Detektor spamu w wiadomościach SMS

Opis problemu:

Spam SMS, czyli niechciane wiadomości tekstowe są obecnie wysyłane masowo w celach reklamowych, bądź też oszukańczych. Wysyłanie tego typu powiadomień zwykle bywa bardzo uciążliwe dla odbiorców i podnosi ryzyko stania się ofiarą groźnych oszustw. Problem dotyczy wielu branż m.in. branży telekomunikacyjnej, czy finansowej. Wykrywanie spamu SMS ma na celu ochronę użytkowników przed niechcianymi wiadomościami, ryzkiem zostania oszukanym oraz poprawę jakości usług komunikacyjnych.

Cel projektu

Celem projektu jest opracowanie i wdrożenie skutecznego modelu klasyfikacyjnego, który będzie w stanie automatycznie identyfikować wiadomości SMS o charakterze spamowym.

Kto i w jaki sposób może skorzystać z tego modelu?

Model ten może zostać przeznaczony do użytku przez użytkowników telefonów komórkowych w celu ochrony przed niechcianymi wiadomościami, operatorów telekomunikacyjnych w celu zmniejszenia obciążenia sieci i poprawy jakości świadczonych usług, firmy e-commerce w celu lepszej komunikacji z klientami, eliminacji ryzyka wprowadzenia klientów w błąd przez oszustów.

Źródło danych

Podstawowy zbiór danych zawiera:

- Treść wiadomości SMS: tekst wiadomości
- Typ: klasyfikacja wiadomości jako spam lub ham (wiadomość bezpieczna, normalna)

W celu poprawy jakości modelu, zbiór danych po wcześniejszej analizie zostanie wzbogacony o dodatkowe kolumny wygenerowane na podstawie treści wiadomości . Kolumny te pomogą w wychwyceniu specyficznych cech charakterystycznych dla spamu.

Dataset zawiera 5169 rekordów i po jego refaktoryzacji (wzbogaceniu o dodatkowe kolumny) zostanie podzielony na **zbiór treningowy** o wielkości **3618** rekordów oraz **zbiór testowy** o wielkości **1551** rekordów

Link do datasetu: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

Fragment datasetu:

ham 87% spam 13%	5169 unique values
ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got a...
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entr...
ham	U dun say so early hor... U c already then say...
ham	Nah I don't think he goes to usf, he lives around here though
spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for

Struktura pracy nad modelem

1. Pozyskanie i refaktoryzacja danych: Zebranie danych i przetworzenie ich na strukturę zgodną z wymaganiami.
2. Przetwarzanie danych: Usuwanie brakujących wartości, utworzenie dodatkowych kolumn, przygotowanie cech numerycznych, utworzenie wykresów i analiz.
3. Podział zbioru danych: Podział na dane treningowe (70%) i testowe (30%) w celu przeprowadzenia walidacji i kształcenia modelu.
4. Trenowanie modelu: Wybór algorytmu klasyfikacji, dostosowanie parametrów modelu do osiągniecia wyznaczonych celów.
5. Walidacja i testowanie: Weryfikacja skuteczności na zbiorze testowym, obliczenie kluczowych miar efektywności i zależności modelu.
6. Wdrożenie: Konteneryzacja modelu do wykorzystania produkcyjnego.
7. Publikacja modelu i jego monitorowanie: Monitorowanie działania modelu na środowisku produkcyjnym i aktualizacja parametrów jeśli zajdzie taka potrzeba.

Prototypowanie i eksploracja danych z automatyczną analizą modeli

Wybrane narzędzie AutoML: **H2O**

H2O to zaawansowane narzędzie do automatycznego uczenia maszynowego, które umożliwia budowanie, trenowanie i optymalizowanie modeli ML, wybrałem je z uwagi na rekomendacje internautów problemy z wdrożeniem i działaniem TPOT.

Wnioski z raportu H2O:

Na podstawie raportu wygenerowanego przez H2O dla tego konkretnego problemu detekcji spamu, narzędzie zasugerowało kilka modeli, z których najlepszym okazały się modele typu **Gradient Boosting Machine (GBM)**. Model ten wyróżniał się wysoką dokładnością, która na zbiorze testowym wynosiła 98.744%, oraz średnim błędem bezwzględnym (MAE) równym 0.1310.

```
| (done) 100%
model_id                                                  auc    logloss     aucpr    mean_per_class_error      rmse        mse
GBM_1_AutoML_1_20241114_183734                          0.993047  0.0591547  0.973007              0.052208   0.124889  0.0155972
GBM_2_AutoML_1_20241114_183734                          0.992783  0.0596143  0.973301              0.0450771  0.121669  0.0148033
StackedEnsemble_AllModels_1_AutoML_1_20241114_183734    0.992635  0.0573297  0.973615              0.054011   0.121026  0.0146474
GBM_4_AutoML_1_20241114_183734                          0.992303  0.0609244  0.972391              0.0456702  0.122559  0.0150207
GBM_3_AutoML_1_20241114_183734                          0.991529  0.0599826  0.972122              0.0489223  0.120545  0.0145312
StackedEnsemble_BestOfFamily_1_AutoML_1_20241114_183734 0.991066  0.0607364  0.972653              0.052208   0.124318  0.0154549
DRF_1_AutoML_1_20241114_183734                          0.988855  0.100125   0.970353              0.044575   0.126286  0.0159483
XRT_1_AutoML_1_20241114_183734                          0.988555  0.0804946  0.971448              0.0439819  0.125207  0.0156768
GLM_1_AutoML_1_20241114_183734                          0.984135  0.0974686  0.944133              0.0752083  0.156549  0.0245075
[9 rows x 7 columns]

gbm prediction progress: | (done) 100%
ModelMetricsBinomial: gbm
** Reported on test data. **

MSE: 0.01181960352495502
RMSE: 0.10871800000439219
LogLoss: 0.049205800924956196
Mean Per-Class Error: 0.03631126907956141
AUC: 0.9931459710188651
AUCPR: 0.979560853118354
Gini: 0.9862919420377303

Confusion Matrix (Act/Pred) for max f1 @ threshold = 0.46176638303065554
       0     1    Error    Rate
-----  ----  ---  -------  -------------
0      1447  6    0.0041   (6.0/1453.0)
1      15    204  0.0685   (15.0/219.0)
Total  1462  210  0.0126   (21.0/1672.0)
```

```
Dokładność modelu: 0.9874401913875598
```

Gradient Boosting Machine (GBM) to algorytm łączący słabe modele, głównie drzewa decyzyjne, w jeden silny model. Proces polega na tworzeniu modeli jeden po drugim, gdzie każdy kolejny model stara się naprawić błędy poprzedniego.

Uzasadnienie:

GBM, dzięki swojemu procesowi boosowania, efektywnie identyfikuje subtelne różnice między wiadomościami spamowymi a normalnymi, co czyni go doskonałym wyborem do klasyfikacji spamu, szczególnie w tym przypadku gdy dane są bogate w różnorodne cechy.

Instrukcja uruchamiania pipeline i serwisu REST API:

Jak uruchomić środowisko Airflow:

1. Zbuduj i uruchom kontenery Docker dla Airflow tą komendą:

docker-compose up -d

2. Zaloguj się, domyślne dane logowania: login: airflow, hasło: airflow
3. Włącz pipeline:

Uruchom pipeline klikając na ikonę Play w szczegółach DAGa.

Jak uruchomić serwis REST API:

1. Zbuduj obraz Dockera dla serwisu API:

docker build -t ml_api .

2. Uruchom kontener dla API:

docker run -p 5000:5000 ml_api

Jak przetestować API przy użyciu Postman:

1. Ustaw URL: http://localhost:5000/predict
2. Wybierz metodę POST
3. W zakładce "Body" wybierz opcję raw i format JSON
4. Wklej przykładowe dane np.:

{

"v2": "CONGRATS!!! You have won 2000$ in our prize draw! Call 111333444 to claim your reward :).",

"message_length": 85,

"num_digits": 9,

"num_uppercase": 8,

"num_special_chars": 5,

"num_words": 16,

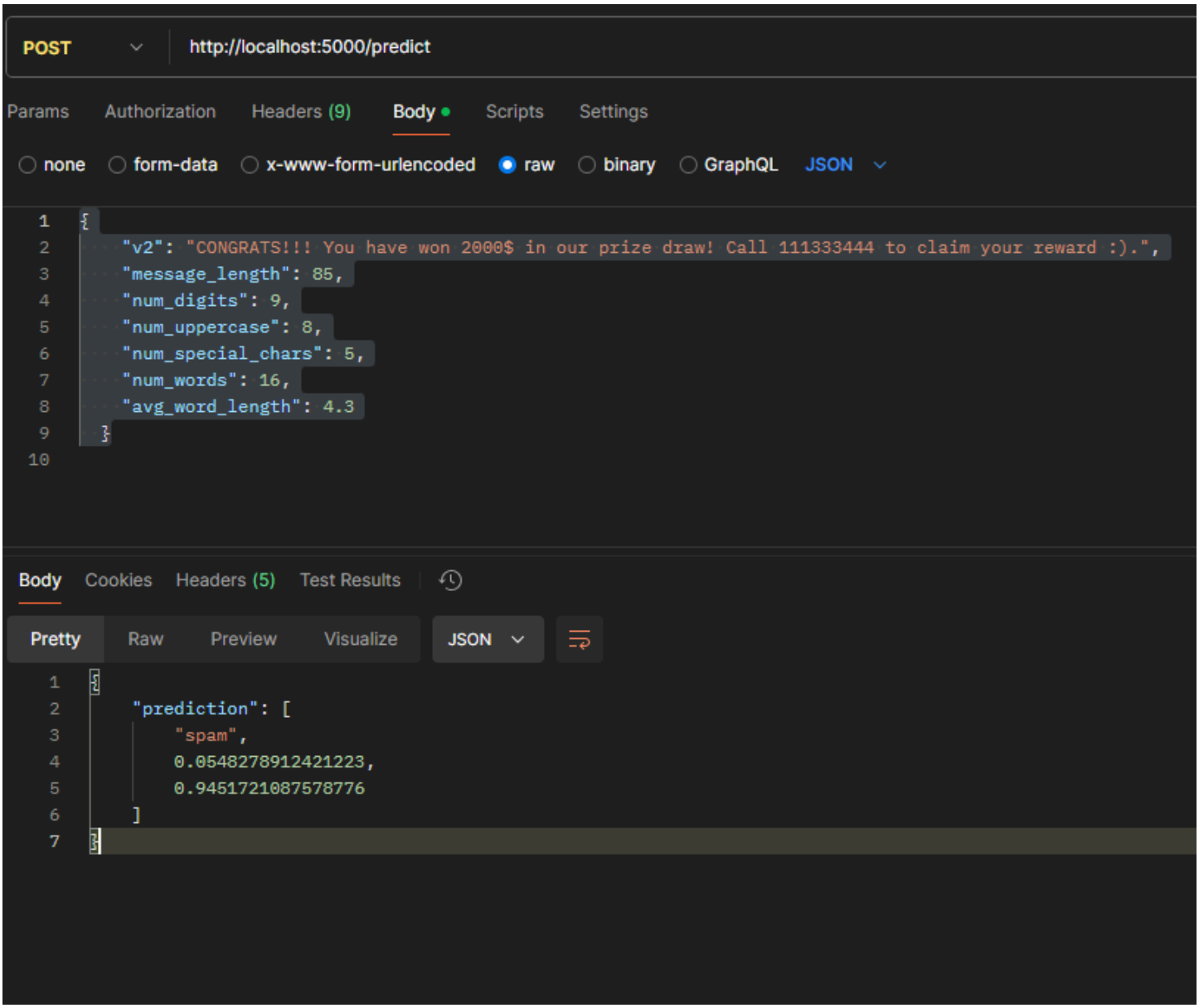
"avg_word_length": 4.3

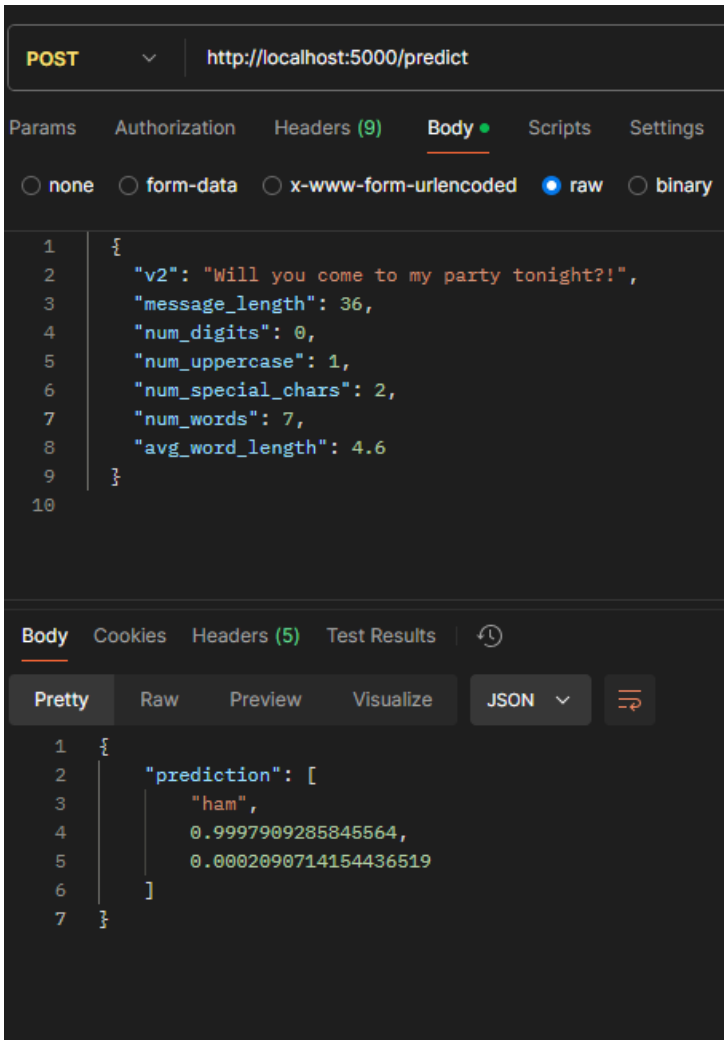
}

5. Kliknij “Send”, po chwili otrzymasz odpowiedź od serwera z wynikiem predykcji ham lub spam

Wyniki testów REST API:

Testy przyniosły oczekiwane rezultaty, model prawidłowo przewidywał typy wiadomości, użyto przykładowych danych wymyślonych “z głowy”, które odpowiadają typowym zwykłym wiadomościom i wiadomościom spam np. *“CONGRATS!!! You have won 2000$ in our prize draw! Call 111333444 to claim your reward :).” lub "Will you come to my party tonight?!"*


POST
http://localhost:5000/predict
Params
Authorization
Headers (9)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"v2": "CONGRATS!!! You have won 2000$ in our prize draw! Call 111333444 to claim your reward :).",
"message_length": 85,
"num_digits": 9,
"num_uppercase": 8,
"num_special_chars": 5,
"num_words": 16,
"avg_word_length": 4.3
}
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"prediction": [
"spam",
0.0548278912421223,
0.9451721087578776
]
}


POST
http://localhost:5000/predict
Params
Authorization
Headers (9)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
{
"v2": "Will you come to my party tonight?!",
"message_length": 36,
"num_digits": 0,
"num_uppercase": 1,
"num_special_chars": 2,
"num_words": 7,
"avg_word_length": 4.6
}
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"prediction": [
"ham",
0.9997909285845564,
0.0002090714154436519
]
}