Rapport de Projet : Prétraitement et Enrichissement du Dataset TED Talks

1. Source et Contexte du Jeu de Données

1.1 Origine et Description

Le jeu de données sélectionné provient de **Kaggle** et contient des informations sur **2 550 présentations TED Talks**. Ce dataset open-data représente une collection exhaustive de présentations issues de la plateforme TED, une organisation renommée pour la diffusion d'idées novatrices dans les domaines de la technologie, du divertissement, du design et des sciences.

Le choix de ce dataset s'explique par sa **richesse et sa complexité** : il présente des défis typiques de prétraitement de données réelles avec des variables de nature mixte (numériques, catégorielles, textuelles, temporelles) et des structures JSON complexes nécessitant un travail approfondi d'extraction et de transformation.

1.2 Structure Initiale des Données

Le dataset original comprend **17 variables** de types variés :

Variables Numériques :

- `comments` : Nombre de commentaires (2 à 6,404)
- `duration` : Durée en secondes (135 à 5,256 secondes)
- `languages` : Nombre de langues de traduction (0 à 72)
- `num_speaker` : Nombre d'intervenants (1 à 5)
- `views` : Nombre de vues (50,443 à 47,227,110)

Variables Temporelles :

- `film_date` et `published_date` : Timestamps Unix de tournage et publication

Variables Catégorielles :

- `event` : Nom de l'événement TED (355 modalités uniques)
- `main_speaker` : Nom de l'intervenant principal (2,156 valeurs uniques)
- `speaker_occupation` : Profession de l'intervenant (1,458 modalités)

Variables Textuelles et Structurées :

- `title`, `description` : Titre et description des présentations
- `tags` : Liste de thèmes sous format JSON
- `ratings` : Évaluations détaillées des spectateurs (structure JSON complexe)
- `related_talks` : Présentations connexes recommandées (JSON)

1.3 Objectifs du Prétraitement

L'objectif est de transformer ce dataset brut en données exploitables pour des analyses ultérieures, en respectant les principes de **traçabilité** et de **reproductibilité**. Le preprocessing vise à :

- Nettoyer et structurer les données complexes
- Créer des variables dérivées pertinentes
- Optimiser la représentation des informations pour l'analyse

2. Démarche Méthodologique

2.1 Exploration et Nettoyage Initial

Examen Dimensionnel : L'exploration initiale révèle un dataset de **2,550 observations × 17 variables** avec des types de données hétérogènes nécessitant un traitement spécifique pour chaque catégorie.

Analyse des Valeurs Manquantes :

- Identification de **6 valeurs manquantes** (0,24%) dans la variable `speaker_occupation`
- **Stratégie adoptée :** Suppression des lignes concernées car le taux de valeurs manquantes est négligeable et cette variable est critique pour l'analyse

Détection des Doublons : Aucun doublon détecté grâce à l'unicité naturelle des présentations TED.

Traitement des Structures Complexes :

```python
# Conversion des chaînes JSON en structures Python exploitables
df['tags'] = df['tags'].apply(lambda x: ast.literal_eval(x))
df['ratings'] = df['ratings'].apply(lambda x: ast.literal_eval(x))
df['related_talks'] = df['related_talks'].apply(lambda x: ast.literal_eval(x))
```

2.2 Transformation et Enrichissement (Feature Engineering)

Transformation des Variables Temporelles :

```python
```

```python
# Conversion des timestamps Unix en format datetime
df['film_date'] = pd.to_datetime(df['film_date'], unit='s')
df['published_date'] = pd.to_datetime(df['published_date'], unit='s')

# Extraction de variables temporelles dérivées
df['published_year'] = df['published_date'].dt.year
df['publish_month'] = df['published_date'].dt.month
df['publish_day'] = df['published_date'].dt.day
```

Simplification de Variables Catégorielles : La variable `event` (355 modalités) a été simplifiée en classification binaire :

```python
df['event'] = df['event'].apply(lambda x: 'TEDx' if 'TEDx' in x else 'TED')
```

Création de Variables Dérivées :

1. **Variables d'Engagement :**

```python
# Extraction des métriques d'engagement depuis les ratings JSON
df['num_ratings'] = df['ratings_list'].apply(lambda x: sum(d['count'] for d in x))
df['inspiring_ratings'] = df['ratings_list'].apply(lambda x: get_rating_count(x, 'Inspiring'))
df['funny_ratings'] = df['ratings_list'].apply(lambda x: get_rating_count(x, 'Funny'))
```

2. **Variables d'Intervenant :**

```python
# Calcul des vues moyennes par intervenant
speaker_view = df.groupby('main_speaker').agg({'views': 'mean'})
df['avg_speaker_view'] = df['main_speaker'].map(speaker_view_dict)
```

3. **Variables de Contenu :**

```python
# Transformation des tags en représentation textuelle pour TF-IDF
df['tags_str'] = df['tags'].apply(lambda x: ' '.join(x))
```

2.3 Gestion des Valeurs Aberrantes

Analyse des Distributions : Les variables `views`, `comments`, `duration` et `related_views` présentent des distributions fortement asymétriques avec des valeurs extrêmes significatives.

Stratégie de Transformation : Application d'une **transformation logarithmique** plutôt qu'une suppression d'outliers :

```python
# Transformation log1p pour gérer les valeurs nulles
skewed_cols = ['duration', 'comments', 'views', 'related_views']
for col in skewed_cols:
    df[col] = np.log1p(df[col])
```

Justification : Cette approche préserve l'information des contenus viraux tout en normalisant les distributions pour l'analyse.

2.4 Encodage et Mise à l'Échelle

Encodage des Variables Catégorielles :

1. **Target Encoding** pour les variables à haute cardinalité :

```python
# Application sur speaker_occupation (1,458 modalités)
target_encoder = ce.TargetEncoder(cols=['speaker_occupation', 'event'])
X_train_encoded = target_encoder.fit_transform(X_train, y_train)
```

2. **TF-IDF Vectorization** pour les variables textuelles :

```python
# Transformation des tags en features numériques
tfidf = TfidfVectorizer(stop_words='english', max_features=100)
tags_tfidf = tfidf.fit_transform(X_train['tags_str'])
```

Standardisation :

```python
```

```
# StandardScaler pour les variables numériques
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train[numerical_cols])
```

Principe de Non-Leakage : Tous les encodeurs et scalers sont ajustés uniquement sur les données d'entraînement puis appliqués aux données de test.

3. Sélection de Variables

3.1 Analyse de Variance et Corrélations

Matrice de Corrélation : L'analyse révèle des corrélations significatives :

- `comments` ↔ `views` : r = 0,53 (corrélation modérée positive)
- `languages` ↔ `views` : r = 0,38 (corrélation faible positive)
- `duration` ↔ `languages` : r = -0,29 (corrélation négative)

Suppression de Redondances : Élimination de variables redondantes comme `name` (combinaison de `main_speaker` + `title`) et des variables non-informatives (`url`, `description`).

3.2 Application de Méthodes de Sélection

Méthode Filter - Analyse Univariée : Suppression des variables à faible variance et des features TF-IDF peu discriminantes.

Méthode Wrapper - Forward Selection :

```python
# Sélection progressive basée sur l'amélioration du score MSE
selected_features_fw = []  # 49 features sélectionnées
for feature in remaining_features:
    score = cross_val_score(LinearRegression(), X_train[features], y_train, cv=5)
    # Sélection si amélioration du score
```

Méthode Embedded - Lasso Regression :

```python
# Régularisation L1 avec sélection automatique
lasso_cv = LassoCV(cv=5, random_state=42, max_iter=10000)
selected_features_lasso = X_train.columns[lasso_cv.coef_ != 0]  # 62 features
```

Méthode Embedded - Random Forest Importance :

```python
# Importance basée sur les arbres de décision
rf = RandomForestRegressor(n_estimators=100, random_state=42)
importances = rf.feature_importances_
selected_features_rf = features[importances > 0.005]  # 10 features principales
```

4. Principaux Résultats du Prétraitement

4.1 Transformation du Dataset

Évolution Structurelle :

- **Dataset initial :** 2,550 × 17 variables mixtes
- **Dataset final :** 2,544 × 115+ variables numériques exploitables
- **Réduction :** 6 observations supprimées (valeurs manquantes)
- **Enrichissement :** +98 nouvelles variables créées

Variables Finales Optimales : Les 15 variables les plus importantes identifiées :

Variable	Type	Importance	Description
`num_ratings`	Dérivée	0,536	Nombre total d'évaluations
`speaker_occupation`	Encodée	0,274	Profession (Target Encoded)
`languages`	Originale	0,037	Nombre de traductions
`publish_year`	Dérivée	0,021	Année de publication
`comments`	Transformée	0,009	Commentaires (log-transformés)

4.2 Qualité des Transformations

Normalisation des Distributions :

- **Avant transformation :** Distributions fortement asymétriques (skewness > 3)
- **Après log-transformation :** Distributions quasi-normales (skewness < 1)

Efficacité de l'Encodage :

- **Target Encoding :** Réduction de 1,458 modalités (`speaker_occupation`) à 1 variable numérique
- **TF-IDF :** Transformation de 2,530 tags uniques en 100 features optimisées

4.3 Validation de la Qualité

Absence de Data Leakage : Validation par séparation train/test avant tout processus d'ajustement des transformations.

Reproductibilité : Utilisation de `random_state` constants et documentation complète de chaque étape.

5. Justification des Choix Opérés

5.1 Stratégies de Nettoyage

Suppression vs Imputation : Choix de la suppression des valeurs manquantes (0,24%) plutôt que l'imputation pour préserver l'intégrité des données de professions d'intervenants.

Conservation des Outliers : Transformation logarithmique préférée à la suppression car les vidéos virales représentent des cas d'usage légitimes et informatifs.

5.2 Approches d'Encodage

Target Encoding vs One-Hot : Pour `speaker_occupation` (1,458 modalités), le Target Encoding évite l'explosion dimensionnelle tout en capturant la relation avec la variable cible.

TF-IDF vs Bag-of-Words : TF-IDF choisi pour les tags afin de pondérer l'importance relative des thèmes selon leur fréquence globale.

5.3 Méthodes de Sélection

Approche Multi-Méthodes : Combinaison de trois approches (Filter, Wrapper, Embedded) pour validation croisée des résultats et robustesse de la sélection.

Seuils de Sélection :

- Random Forest : seuil d'importance > 0,005 (balance précision/parsimonie)
- Lasso : alpha optimal via CV (0,00064) pour régularisation optimale

6. Conclusion et Impact

6.1 Achievements du Prétraitement

Le processus de prétraitement a permis de :

- **Structurer** des données JSON complexes en variables exploitables
- **Enrichir** le dataset de +98% de variables dérivées pertinentes
- **Optimiser** la représentation pour des analyses ultérieures

- **Garantir** la reproductibilité et la traçabilité de chaque transformation

6.2 Dataset Final Exploitable

Le dataset prétraité est désormais prêt pour :

- **Analyses statistiques** avancées avec variables normalisées
- **Modélisation prédictive** avec features optimisées
- **Analyses de clustering** sur variables d'engagement
- **Études temporelles** avec variables chronologiques extraites

6.3 Bonnes Pratiques Appliquées

- **Séparation train/test précoce** (prévention du data leakage)
- **Documentation systématique** de chaque transformation
- **Validation multiple** des méthodes de sélection
- **Préservation de l'information** via transformations plutôt que suppressions

Cette démarche méthodologique rigoureuse garantit un dataset de haute qualité, exploitable pour des analyses robustes et des insights métier significatifs.