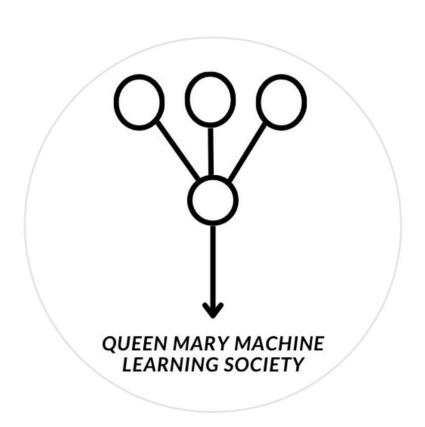
Kaggle Seasons #04



Encoding - Categorical

- What is Categorical Data?
 - Ordinal (e.g. Shirt sizes: Small < Medium < Large)
 - Nominal (e.g., Gender: Male/Female)
- How to Encode it?
 - Label Encoding (Mapping)
 - One-Hot-Encoding



Template: Load & Preprocess

Imports and Installs

Data Loading

(Optional) Data Enhancement

(Optional) EDA (Explorative Data Analysis)

- Data Preprocessing
- Data Cleaning

Missing Value Handling

Encoding

(Optional) Feature Engineering

Train-Test Split

Normalisation



Template: Model Handling

Model Handling

Prediction

Submission File Creation

Model Selection Model Evaluation # Model training # Model predictions # Performance summary ### Recommended: complete the model training and model prediction during cross-validation Model Tuning Hyperparameter Optimisation (Optional) Parameter Summary (Optional) Ensemble Learning Training **Fvaluation** Submission Data Preprocessing



Sample Submission (Pt. 1)

Imports and Installs

```
[] import pandas as pd
import numpy as np
from sklaenr.neprocessing import OneHotEncoder
from sklaenr.model_selection import train_test_split, StratifiedKFold
from sklaenr.model_selection import accuracy_score
from sklaenr.linear_model import LogisticRegression
```

Data Loading

```
[ ] data = pd.read_csv('train.csv')
  test_data = pd.read_csv('test.csv')

[ ] data = data.drop(columns=['Name', 'isd'])
  test_data = test_data.drop(columns=['Name'])
```

Data Preprocessing

Data Cleaning

```
for d in [data, test_data]:
for column in d.columns:
    if d.isma().sum()[column] > 0:
    mode_value = d[column].dropna().mode()[0]
    d[column].fillna(mode_value, inplace=True)
```

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
d[column].fillna(mode_value, inplace=True)
```

Encoding

```
[] nominal_cols = data_select_dtypes(includes['object', 'category']).columns.tolist()
ohe = OneMotEncoder(sparse_output=false, handle_unknoam='ignore')
data_ohe = pd_nateFaree(low_fit_transform(data_nominal_cols), columns=ohe.get_feature_names_out(nominal_cols), index=data.index)
test_data_ohe = pd_nateFaree(low_transform(test_data_nominal_cols)), columns=ohe.get_feature_names_out(nominal_cols), index=test_data.index)
data = data_drop(columns=nominal_cols).join(data_ohe)
test_data = test_data_drop(columns=nominal_cols).join(test_data_ohe)
```



Sample Submission (Pt. 2)

Train-Test Split

Model Handling

Model Selection

[] model = LogisticRegression(random_state=42)

Model Evaluation

```
[] def cross_validate(mode), X_train, y_train, cv=5):
    skf = StratifiedRofLd(n_splits=cv, shuffle=True, random_state=42)
    accuracy_scores = []
    for train_index, test_index in skf.split(X_train, y_train):
    # Model training
    X_train_fold, X_test_fold = X_train.iloc[train_index], X_train.iloc[test_index]
    y_train_fold, Y_test_fold = X_train.iloc[train_index], y_train.iloc[test_index]
    wodel.fit(X_train_fold, y_train_fold)

# Model predictions
    y_pred = model.predict(X_test_fold)
    accuracy = accuracy_score(y_test_fold, y_pred)
    accuracy_scores.append(accuracy)
# Perforance summary
    return np.mean(accuracy_scores)
cross_validate(model, X_train, y_train)
```

kaggle

Sample Submission (Pt. 3)

- Model Tuning
- Hyperparameter Tuning

```
[] params = {
    'max_iter': 1000,
    'class_weight': 'balanced'
    }
```

- Re-evaluation
- [] model_ht = LogisticRegression(**params)
 cross_validate(model, X_train, y_train)
- → 0.9382871357498223
- Submission
- [] # Without hyperparameter tuning
 model.fit(X_train, y_train)
 test_predictions = model.predict(X_test.drop(columns=['id']))
 submission = pd.DataFrame({'id': X_test["id"], 'Depression': test_predictions}))
 submission.to_csv('basic_submission.csv', index=False)
- [] # With hyperparamter tuning
 model_ht.fit(X_train, __train)
 test_predictions = model_ht.predict(X_test.drop(columns=['id']))
 submission = pd.DataFrame(('id': X_test["id"], 'Depression': test_predictions})
 submission.to_csv('tuned_submission.csv', index=False)



Good luck!

