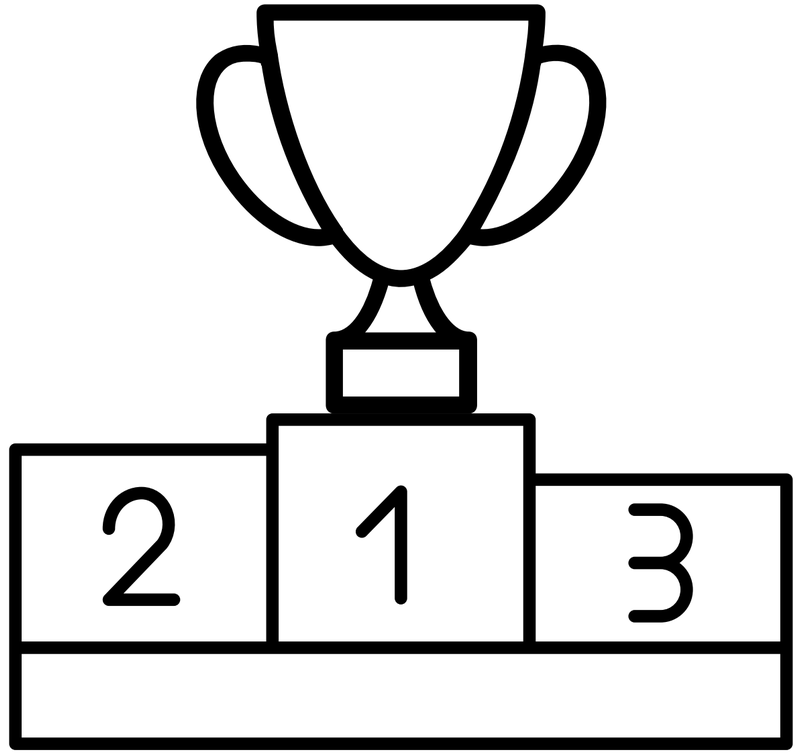
Cécile Huang & Alice Liu

Kaggle compétition

Start



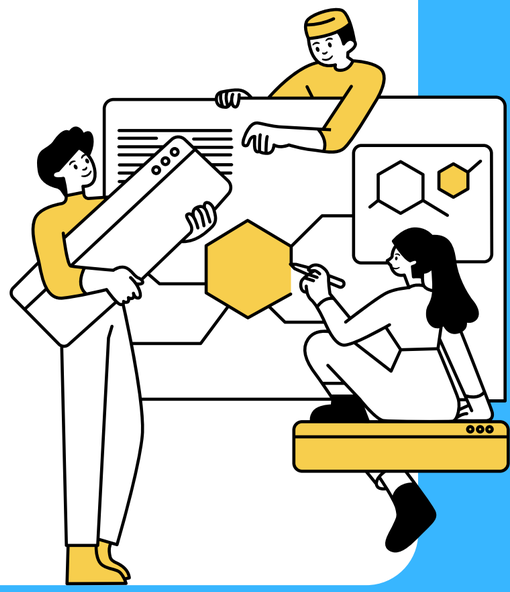
Sujet

Estimate CO2 emissions from cars in europe

Nous avons trois fichiers:

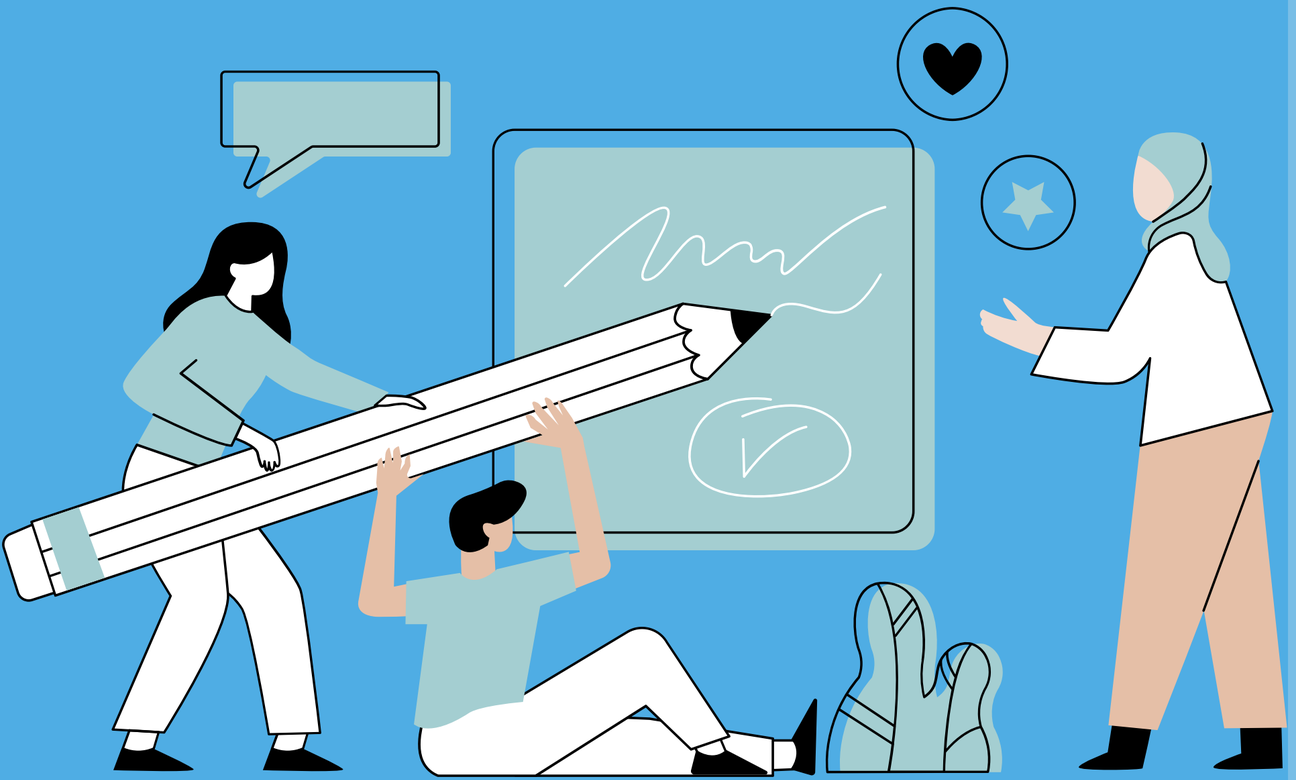
- un fichier train,
- un fichier test,
- un fichier submission.

A partir du fichier train et test nous devons créer un modèle de prédiction qui va s'entrainer sur le train et prédire sur le test. Pour évaluer la performance de notre modèle nous devons utiliser la métrique MAE qui mesure l'erreur moyenne absolue que fait notre modèle lorsqu'elle prédit.
pour soumettre notre modèle sur Kaggle nous devons créer un fichier submission.



01

Plan Experimental



02

Modèle retenu

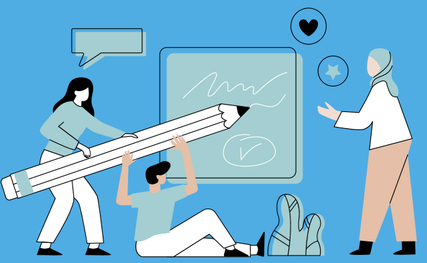
03

Résultats

04

Conclusion

01
Plan Experimental

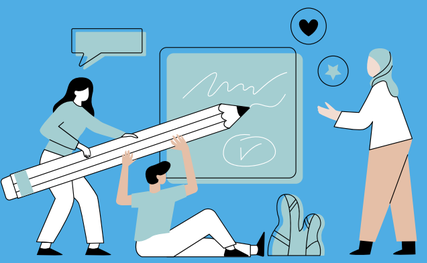


02
Modèle retenu

03
Résultats

04
Conclusion

01
Plan Experimental

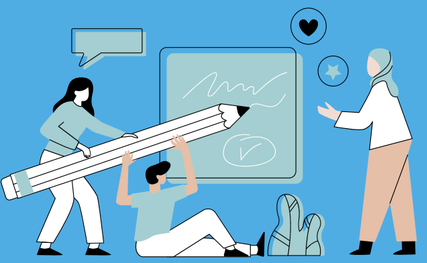


02
Modèle retenu

03
Résultats

04
Conclusion

01
Plan Experimental



02
Modèle retenu

03
Résultats

04
Conclusion

Plan experimental

Prototype Modèles Encodage Paramètres Autres

Etape 1: Premier modèle prototype

- Suppression des variables avec un taux de valeurs manquantes >50%.
- Récupération des valeurs numériques seulement.
- Modèle linéaire.
- Première soumission.

Plan experimental

Prototype **Modèles** Encodage Paramètres Autres

Etape 2: Plusieurs modèles

On a testé trois modèles:

- Régression linéaire
- XGBoost
- RandomForest

Nous avons eu de bon score MAE avec Random forest donc nous avons insistés sur ça.

Plan experimental

Prototype Modèles **Encodage** Paramètres Autres

Etape 3: Encodage et sélection de variables

L'encodage par OneHot :
- Impossible à utiliser, trop de modalités.

L'encodage par LabelEncoder:
- Nouvelles modalités dans test qui ne sont pas dans le train.
- Rassemblement des modalités de df_train et df_test.

L'encodage par Catboostencoder:
- fonctionne, n'a pas le problème de nouvelles modalités ni trop de modalités.

Sélection de variables numériques puis catégorielles.

Plan experimental

Prototype Modèles Encodage **Paramètres** Autres

Etape 4: Les paramètres de Randomforest

Gridsearch trop long pour des ordinateurs pas assez puissants

Les paramètres sélectionnés:
- Le nombre d'arbres
- La profondeur maximale de chaque arbre de la forêt
- Le nombre minimum d'échantillons nécessaires pour diviser un nœud
- Le nombre maximum de variables par noeuds ici 77% des variables

On fait varier notre manière de réaliser le préprocessing
- Traitement des outliers ou pas
- Traitement des valeurs manquantes
- Feature Engineering
- Cross validation
- Gridsearch

Plan experimental

Feature Engineering

```
"puissance_masse"] = df_train["ep (KW)"]/df_train["m (kg)"]
"puissance_capacite"] = df_train["ep (KW)"]/df_train["ec (cm3)"]

"efficacite_energie"] = df_train["Electric range (km)"]/df_train["z (Wh/km)"]
"efficacite_energie"].fillna(0, inplace=True)

'Fuel_Details'] = df_train['Ft'] + '_' + df_train['Fm']
Fuel_Details'] = df_test['Ft'] + '_' + df_test['Fm']
```

Tan
e1*2007/46*1918*13
e1*2018/858*00005*02
e1*2007/46*1680*08
e2*2007/46*0639*14
e13*2007/46*2188*07

Plan experimental

Prototype Modèles Encodage Paramètres **Autres**

Etape 5: On test à nouveau d'autres modèles

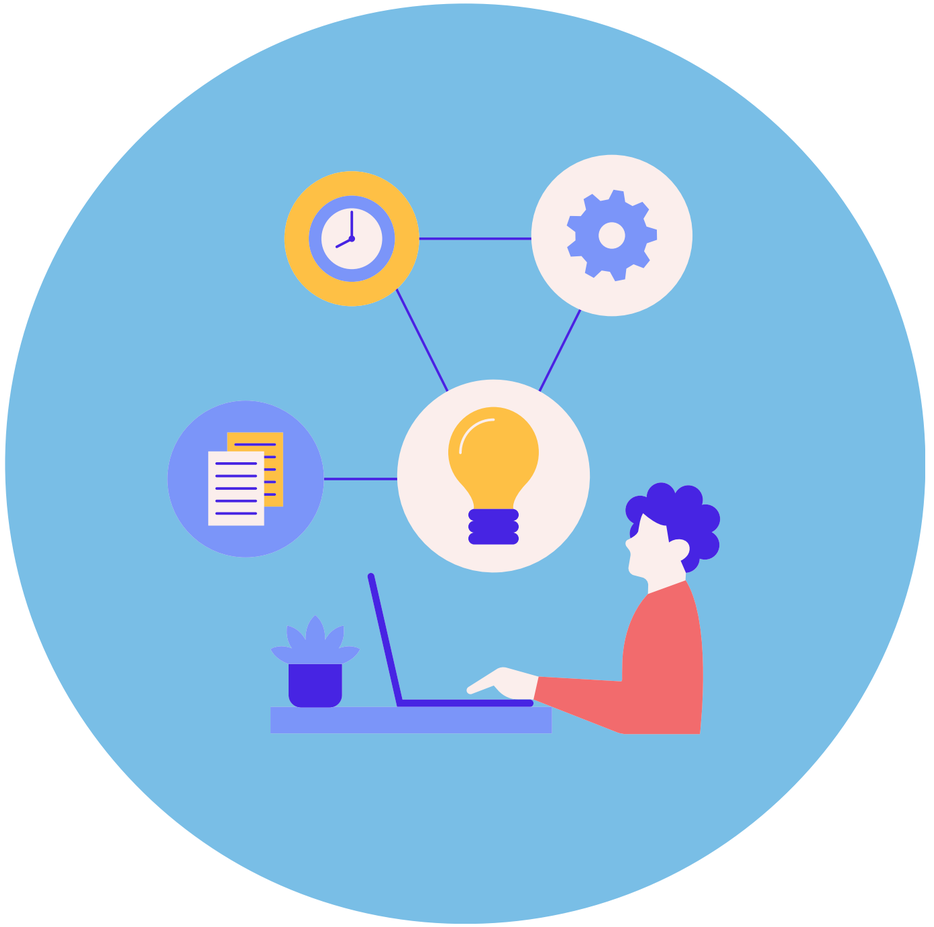
Notre modèles de RandomForest a atteint une limite d'environ 2.81 à cause du nombre de paramètres. la MAE ne descendait plus assez même avec la cross validation.

Nouvelle stratégie:

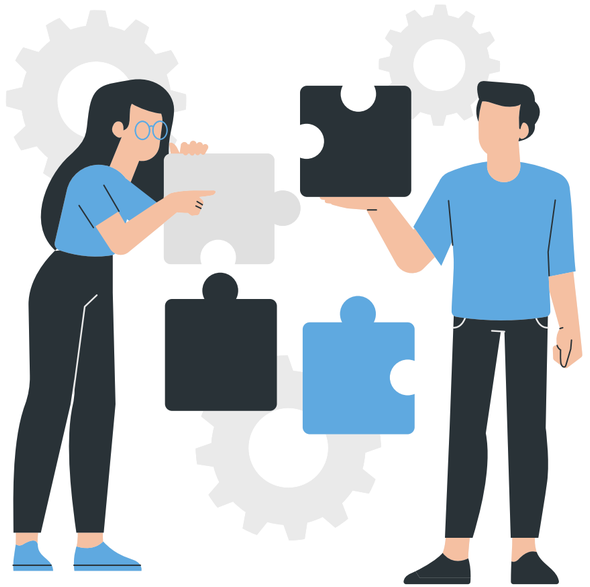
- Faire varier les modèles (XGB, LightGBM)
- Faire varier la méthode d'encodage (CountEncoder, OrdinalEncoder, TargetEncoder, CatBoostEncoder, LeaveOneOutEncoder)

Le modèle retenu

Save Version | 100



Pré-processing



Valeurs manquantes

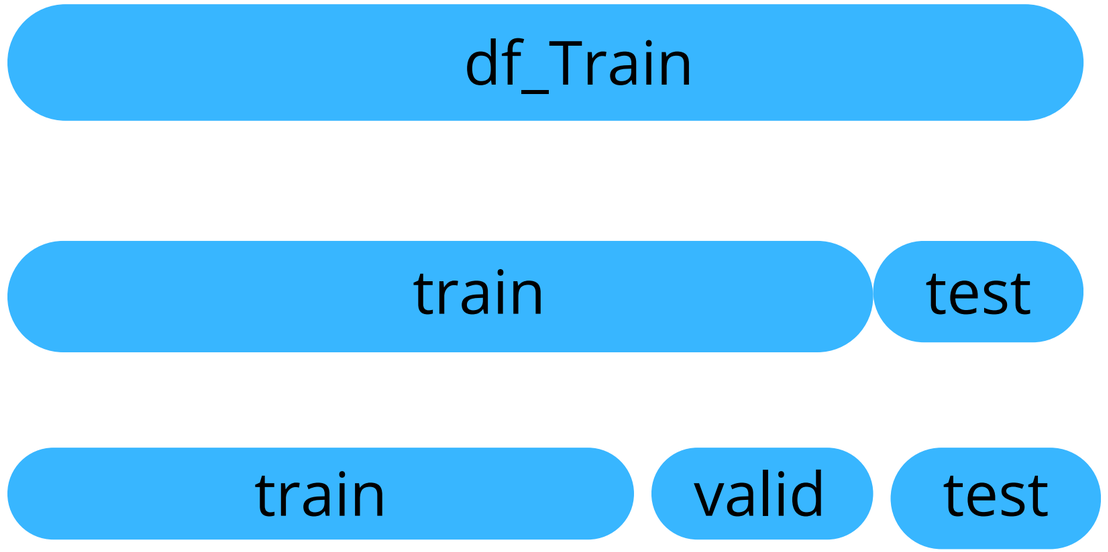
- Seuil à 50% mais on garde certaines variables qui dépassent ce seuil
- Sélections variables (numériques et catégorielles)
- Traitement spécifique pour Z (Wh/km) et Electric range (km).

Encodage

On garde notre Label Encoder modifié.

Pour vérifier la robustesse de cette méthode nous avons crée un notebook qui vient split notre df_train.

Ce notebook NB_TTS_sur_train est excusivement utilisé pour vérifier le surajustement.



Train test split

Par manque de temps et de ressources matérielles la cross validation ne passait pas.
Utilisation du train test split classique.

Cross Validation

```python
params = {'n_estimators' : 50,
          'min_samples_split' : 12,
          'max_depth' : 35,
          'max_features'  : 0.77,
          'random_state' : 2023,
          'n_jobs' : 4}
models=[]

# Nombre de folds pour la validation croisée
num_folds = 10
kf = KFold(n_splits=num_folds, shuffle=True, random_state=42)

# Initialiser une liste pour stocker les prédictions sur le jeu de test de chaque fold
validation_predictions = []
df_test_predictions = []


for train_index, val_index in kf.split(X_train):
    X_train_fold, X_val_fold = X_train[train_index], X_train[val_index]
    y_train_fold, y_val_fold = y_train[train_index], y_train[val_index]

    # Créer et entraîner le modèle RF
    model = RandomForestRegressor(**params)
    model.fit(X_train_fold, y_train_fold)
    models.append(model)
    # Faire des prédictions sur le jeu de validation
    val_preds = model.predict(X_val_fold)

    # Calculer la MAE pour le fold actuel
    fold_mae = mean_absolute_error(y_val_fold, val_preds)
    print(f'MAE for this fold: {fold_mae}')

    # Faire des prédictions sur le jeu de validation
    validation_fold_preds = model.predict(X_validation)
    validation_predictions.append(validation_fold_preds)

    # Faire des prédictions sur le jeu de df_test
    df_test_fold_preds = model.predict(X_test)
```

```python
    # Calculer la moyenne pondérée des prédictions sur le jeu de validation
    final_validation_predictions = np.mean(np.array(validation_predictions), axis=0)

    # Calculer la moyenne pondérée des prédictions sur le jeu de df_test
    final_df_test_predictions = np.mean(np.array(df_test_predictions), axis=0)

    # Évaluer la MAE sur le jeu de validation complet
    final_mae = mean_absolute_error(y_validation, final_validation_predictions)
    print(f'\nFinal MAE on validation set: {final_mae}')
```

```
MAE for this fold: 2.83379544516151
MAE for this fold: 2.827756886122789
MAE for this fold: 2.8261848298412975
MAE for this fold: 2.8195444726324483
MAE for this fold: 2.8266695463587035
MAE for this fold: 2.8113255933340495
MAE for this fold: 2.828890041403032
MAE for this fold: 2.8166533571187187
MAE for this fold: 2.8157087716284863
MAE for this fold: 2.8200166451229487

Final MAE on validation set: 2.8107620548755214
```

Paramètres du modèle

Les paramètres sélectionnés:

- "objective": "reg:squarederror",
- "max_depth":30,
- "learning_rate":0.005,
- "min_split_loss" : 10,
- "random_state" : 42,
- "min_child_weight" : 1
- "max_delta_step" : 0,
- "subsample" : 1,
- "max_leaves" : 0,
- "max_bin" : 800,
- "num_parallel_tree" : 1,
- "reg_alpha" : 0.8,
- "reg_lambda" : 0.15,
- "colsample_bytree":0.85,
- "colsample_bynode" : 0.85,
- "colsample_bylevel" : 0.85,
- 'eval_metric': "mae",

- n=3000
- verbose=200
- early_stopping_rounds=20

Résultats

kaggle

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

Estimate CO2 emissions from cars

Submit Prediction

Overview Data Code Models Discussion Leaderboard Rules Team Submissions

△	Team	Members	Score	Entries
—	Jynaldo & Yoan		2.7834	49
▲ 1	**Cécile & Alice**		2.7872	35
▼ 1	Khidour Alae		2.7886	28
▲ 1	Mélanie & Johanna		2.7924	42
▲ 1	Mouad & Alhassane		2.7935	51
▼ 2	walid		2.7936	37

Conclusion et axes d’améliorations

Merci

Alice Liu & Cécile Huang