

Oral Presentation

Good morning everyone!

Today I will present the results of an academic study focused on Fake News detection, using **Machine Learning** techniques and algorithms.

1. Introduction

Fake News are false information, usually spread on social networks, creating serious impacts. Since spreading fake news is easy, we need methods to help identify what is true or not.

2. Objective

The aim of this work was to analyze different algorithms to automatically detect Fake News, seeking the most efficient model.

3. Methodology

We collected many news articles (both fake and true) from Kaggle. We cleaned, standardized, and shuffled the data. We visually analyzed them with WordCloud and split into training and testing sets.

4. Algorithms Tested

We used: Logistic Regression, Decision Tree, Random Forest, SVM, KNN.

Each model “learns” to identify fake news.

5. Results

Four models had accuracy above 90%. Decision Tree was the best. KNN made more mistakes.

We analyzed accuracy, precision, sensitivity, specificity.

6. Limitations and Future

Difficulty finding datasets in Portuguese.
It's necessary to test more algorithms, expand datasets, and validate better.

7. Conclusion

ML is powerful for detecting Fake News and protecting society!

8. Code

This code installs the libraries we will use. They are like toolboxes for programming.

```
```python
!pip install wordcloud seaborn nltk
```
```

We import the tools we just installed. This way, we can use their superpowers in the code.

```
```python
import pandas as pd
import numpy as np
import string
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,
confusion_matrix
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
```
```

Here we load the fake and real news we will study. Imagine two different books!

```
```python
fake = pd.read_csv('Fake.csv') # Upload 'Fake.csv' to Colab
true = pd.read_csv('True.csv') # Upload 'True.csv' to Colab
fake['target'] = 1 # Fake News
true['target'] = 0 # True News
```
```

We put everything together in a big table and shuffle it like a deck of cards! We remove title and date to focus only on the text.

```
```python
data = pd.concat([fake, true], ignore_index=True)
data = data.sample(frac=1).reset_index(drop=True)
data.drop(['title', 'date'], axis=1, inplace=True)
```
```

Let's make words simpler: lowercase only, no punctuation or words that don't help, like "a, the, of".

```
```python
stop_words = set(stopwords.words('portuguese'))
def clean_text(text):
 text = str(text).lower()
 text = text.translate(str.maketrans('', '',
string.punctuation))
 text = " ".join([word for word in text.split() if word not in
stop_words])
 return text

data['text'] = data['text'].apply(clean_text)
```
```

See a pretty drawing with the words that show up most in the news, the word cloud.

```
```python
wc = WordCloud(width=800, height=400,
background_color='white').generate(' '.join(data['text']))
plt.figure(figsize=(10, 5))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```
--
```

We turn texts into numbers that computers understand and split them into training and testing data for the computer to learn and practice.

```
```python
X = data['text']
y = data['target']
vectorizer = TfidfVectorizer(max_features=5000)
X_vect = vectorizer.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_vect, y,
test_size=0.2, random_state=42)
```
--
```

Now comes the magic! We will train five smart machines to tell if news is fake or real.

```
```python
modelos = {
 'Logistic Regression': LogisticRegression(),
 'Decision Tree': DecisionTreeClassifier(),
 'Random Forest': RandomForestClassifier(),
 'SVM': SVC(),
 'KNN': KNeighborsClassifier()
}
for nome, modelo in modelos.items():
 print(f'\n--- {nome} ---')
 modelo.fit(X_train, y_train)
 y_pred = modelo.predict(X_test)
 print('Confusion Matrix:')
 print(confusion_matrix(y_test, y_pred))
 sns.heatmap(confusion_matrix(y_test, y_pred), annot=True,
fmt='d', cmap='Blues')
 plt.title(f'Confusion Matrix - {nome}')
 plt.xlabel('Predicted')
 plt.ylabel('Actual')
 plt.show()
 print('Classification Report:')

```

```
... print(classification_report(y_test, y_pred))
```

```
--
```

***If you want to save a trained machine to use later, just save it!***

```
```python
# from joblib import dump, load
# dump(modelos['Decision Tree'], 'decision_tree_model.joblib')
```
```

```
--
```

**\*\*Conclusion\*\***

Machine Learning proved promising for detecting Fake News. With research, we can build a safer world for everyone!