Projet 5 : Soutenance

Catégorisez automatiquement des questions

Gaëtan PELLETIER

Sommaire

- Problématique, interprétation et pistes de recherche envisagées
- Acquisition et nettoyage des données
- Modélisation non supervisée
- Modélisation supervisée
- Déploiement d’une API
- Synthèse

Projet 5 : Soutenance

Problématiques,
interprétation
et pistes de recherche envisagées

Problématiques

D’après les questions postées sur Stack Overflow, les problématiques sont :

- Comment pouvons-nous prédire les tags d’une question?
- Comment pouvons-nous déployer nos modèles de prédiction sur le web ?

Interprétation

- **Comment pouvons-nous prédire les mots-clefs d'une phrase ?**
  - Extraction des mots pertinents au sein d'une phrase
  - Mise en place d'algorithmes de prédictions
  - Évaluation de la qualité des prédictions
- **Comment pouvons-nous déployer nos modèles de prédiction sur le web ?**
  - Gestion des versions du code
  - Déploiement de l'application web

Pistes de recherche envisagées

- Obtention de données de « bonne » qualité
- Analyse de la répartition des tags
- Nettoyage des données
- Modélisation non-supervisée
- Modélisation supervisée
- Déploiement d'une API

Acquisition et nettoyage des données

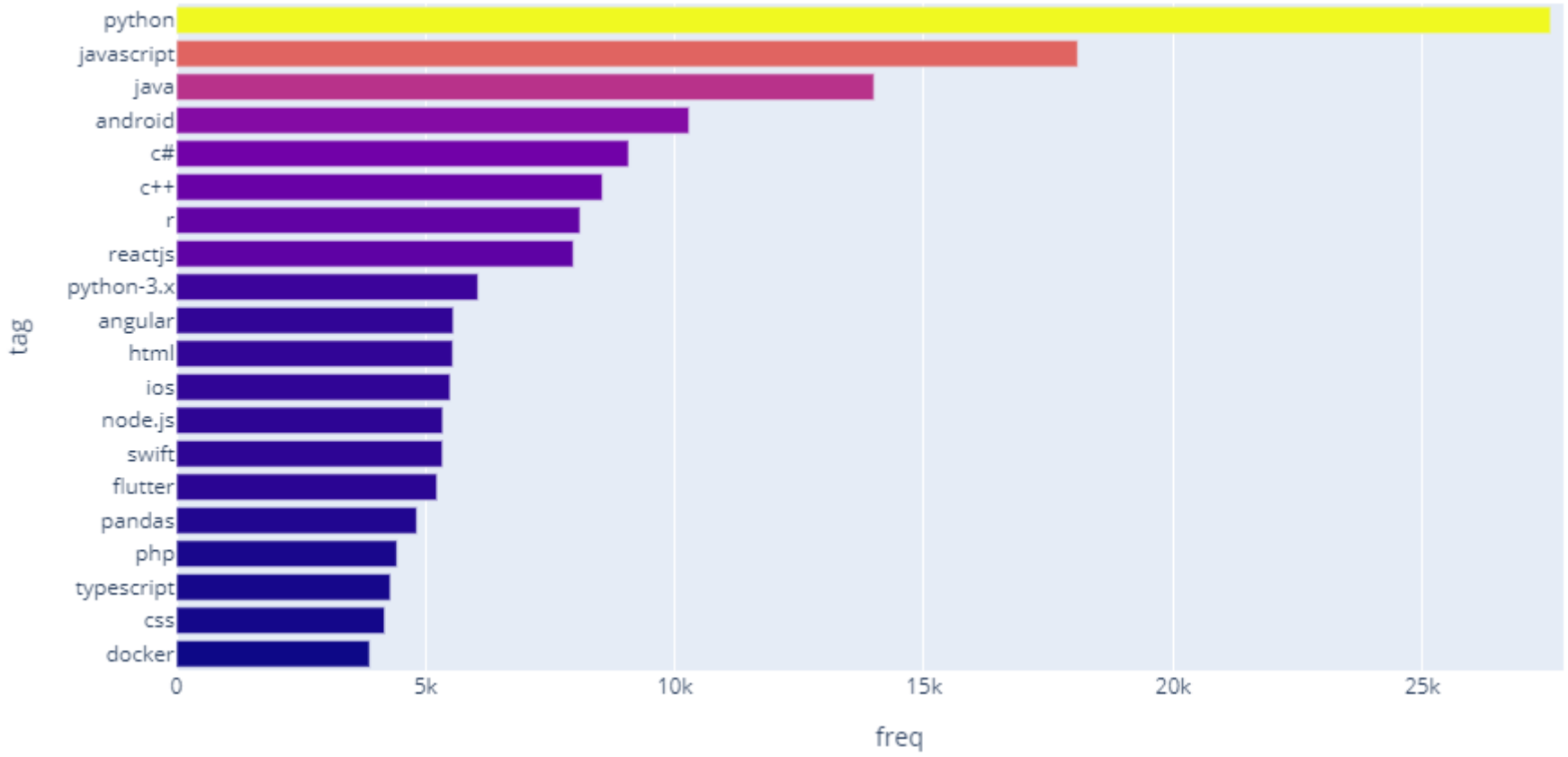
Acquisition des données

- **Requêtes SQL sur le site *stackexchange explorer***
- **Récupération de données de qualité :**
  - *Title* et *Tags* non nuls
  - *Score*, *ViewCount* et *FavoriteCount* > 0
  - Questions récentes (vocabulaire moderne)

```
SELECT CreationDate, Body, Tags, Title, Score, ViewCount
FROM posts
WHERE Tags IS NOT NULL
AND Title IS NOT NULL
AND Score > 0
AND ViewCount > 0
AND FavoriteCount > 0
ORDER BY CreationDate desc
```

Nettoyage des données

Analyse des tags



296 tags retenus (~ 60 % du nombre total)

Nettoyage des données

- Suppression des questions n'utilisant pas l'un des 296 tags
- Suppression des balises html
- Tokenization des questions et titres
- Suppression des mots avec peu de valeur sémantique :
  - Adjectifs
  - Adverbes
- Concaténation des questions et titres
- Lemmatization des tokens

Modélisation non supervisée : Latent Dirichlet Allocation

Modélisation non supervisée

Entraînement algorithme TF-IDF (~700 features)

Liste de mots supprimés

```
['マミムメモ たちつてと',
 'merge index',
 'constructor nature',
 'plane ship',
 'myinfo',
 'stop jvm',
 'throw relate',
 'key match',
 'interview question',
 'exclude actionmailer',
 'choice user',
 'app androidmanifest',
 'maingit',
 'configuration dockerfile',
 'itershape',
 'allquestionsandanswersclass']
```

(Listes non exhaustives)

Features et coeff TF-IDF

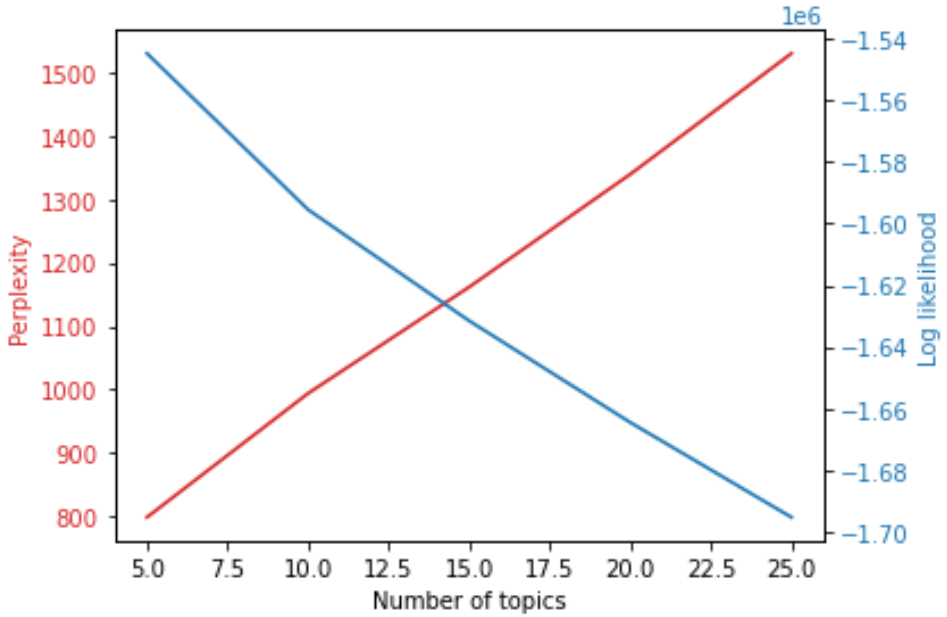
	features	tf-idf
497	reduce	5.602156
172	developer	5.602156
656	usage	5.602156
322	invoke	5.600216
180	disable	5.598280
90	cloud	5.598280
558	set code	5.598280
549	security	5.598280
85	cli	5.598280
112	compose	5.594419

	features	tf-idf
636	try	1.884236
91	code	2.063309
675	want	2.182489
692	work	2.196954
524	return	2.276282
657	use	2.332919
204	error	2.470299
679	way	2.519712
138	create	2.567390
665	value	2.574229

Modélisation non supervisée

LDA – Nombre de topics

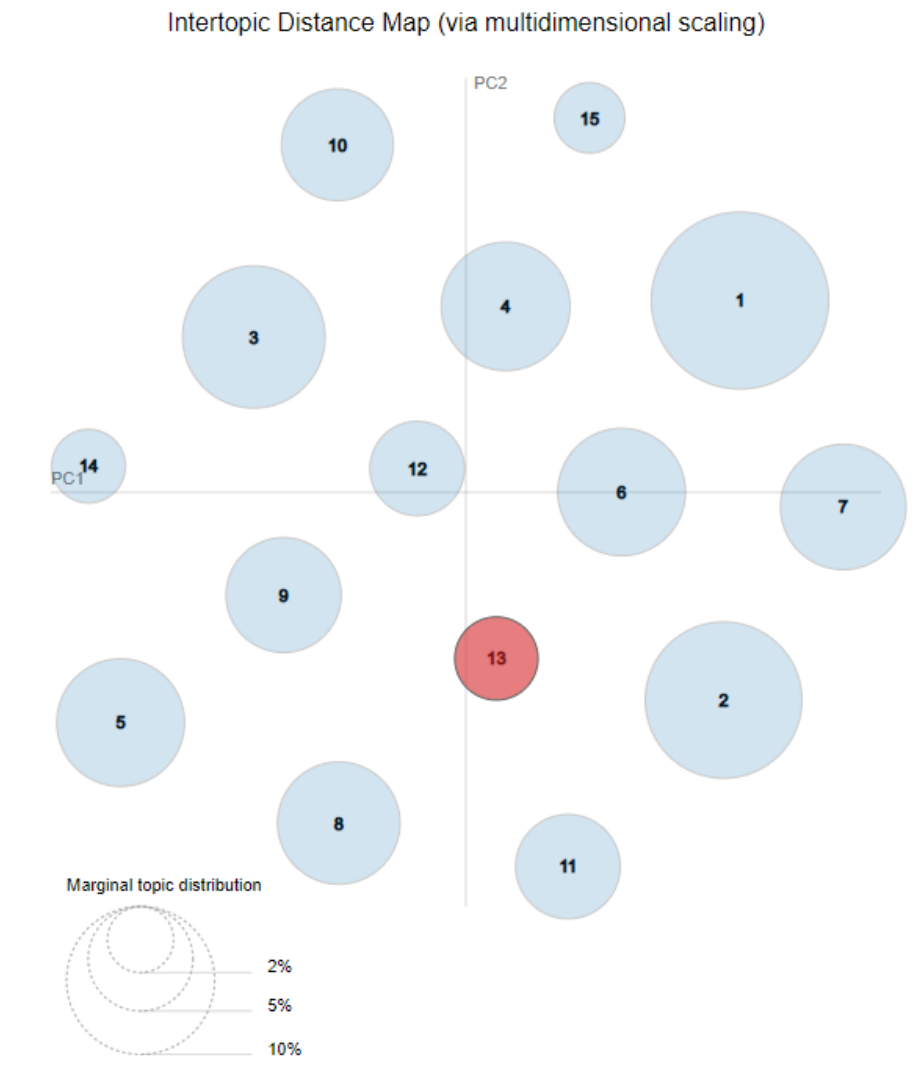
- Entraînement avec différents nombres de topics
- Calcul de la perplexité et de la vraisemblance

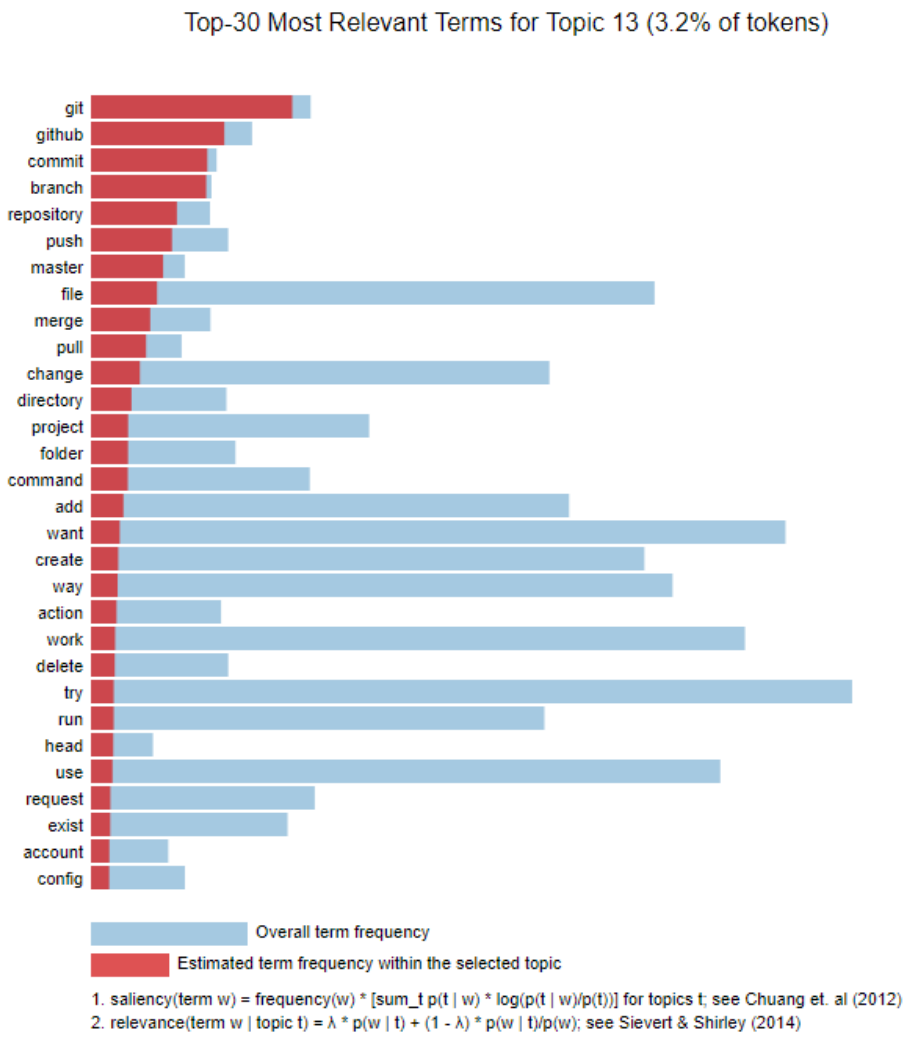


15 topics semblent être un bon compromis

Modélisation non supervisée

LDA – Visualisation des topics créés





Modélisation non supervisée

Évaluation de la qualité des prédictions :

- **Pour un échantillon de 30 questions :**
  - Attribution manuelle d'un numéro de topic de LDA (y_true)
  - Prédiction du numéro d'un topic par LDA (y_pred)
  - Comparaison des classes prédites (y_pred) avec les classes réelles (y_true)
  - F1 score = 0,35 Jaccard score = 0,25
- **Limitations : choix arbitraire + échantillon réduit**

Modélisation supervisée

Modélisation supervisée

- **Séparation du dataset :**
  - Jeu d'entraînement
  - Jeu de test

- **Encodage des tags :**

  → transformation de la colonne tags avec MultiLabelBinarizer

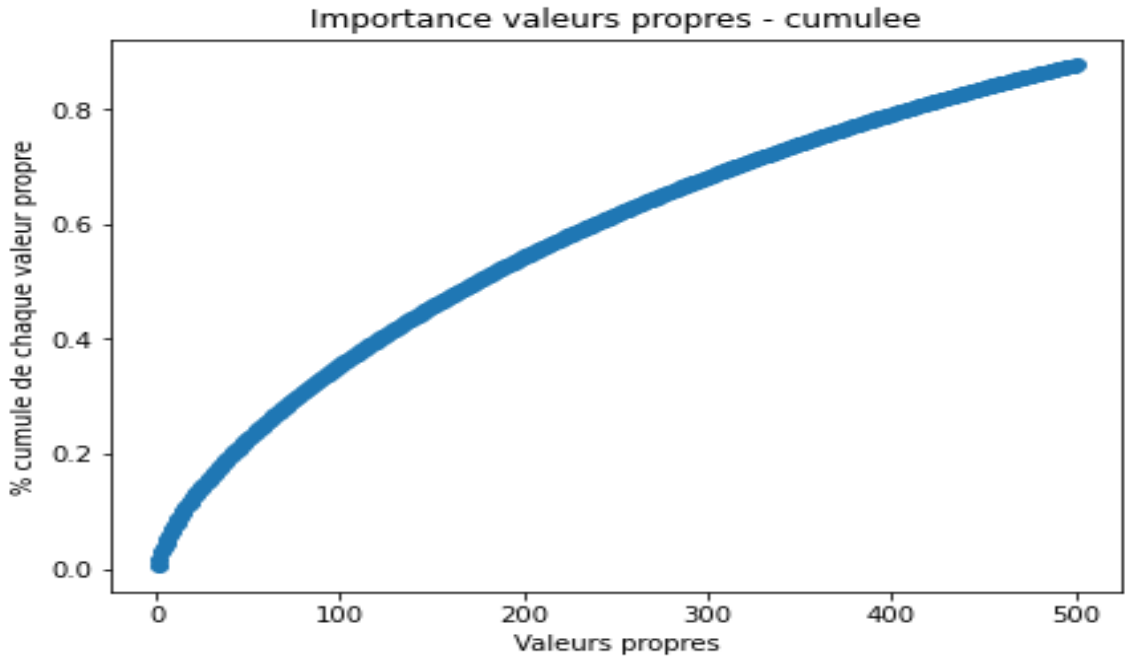
- **Préparation du décodage des prédictions :**

  → entraînement d'un LabelEncoder
  avec les classes de MultiLabelBinarizer

Modélisation supervisée

- **Entraînement algorithme TF-IDF** (~700 features)
- **Réduction de dimensions : Truncated SVD**

  → 420 composantes représentent environ 81% de la variance des données



Modélisation supervisée

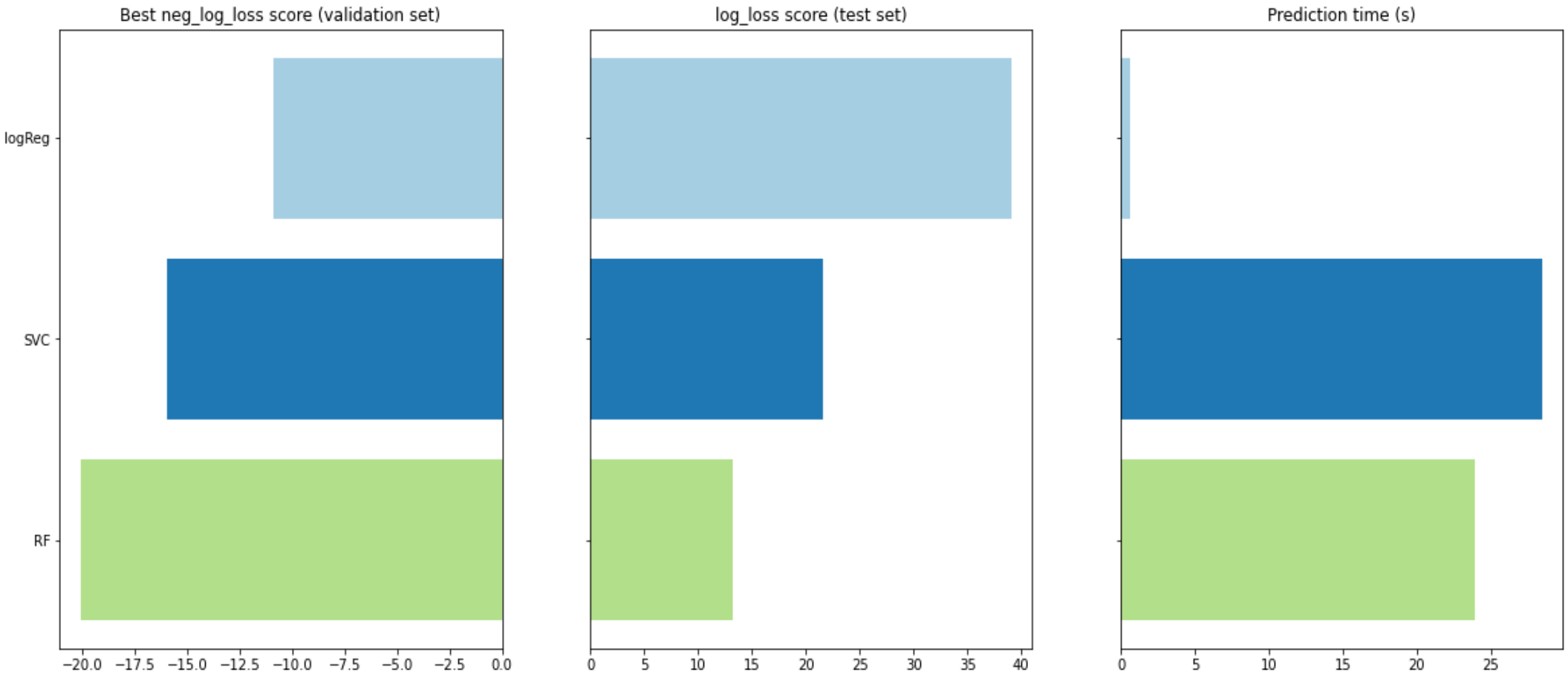
- **Problème de classification multiclasse :**

  → utilisation de la stratégie One-versus-the-Rest

- **Entraînement de trois modèles sur des jeux réduits :**
  - Linéaire → Régression Logistique
  - Non linéaire → SVM à noyau gaussien
  - Ensembliste → Forêt Aléatoire
- **Recherche sur grille et validation croisée à 3 plis**
- **Métriques utilisées :**
  - neg_log_loss (sur le jeu de validation)
  - log_loss (sur le jeu de test)
  - Temps de prédiction

Modélisation supervisée

Choix du meilleur modèle



Modélisation supervisée

- **Entraînement de notre régression logistique**
  - Stratégie One vs Rest
  - Recherche sur grille pour optimiser paramètre de régulation C
  - Validation croisée à 5 plis

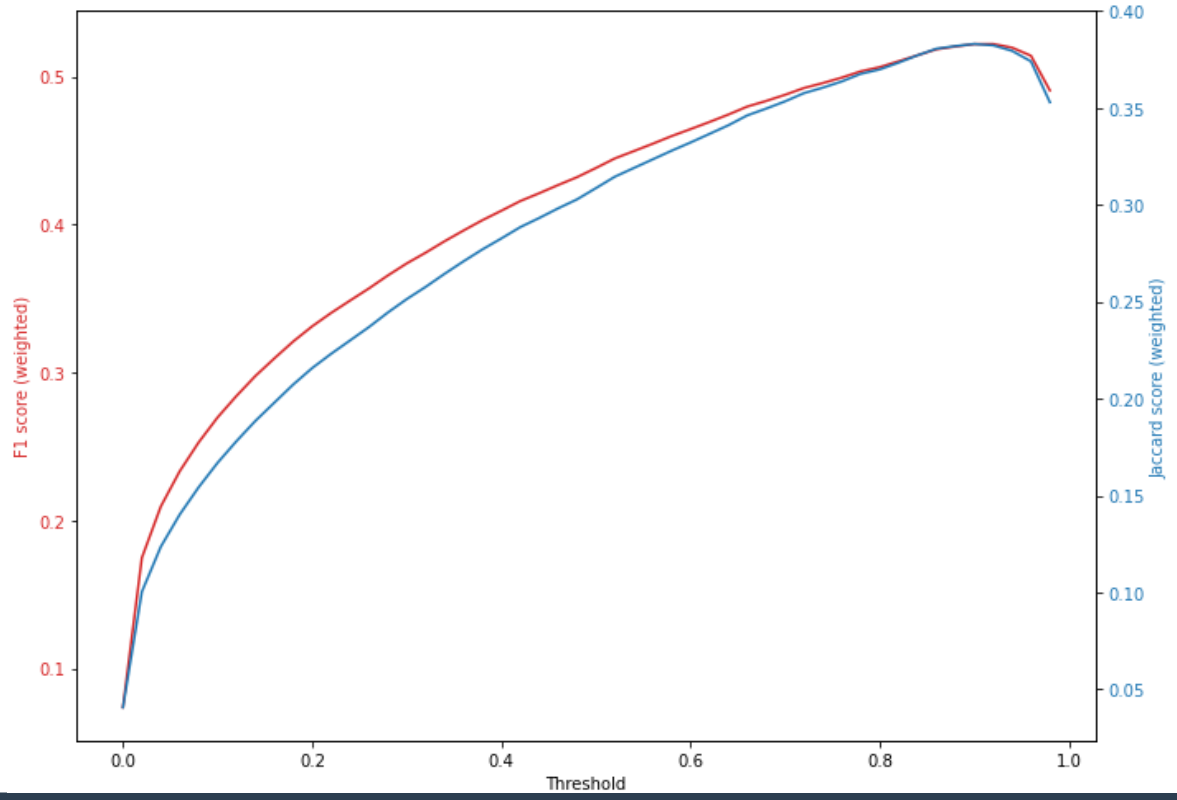
- **Évaluation de la qualité des prédictions**

  → calcul des scores F1 et Jaccard

Modélisation supervisée

Évaluation de la qualité des prédictions :

- Prédictions effectuées avec predict_proba
- Calcul des métriques en fonction du seuil de décision




```
max scores

Threshold =  0.9

Scores (average == weighted):

Jaccard score     = 0.38
F1 score          = 0.52
```

Déploiement d’une API

Déploiement d'une API

- Enregistrement des modèles avec **dill** (*pickle*)
- Création d'une API locale avec **Flask**
- Gestion des versions de l'application avec **Git**
- Déploiement de l'application avec **Heroku** : **https://gp-iml-project5-tags.herokuapp.com/**

Synthèse

Synthèse

- **Acquisition et nettoyage des données**
  - Téléchargement de questions de bonnes qualité
  - Conservation des tags les plus fréquents
  - Tokenization et conservation de mots avec sémantiques pertinentes
  - Lemmatization
- **Modèle non supervisé : LDA**
  - Visualisation des topics
  - Évaluation de la qualité des prédictions
- **Modèles supervisés :**
  - Stratégie de One vs Rest (problème multiclasse)
  - Comparaison de trois modèles (LogReg, SVM et Random Forest)
  - Entraînement du modèle final : régression logistique
  - Évaluation de la qualité des prédictions
- **Déploiement des modèles dans une API web**

Merci de votre attention

Annexes

Projet 5 : Annexe 1

Find the tags of your question

Finding duplicate values in a SQL table

It's easy to find duplicates with one field. So if we have a table. This query will give us John, Sam, Tom, Tom because they all have the same email. However, what I want is to get duplicates with the same email and name. That is, I want to get 'Tom', 'Tom'. The reason I need this: I made a mistake, and allowed to insert duplicate name and email values. Now I need to remove/change the duplicates, so I need to find them first.

Suggest tags

Supervised algorithm:

hibernate | php | sql

Unsupervised algorithm:

query | table | select | database | field