Instituto Tecnológico campus Culiacán

Inteligencia Artificial

Carrera:

Ingeniería en Sistemas Computacionales

Actividad:

Clasificador de Spam

Alumno:

21170293 Fernando Chiquete Velazquez

21170387 Omar Manjarrez Rodelo

Maestr@:

Zuriel Dathan Mora Felix

01/05/2025

Clasificador de Spam con Naïve Bayes

Este proyecto implementa un clasificador de spam utilizando Naïve Bayes Multinomial y técnicas de Procesamiento de Lenguaje Natural para detectar correos electrónicos no deseados.

Se utiliza el dataset de SpamAssassin.

Código

```
# Cargar datos
data = pd.read_csv("spam_assassin.csv")
# Preprocesamiento
data["text"] = data["text"].str.lower()
data["text"] = data["text"].str.replace("[^a-zA-Z0-9]", " ", regex=True)
data["text"] = data["text"].str.replace("\s+", " ", regex=True)
data["text"] = data["text"].str.strip()
# === Análisis de palabras y peso de repetición ===
word_counts = []
unique_word_counts = []
repetition_weights = []

for text in data["text"]:
    words = text.split()
    total_words = len(words)
    word_counts.append(total_words)

    counter = Counter(words)
    unique_words = len(counter)
    unique_word_counts.append(unique_words)

    # Peso por repetición: (total - únicas) / total
    repetition_weight = (total_words - unique_words) / total_words if total_words > 0 else 0
    repetition_weights.append(repetition_weight)
```

Lo primero que realizamos es la separación de las palabras para calcular el peso de cada una de ellas y determinar la frecuencia de cada una de ellas.

```
# Vectorización
vectorizer = TfidfVectorizer(stop_words="english")
features = vectorizer.fit_transform(data["text"])
results = data["target"]

# Dividir datos en entrenamiento y prueba
features_train, features_test, results_train, results_test = train_test_split(
    features, results, test_size=0.2, random_state=42
)
data_train, data_test = train_test_split(data, test_size=0.2, random_state=42)
```

Aquí lo que hacemos es vectorizar el texto para analizar y determinar si son palabras realmente en ingles, además de indicarle cual será el tamaño del entrenamiento y cual será el tamaño de las pruebas

```python
# Modelo de Naive Bayes
model = MultinomialNB()
model.fit(features_train, results_train)

# Predicción
results_pred = model.predict(features_test)

# Evaluación del modelo
print("Precisión del modelo: ", accuracy_score(results_test, results_pred))
# Identificar correos clasificados como spam
spam_indices = [i for i, pred in enumerate(results_test) if pred == 1]
print("Índices de correos clasificados como spam:")
print(spam_indices)
# Imprimir solo los pesos de repetición de esos correos
print("\nPesos de repetición de los correos clasificados como SPAM:")
for i in spam_indices:
    peso = data_test.iloc[i]["repetition_weight"]
    print(f"Correo #{i} - Peso de repetición: {peso:.4f}")
```

Por ultimo lo que hacemos es llamar el modelo de Naives para hacer los cálculos e imprimimos los datos correspondientes referentes a lo analizado

Resultados

```
PS D:\Archivos\Documentos\Escuela-Tec\8vo-Semestre\IA\Unidad-2\Tarea 3-Analizador de Spam> & C:/Users/ferch/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/Archivos/Documentos/Escuela-Tec/8vo-Semestre/IA/Unidad-2/Tarea 3-Analizador de Spam/clasificador.py"
Precisión del modelo:  0.925
Índices de correos clasificados como spam:
[4, 8, 10, 11, 12, 13, 17, 21, 26, 27, 31, 36, 37, 39, 40, 41, 46, 47, 49, 50, 55, 56, 66, 67, 69, 71, 75, 76, 77, 78, 81, 82, 87, 91, 95, 99, 101
, 102, 104, 105, 108, 109, 111, 114, 115, 117, 120, 121, 124, 125, 127, 128, 139, 142, 143, 149, 150, 151, 154, 155, 161, 163, 164, 169, 170, 171,
 173, 176, 178, 179, 182, 185, 187, 192, 195, 197, 201, 203, 209, 211, 213, 219, 220, 222, 224, 225, 231, 235, 239, 242, 243, 249, 252, 254, 256,
257, 258, 259, 264, 267, 269, 271, 274, 275, 285, 289, 294, 295, 306, 311, 317, 318, 330, 331, 333, 340, 345, 349, 353, 357, 360, 363, 366, 369, 3
70, 372, 377, 378, 379, 380, 387, 388, 389, 390, 399, 405, 406, 410, 416, 418, 421, 423, 426, 434, 435, 436, 438, 440, 442, 448, 449, 450, 454, 45
8, 459, 460, 465, 466, 473, 474, 477, 480, 487, 488, 489, 490, 495, 496, 504, 512, 518, 523, 527, 529, 532, 541, 543, 549, 554, 557, 564, 573, 580
, 582, 583, 584, 585, 587, 589, 590, 594, 596, 597, 599, 601, 602, 604, 612, 613, 615, 621, 622, 623, 624, 627, 628, 629, 632, 633, 652, 653, 656,
 659, 660, 662, 664, 669, 670, 671, 673, 676, 678, 682, 687, 693, 699, 705, 711, 715, 717, 718, 720, 721, 723, 727, 728, 732, 733, 737, 739, 742,
743, 744, 746, 748, 749, 752, 756, 757, 758, 759, 762, 765, 766, 768, 771, 773, 776, 781, 787, 791, 793, 795, 801, 808, 810, 811, 813, 819, 823, 8
24, 826, 828, 831, 832, 834, 859, 865, 866, 869, 871, 879, 881, 882, 891, 898, 899, 900, 902, 904, 906, 907, 912, 913, 915, 919, 920, 921, 924, 92
7, 935, 945, 949, 951, 952, 953, 956, 962, 965, 969, 972, 975, 976, 977, 978, 979, 980, 982, 983, 986, 987, 988, 989, 990, 997, 998, 1000, 1005, 1
006, 1008, 1009, 1010, 1016, 1026, 1028, 1029, 1031, 1038, 1039, 1040, 1045, 1049, 1051, 1053, 1054, 1058, 1060, 1061, 1063, 1064, 1068, 1073, 107
6, 1079, 1084, 1085, 1093, 1094, 1095, 1096, 1098, 1099, 1100, 1102, 1104, 1106, 1108, 1111, 1114, 1117, 1121, 1127, 1133, 1136, 1147, 1148, 1153,
 1154, 1156, 1158, 1159]

Pesos de repetición de los correos clasificados como SPAM:
Correo #4 - Peso de repetición: 0.5649
Correo #8 - Peso de repetición: 0.4672
Correo #10 - Peso de repetición: 0.6171
Correo #11 - Peso de repetición: 0.4557
Correo #12 - Peso de repetición: 0.4100
Correo #13 - Peso de repetición: 0.4306
Correo #17 - Peso de repetición: 0.4598
Correo #21 - Peso de repetición: 0.4577
```