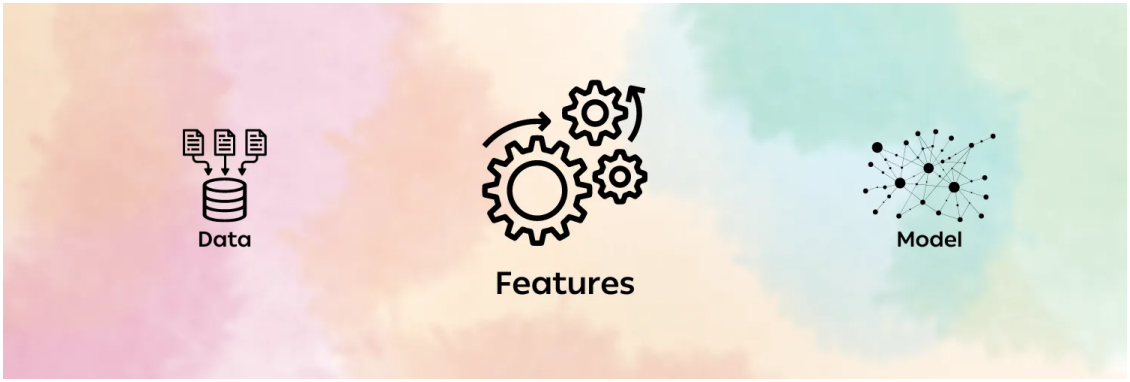
feature_Eng_Xboost

September 19, 2024

1 Feature Engineering in XGBoost: A Practical Guide



L'ingénierie des caractéristiques est l'une des étapes les plus critiques dans la construction de modèles d'apprentissage automatique de haute performance. Dans ce guide, je vais passer en revue les étapes pratiques pour effectuer l'ingénierie des caractéristiques dans XGBoost. Que vous soyez ingénieur en IA/ML ou novice en apprentissage automatique, j'espère que ce guide vous fournira une approche claire et pratique pour améliorer vos modèles.

1.1 Qu'est-ce que le Feature Engineering et pourquoi est-il important ?

Le Feature Engineering est le processus de transformation des données brutes en caractéristiques qui facilitent la compréhension des modèles par le modèle d'apprentissage automatique. Par exemple, au lieu d'utiliser l'emploi et la localisation de l'utilisateur (ex. Los Angeles/LA/Langeles, etc.), vous pouvez l'encoder pour représenter la proximité d'un demandeur d'emploi par rapport au bureau avec une distance (ex. 10 km). Ces transformations améliorent les performances du modèle en créant des données qui représentent mieux le problème à résoudre.

Ces transformations améliorent les performances du modèle en créant des données qui représentent mieux le problème en question. Dans cet exemple de cas d'utilisation, je cherche à faire correspondre les demandeurs d'emploi avec les offres d'emploi en utilisant un ensemble de données qui contient des informations sur les emplois, les demandeurs d'emploi et leurs candidatures. L'objectif de l'ingénierie des caractéristiques ici est d'aider XGBoost à faire des prédictions plus précises sur les offres d'emploi qui intéressent un demandeur d'emploi particulier ou qui lui conviennent.

1.2 Pourquoi l'ingénierie des caractéristiques est-elle importante pour XGBoost ?

XGBoost (Extreme Gradient Boosting) est un algorithme d'apprentissage automatique puissant et populaire. Il excelle dans les tâches de classification et de classement, telles que la détermination des offres d'emploi les plus susceptibles de correspondre à un demandeur d'emploi donné. Cependant, l'efficacité de l'algorithme repose sur la qualité des caractéristiques d'entrée. Des caractéristiques bien conçues peuvent améliorer considérablement les performances de XGBoost, en l'aidant à faire de meilleures prédictions.

Présentation de l'ensemble de données Je vais utiliser l'ensemble de données Kaggle Job Recommendation Challenge, qui comprend : Offres d'emploi : Contient des caractéristiques telles que le titre de l'emploi, la description du poste, le lieu et les compétences requises. Profils des demandeurs d'emploi : Contient des informations telles que l'intitulé du poste actuel du demandeur d'emploi, ses compétences, ses années d'expérience et son lieu de résidence. Candidatures : Un enregistrement des demandeurs d'emploi qui ont postulé à telle ou telle offre d'emploi, ce qui nous aide à comprendre leurs préférences.

1.3 Loading the Dataset

Let's start by loading the dataset:

[2]:
```
import pandas as pd
```

[1]:
```
import pandas as pd

jobs_df = pd.read_csv('jobs.tsv', sep='\t', on_bad_lines='skip')
apps_df = pd.read_csv('apps.tsv', sep='\t', on_bad_lines='skip')
```

```
/tmp/ipykernel_64222/5581363.py:3: DtypeWarning: Columns (8) have mixed types.
Specify dtype option on import or set low_memory=False.
  jobs_df = pd.read_csv('jobs.tsv', sep='\t', on_bad_lines='skip')
```

[2]:
```
job_seekers_df = pd.read_csv('users.tsv', sep='\t', on_bad_lines='skip')
```

Je vais fusionner ces trois cadres de données pour obtenir toutes les informations que je vais utiliser.

[3]:
```
df = apps_df.merge(jobs_df, on="JobID", how="inner")
df = df.merge(job_seekers_df, on="UserID", how="inner")
```

[28]:
```
df = df.rename(columns={'City_x': 'Job_City', 'State_x': 'Job_State',␣
↪'Country_x': 'Job_Country', 'City_y': 'User_City', 'State_y': 'User_State',␣
↪'Country_y': 'User_Country'})
```

Renommons également certaines colonnes qui portent le même nom.

[29]:
```
df = df.rename(columns={'City_x': 'Job_City', 'State_x': 'Job_State',␣
↪'Country_x': 'Job_Country', 'City_y': 'User_City', 'State_y': 'User_State',␣
↪'Country_y': 'User_Country'})
```

1.4 Techniques d'ingénierie des caractéristiques

Passons en revue quelques techniques pratiques d'ingénierie des caractéristiques à l'aide de cet ensemble de données. Traitement des valeurs manquantes J'ai appris qu'il ne faut pas s'attendre à trouver les données dans des conditions idéales. Je peux afficher toutes les colonnes contenant des valeurs nulles dans le cadre de données df en appelant la méthode sum() sur

df.isnull() .

[30]:
```
df.isnull().sum()
```

[30]:
```
UserID                        0
WindowID_x                    0
Split_x                       0
ApplicationDate               0
JobID                         0
WindowID_y                    0
Title                         0
Description                  11
Requirements              21461
Job_City                      1
Job_State                     0
Job_Country                   0
Zip5                     491435
StartDate                     0
EndDate                       2
WindowID                      0
Split_y                       0
User_City                     0
User_State                 2228
User_Country                  0
ZipCode                   10058
Major                    391347
GraduationDate           483332
WorkHistoryCount              0
TotalYearsExperience          0
CurrentlyEmployed        183486
ManagedOthers                 0
ManagedHowMany                0
experience_level           4142
DegreeType_Associate's        0
DegreeType_Bachelor's         0
DegreeType_High School        0
DegreeType_Master's           0
DegreeType_PhD                0
DegreeType_Vocational         0
location_encoded              0
location_similarity           0
```

```
dtype: int64
```

XGBoost peut gérer les valeurs manquantes, mais il est parfois utile de les remplir nous-mêmes, surtout si elles sont manquantes dans un but précis.

[6]:
```
df['TotalYearsExperience'] = df['TotalYearsExperience'].fillna(0)
```

1.5 2. Regroupement des variables continues

Un autre moyen d'aider le modèle à reconnaître des modèles plus généraux consiste à regrouper les variables continues, telles que les années d'expérience, en catégories. Par exemple, les années d'expérience des demandeurs d'emploi peuvent être classées en trois catégories : débutant, intermédiaire, senior et expert.

[7]:
```
bins = [0, 2, 5, 10, 60]
labels = ['Entry-level', 'Mid-level', 'Senior', 'Expert']
df['experience_level'] = pd.cut(df['TotalYearsExperience'], bins=bins,␣
 ↪labels=labels, right=False)
```

[13]:
```
df
```

[13]:

	UserID	WindowID_x	Split_x	ApplicationDate	JobID	WindowID_y	Title
0	47	1	Train	2012-04-04 15:56:23.537	169528	1	Resort Host/Marketing Coordinator - Anaheim, CA
1	47	1	Train	2012-04-06 01:03:00.003	284009	1	Administrative Assistant
2	47	1	Train	2012-04-05 02:40:27.753	2121	1	MEDICAL- FRONT OFFICE
3	47	1	Train	2012-04-05 02:37:02.673	848187	1	Administrative Assistant
4	47	1	Train	2012-04-05 22:44:06.653	733748	1	Administrative Assistant
…	…	…	…	…	…	…	…
1603073	1470823	7	Train	2012-06-26 14:49:55.42	640906	7	Restaurant Team Member - Crew
1603074	1471000	7	Train	2012-06-24 14:28:46.15	904025	7	Vice President of Operations / Audit
1603075	1471706	7	Train	2012-05-29 13:38:30.463	354772	7	Sales Associate Job
1603076	1471706	7	Train	2012-06-19 09:13:47.383	958259	7	Wireless Sales Associate - Part-Time - Target …
1603077	1471901	7	Train	2012-06-20 18:25:52.77	358993	7	Project support assistant

```
                                                 Description
0        <P STYLE="MARGIN-TOP: 0px; MARGIN-BOTTOM: 0px"…  \
1        <p><strong>Administrative Assistant $60K - $65…
2        Medical -  Front Office\r\nIndustrial Clinic, …
3        <DIV>\r<P><SPAN>ADMINISTRATIVE ASSISTANT</SPAN…
4        One of the largest oilfield services companies…
…                                                      …
1603073  <strong></strong>  Building the perfect burrit…
1603074  <p style="text-align: center"><strong>Vice Pre…
1603075  </b>\r\n\r\nExperience the excitement of Retai…
1603076  <strong>Job Functions / Duties / Responsibilit…
1603077  <b>Fremont area. Temp to hire position. Do you…

                                          Requirements     Job_City  …
0        • Bachelor's Degree preferred;<BR> \r\n• Posse…      Anaheim  …  \
1        Requirement:  2-4 years of significant ex…  Los Angeles  …
2        Please refer to the Job Description to view th…  Los Angeles  …
3        <P><SPAN style="TEXT-DECORATION: underline"></…  Los Angeles  …
4                                                    NaN   Long Beach  …
…                                                      …           …  …
1603073                                                Pasadena  …
1603074  <p>Please visit <a title="     " href="ht…    Oak Creek  …
1603075                                                  Baxter  …
1603076                                                  Baxter  …
1603077              \r\n<b>Job Code</b> : 4752B1848 \r\n      Fremont  …

        ManagedOthers ManagedHowMany experience_level DegreeType_Associate's
0                  No              0           Expert                   False  \
1                  No              0           Expert                   False
2                  No              0           Expert                   False
3                  No              0           Expert                   False
4                  No              0           Expert                   False
…                 …               …                …                       …
1603073            No              0           Senior                   False
1603074            No              0           Expert                   False
1603075            No              0           Expert                   False
1603076            No              0           Expert                   False
1603077            No              0           Expert                   False

        DegreeType_Bachelor's  DegreeType_High School DegreeType_Master's
0                       False                    True                False  \
1                       False                    True                False
2                       False                    True                False
3                       False                    True                False
4                       False                    True                False
…                       …                        …                   …
1603073                 False                    True                False
```

```
1603074                False                   False                True
1603075                False                    True               False
1603076                False                    True               False
1603077                False                   False               False

         DegreeType_PhD DegreeType_Vocational location_encoded
0                 False                 False                5
1                 False                 False                5
2                 False                 False                5
3                 False                 False                5
4                 False                 False                5
…                     …                     …                …
1603073           False                 False                5
1603074           False                 False               53
1603075           False                 False               24
1603076           False                 False               24
1603077           False                 False               37

[1603078 rows x 36 columns]
```

1.6 3. Interaction des caractéristiques

La création de caractéristiques d'interaction peut révéler des relations complexes entre différentes variables. Par exemple, un moyen simple et peu sophistiqué de saisir la similitude entre la localisation du demandeur d'emploi et celle de l'emploi serait une correspondance floue.

[14]:
```
pip install fuzzywuzzy
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting fuzzywuzzy
  Downloading fuzzywuzzy-0.18.0-py2.py3-none-any.whl.metadata (4.9 kB)
Downloading fuzzywuzzy-0.18.0-py2.py3-none-any.whl (18 kB)
Installing collected packages: fuzzywuzzy
Successfully installed fuzzywuzzy-0.18.0
Note: you may need to restart the kernel to use updated packages.
```

[23]:
```
from fuzzywuzzy import fuzz

df['location_similarity'] = df.apply(
    lambda row: fuzz.ratio(
        f"{row['User_City']}, {row['User_State']}, {row['User_Country']}",
        f"{row['Job_City']}, {row['Job_State']}, {row['Job_Country']}"
    )/100, axis=1
)
```

Cette fonctionnalité location_similarity fournit un score de similarité à l'aide de fuzz.ratio qui reflète le degré de correspondance entre la localisation de l'utilisateur et celle de l'emploi et qui peut être utilisé dans le modèle pour améliorer les prédictions. Comme il s'agit d'une comparaison

de chaînes, cette fonctionnalité d'interaction pourrait être améliorée si je convertis cette chaîne ville/état/pays en coordonnées et si je calcule la proximité de ces deux éléments.

1.7 4. Encodage des variables catégorielles

Les demandeurs d'emploi et les offres d'emploi comprennent souvent des variables catégorielles telles que les titres de poste, les secteurs d'activité et les lieux de travail. Ces variables doivent être transformées en représentations numériques pour que le modèle puisse les traiter efficacement.

1.8 One-hot encoding:

une représentation utile pour les variables catégorielles comportant un petit nombre de catégories, telles que le type de diplôme (par exemple, Associate's, Bachelor's, Master's, etc).

[25]:
```
print(df.columns)
```

```
Index(['UserID', 'WindowID_x', 'Split_x', 'ApplicationDate', 'JobID',
       'WindowID_y', 'Title', 'Description', 'Requirements', 'Job_City',
       'Job_State', 'Job_Country', 'Zip5', 'StartDate', 'EndDate', 'WindowID',
       'Split_y', 'User_City', 'User_State', 'User_Country', 'ZipCode',
       'Major', 'GraduationDate', 'WorkHistoryCount', 'TotalYearsExperience',
       'CurrentlyEmployed', 'ManagedOthers', 'ManagedHowMany',
       'experience_level', 'DegreeType_Associate's', 'DegreeType_Bachelor's',
       'DegreeType_High School', 'DegreeType_Master's', 'DegreeType_PhD',
       'DegreeType_Vocational', 'location_encoded', 'location_similarity'],
      dtype='object')
```

[26]:
```
if 'DegreeType' in df.columns:
    df = pd.get_dummies(df, columns=['DegreeType'])
else:
    print("Column 'DegreeType' not found in the DataFrame")
```

```
Column 'DegreeType' not found in the DataFrame
```

df = pd.get_dummies(df, columns=['DegreeType'])

[ ]:
```
df
```

1.9 Label encoding:

mieux adaptée aux variables catégorielles comportant de nombreuses catégories, telles que l'État.

[ ]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['location_encoded'] = le.fit_transform(df['Job_State'])
```

1.10 Target Encoding:

une autre méthodologie pour les variables catégorielles mais avec de nombreuses valeurs uniques, telles que les exigences, qui peut être plus efficace que l'encodage d'un seul point. Cette tech-

nique aide le modèle à se concentrer sur les catégories qui ont le plus de chances d'aboutir à une correspondance (pour cet exemple, j'ai annoté manuellement les données de correspondance).

[ ]:
```
mean_match_rate = df.groupby('Requirements')['match'].mean()
df['job_req_encoded'] = df['Requirements'].map(mean_match_rate)
```

1.11 5. Log Transformation:

Même si cet ensemble de données Kaggle ne l'inclut pas, le salaire a souvent une distribution asymétrique. Pour la démonstration, créons quelques données fictives pour la nouvelle colonne de salaire.

[ ]:
```
import numpy as np

df['salary'] = np.random.randint(20000, 300001, size=len(df))
```

Pour faciliter l'apprentissage du modèle, j'appliquerai une transformation logarithmique qui permettra de normaliser ces données.

[ ]:
```
df['log_salary'] = np.log1p(df['salary'])
```

L'objectif est d'atténuer les variations importantes et de faciliter le traitement des données par le modèle.

1.12 6. Date and Time Features

Je peux également extraire des caractéristiques utiles des colonnes de date et d'heure. Par exemple, la date de début de l'emploi et la date de dépôt de la candidature peuvent influencer les chances d'embauche d'un demandeur d'emploi. J'utilise également la connaissance du domaine pour améliorer encore le modèle.

[ ]:
```
df['ApplicationDate'] = pd.to_datetime(df['ApplicationDate'], errors='coerce')
df['StartDate'] = pd.to_datetime(df['StartDate'], errors='coerce')

df['days_to_start'] = (df['StartDate'] - df['ApplicationDate']).dt.days
```

1.13 Préparation des données pour XGBoost

Après avoir effectué l'ingénierie des caractéristiques, je peux fusionner les ensembles de données en un seul DataFrame qui comprend à la fois les demandeurs d'emploi et les offres d'emploi. Je diviserai ensuite cet ensemble de données en ensembles de formation et de test pour former un modèle XGBoost.

[ ]:
```
import xgboost as xgb
from sklearn.model_selection import train_test_split

X = df[['experience_level', 'location_similarity', 'DegreeType_Associate\'s',␣
 ↪'DegreeType_Bachelor\'s', 'DegreeType_High School',
       'DegreeType_Master\'s', 'DegreeType_PhD', 'DegreeType_Vocational',
```

```
       'location_encoded', 'days_to_start', 'log_salary', 'job_req_encoded']]
y = df['match']


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

dtrain = xgb.DMatrix(X_train, label=y_train, enable_categorical=True)
dtest = xgb.DMatrix(X_test, label=y_test, enable_categorical=True)


params = {
    'max_depth': 20,
    'gamma': 15
}
model = xgb.train(params, dtrain, num_boost_round=100)
```

1.14 Conclusion

L'ingénierie des caractéristiques est une étape essentielle dans l'amélioration des performances des modèles d'apprentissage automatique, en particulier avec des algorithmes comme XGBoost. En transformant les données brutes en caractéristiques significatives, nous pouvons fournir au modèle de meilleures représentations du problème sous-jacent, ce qui conduit à une amélioration de la précision et du pouvoir prédictif. Et rappelez-vous, au fur et à mesure que vous expérimentez et itérez, l'ingénierie des caractéristiques peut souvent ressembler à un art autant qu'à une science, cela peut aussi être une malédiction - embrassez cette créativité et cette curiosité et n'abandonnez jamais.

[ ]:

[ ]:

[ ]: