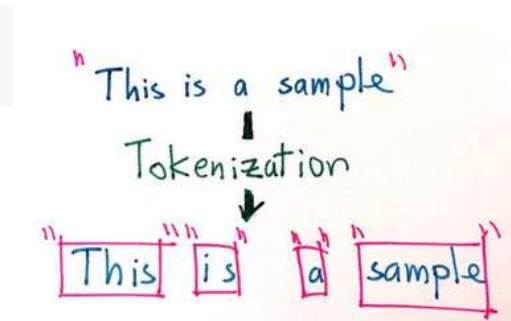
	label	text2	text_clean
0	FAKE	You Can Smell Hillary's Fear Daniel Greenfie...	you can smell hillary s fear daniel greenfield...
1	FAKE	Watch The Exact Moment Paul Ryan Committed Pol...	watch the exact moment paul ryan committed pol...
2	REAL	Kerry to go to Paris in gesture of sympathy U....	kerry to go to paris in gesture of sympathy u ...
3	FAKE	Bernie supporters on Twitter erupt in anger ag...	bernie supporters on twitter erupt in anger ag...
4	REAL	The Battle of New York: Why This Primary Matte...	the battle of new york why this primary matter...

Mengkonstruksi Data

2. Text Tokenization : memecah kalimat menjadi potongan kata-kata individu

```
nltk.download('punkt_tab')
```

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



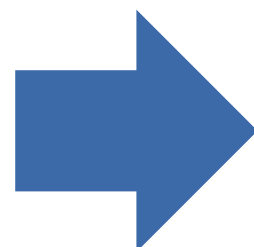
```
df['text_tokenize'] = df.apply(lambda row : nltk.word_tokenize(row['text_clean']), axis=1)
```

3. Menghilangkan stopwords (and, all, am, is dll)

```
nltk.download('stopwords')
```



```
from nltk.corpus import stopwords  
print(stopwords.words('english'))
```



```
#membuat fungsi untuk menghilangkan stopwords  
def remove_stopwords(Text) :  
    stop_words = set(stopwords.words('english'))  
    filtered_text = [word for word in Text if word not in stop_words]  
    return filtered_text
```

```
df['nostopword_text'] = df['text_tokenize'].apply(remove_stopwords)
```


Mengkonstruksi Data

4. Lemmatization : mengubah kata menjadi kata dasarnya

```
nltk.download('wordnet')
```

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

```
from nltk.stem import WordNetLemmatizer
```

```
lemmatizer = WordNetLemmatizer()
```

```
def lemmatize_word(text):  
    lemmas = [lemmatizer.lemmatize(word, pos='v') for word in text]  
    return lemmas
```

```
df["text_lematized"] = df["nostopword_text"].apply(lemmatize_word)
```

better → **good**

5. Create corpus

```
#Creating a corpus of text feature to encode further into vectorized form  
corpus_text= []  
for row in df["text_lematized"]:  
    msg = ' '.join(row)  
    corpus_text.append(msg)
```


Menentukan Label Data

- Mengubah text menjadi vektor (mengubah teks menjadi angka sehingga bisa diproses oleh algoritma mesin learning)

```
#Menggunakan TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

#Mengubah text menjadi angka
# X adalah corpus_text dan kolom text_lematized
tfidf = TfidfVectorizer(stop_words='english', max_df=0.7)
X = tfidf.fit_transform(corpus_text).toarray()
#Featurnya menjadi :
X.dtype
```



- Label Encoder

```
from sklearn.preprocessing import LabelEncoder

#Label encode the Target and use it as y
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(df["label"])
```

Original Data:			Label Encoded Data:		
Color	Size	Price	Color	Size	Price
Blue	L	100	0	0	100
Green	M	150	1	1	150
Red	S	200	2	2	200
Green	XL	120	1	3	120
Red	M	180	2	1	180

Membangun Model

Pembagian data train (80%) dan data test (20%)

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Passive Aggressive Classifiers

```
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
model_pac = PassiveAggressiveClassifier(max_iter=10)
model_pac.fit(X_train, y_train)
```

```
#melakukan prediksi pada data test
y_pred_pac = model_pac.predict(X_test)
```

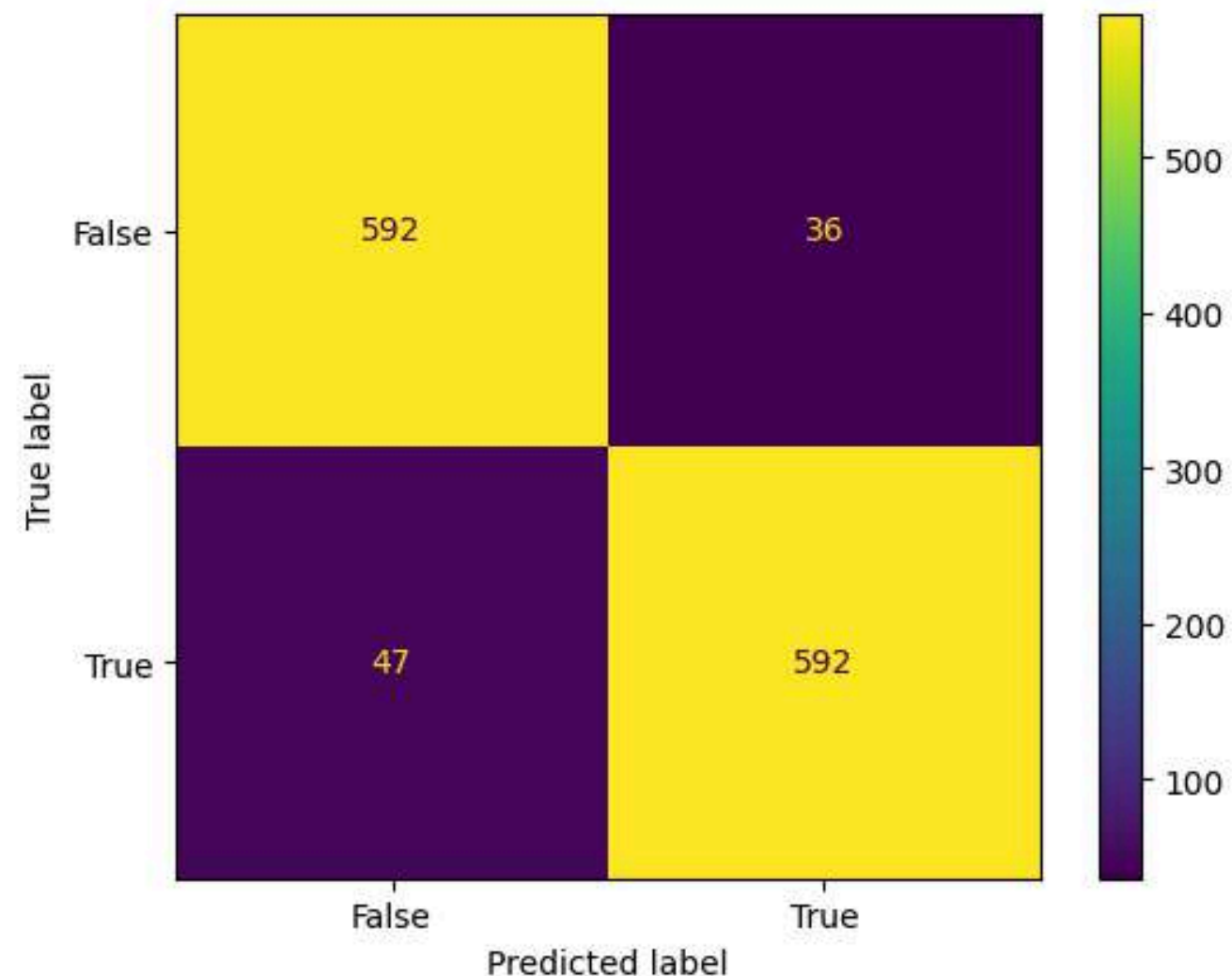

Evaluasi Hasil Model

```
#menghitung akurasi
accuracy_pac = accuracy_score(y_test,y_pred_pac)
print('akurasi', accuracy_pac)
```

akurasi 0.9368587213891081

```
#menghitung classification report
classification_report = classification_report(y_test, y_pred_pac, target_names=label_encoder.classes_)
print ('classification report', classification_report)
```

classification report	precision	recall	f1-score	support
FAKE	0.93	0.94	0.94	628
REAL	0.94	0.93	0.94	639
accuracy		0.94		1267
macro avg	0.94	0.94	0.94	1267
weighted avg	0.94	0.94	0.94	1267



Dari perhitungan confusion matrix didapatkan :

False negatif = 47

False positif = 36

Selisih antara False negatif dan False positif tidak terlalu tinggi → OK

Akurasi Model Passive Aggressive Classifiers tertinggi di antara model yang lain.

Membangun Model

Multinomial NB

```
from sklearn.naive_bayes import MultinomialNB  
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
model_mnb = MultinomialNB()
```

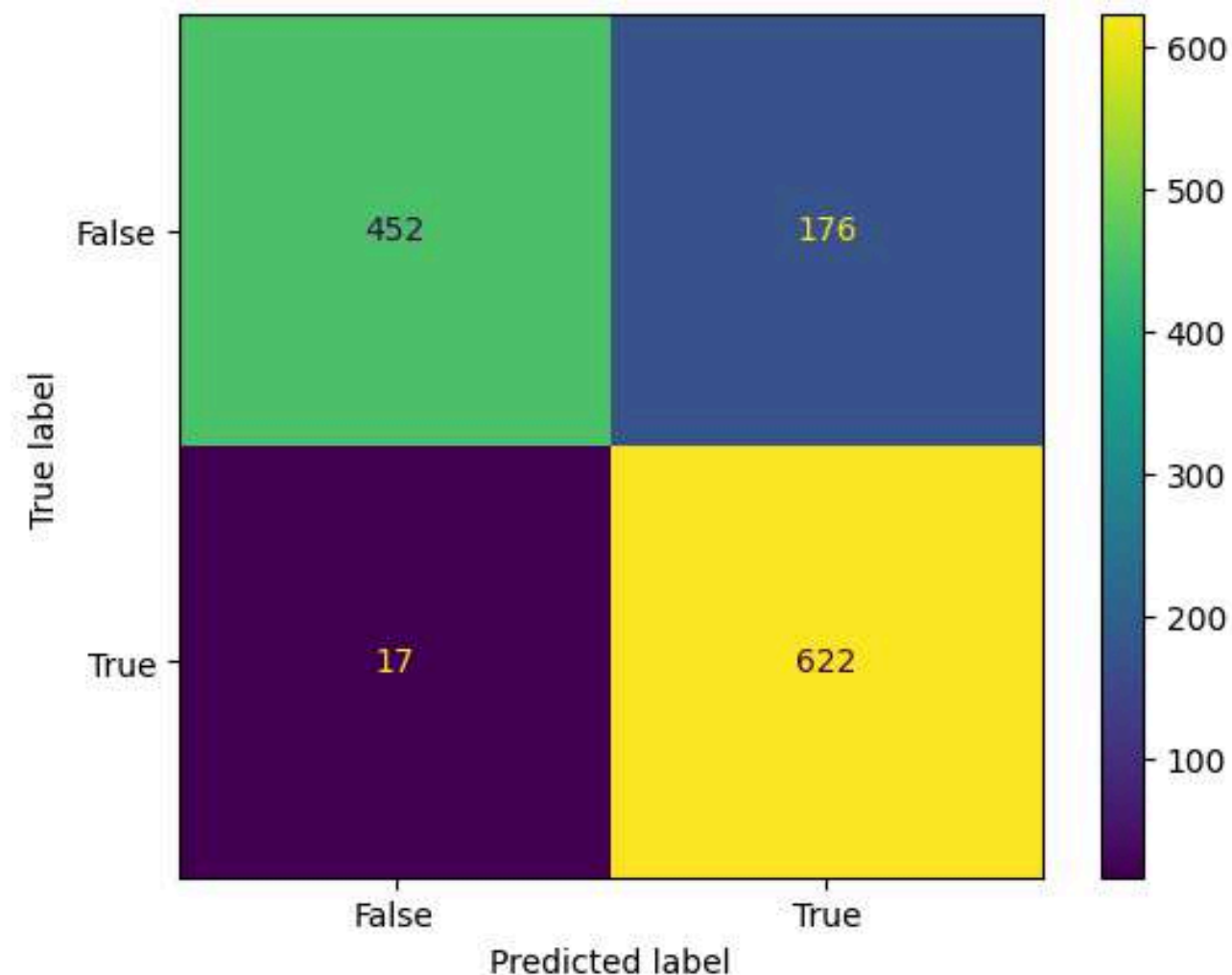
```
model_mnb.fit(X_train, y_train)  
print('pelatihan selesai')
```

```
#melakukan prediksi pada data test  
y_pred_mnb = model_mnb.predict(X_test)
```

Evaluasi Hasil Model

```
#menghitung classification report
classification_report = classification_report(y_test, y_pred_mnb, target_names=label_encoder.classes_)
print ('classification report mnb', classification_report)
```

classification report mnb	precision	recall	f1-score	support
FAKE	0.96	0.72	0.82	628
REAL	0.78	0.97	0.87	639
accuracy			0.85	1267
macro avg	0.87	0.85	0.84	1267
weighted avg	0.87	0.85	0.85	1267



Dari perhitungan confusion matrix didapatkan :

False negatif = 17

False positif = 176

False positif > false negatif → OK

Akurasi lebih rendah dari Passive Aggressive Classifiers

Membangun Model

Random Forest

```
| from sklearn.ensemble import RandomForestClassifier  
| from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
| model_rf = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
| model_rf.fit(X_train, y_train)  
| print('pelatihan selesai')
```

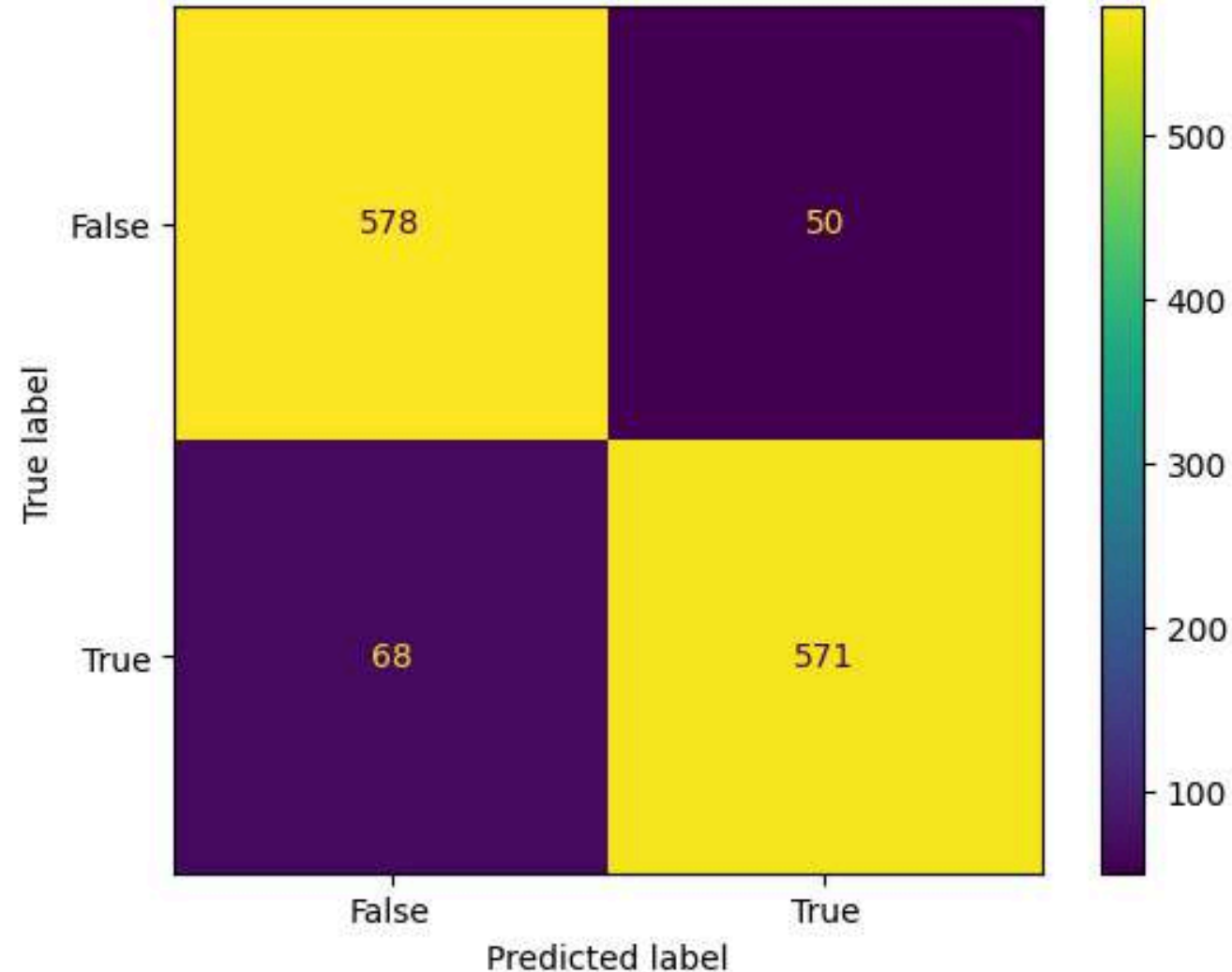
pelatihan selesai

```
| #melakukan prediksi pada data test  
| y_pred_rf = model_rf.predict(X_test)
```

Evaluasi Hasil Model

```
#menghitung classification report
classification_report = classification_report(y_test, y_pred_rf, target_names=label_encoder.classes_)
print ('classification report rf', classification_report)
```

classification report rf	precision	recall	f1-score	support
FAKE	0.89	0.92	0.91	628
REAL	0.92	0.89	0.91	639
accuracy			0.91	1267
macro avg	0.91	0.91	0.91	1267
weighted avg	0.91	0.91	0.91	1267



Dari perhitungan confusion matrix didapatkan :

False negatif = 68

False positif = 50

False negatif > false positif dengan selisih tidak terlalu tinggi → OK

Akurasi lebih rendah dari Passive Aggressive Classifiers

Membangun Model

Support Vector Machine



```
from sklearn.svm import SVC # Import model SVM
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
model_svm = SVC(kernel='linear')
```

```
model_svm.fit(X_train, y_train)
```

▼ SVC ⓘ ?

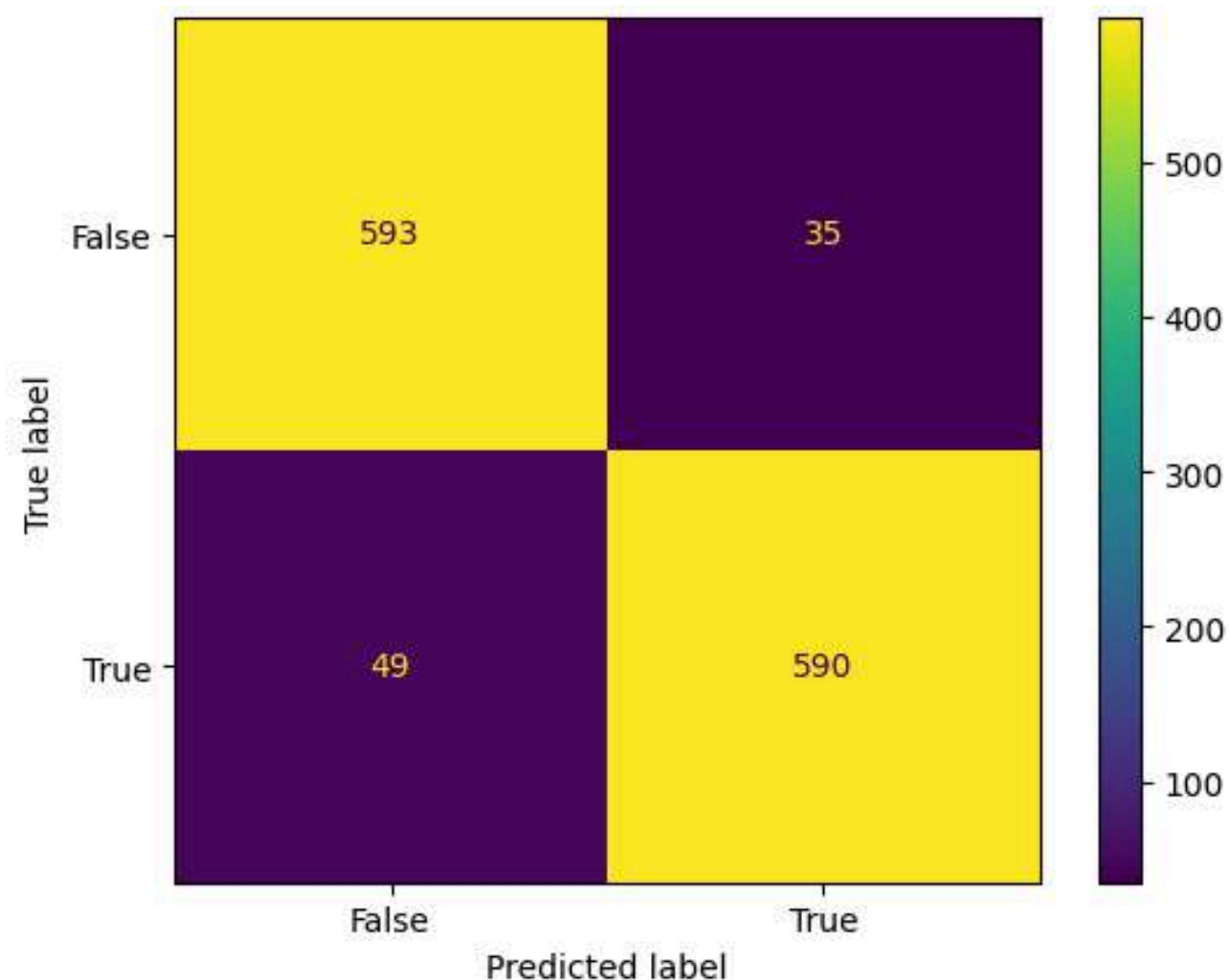
```
SVC(kernel='linear')
```

```
#melakukan prediksi pada data test
y_pred_svm = model_svm.predict(X_test)
```

Evaluasi Hasil Model

```
#menghitung classification report
classification_report = classification_report(y_test, y_pred_svm, target_names=label_encoder.classes_)
print ('classification report svm', classification_report)
```

classification report svm	precision	recall	f1-score	support
FAKE	0.92	0.94	0.93	628
REAL	0.94	0.92	0.93	639
accuracy	0.93			1267
macro avg	0.93	0.93	0.93	1267
weighted avg	0.93	0.93	0.93	1267



Dari perhitungan classification report dan confusion matrix didapatkan :

Accuracy = 93%

False negatif = 35

False positif = 49

False positif > False negatif → OK

Akurasi lebih rendah 1% dari Passive Aggressive Classifiers

Note : waktu yang diperlukan untuk training model paling lama dibandingkan model yang lain

Fake News

Thank you

NEWS

D