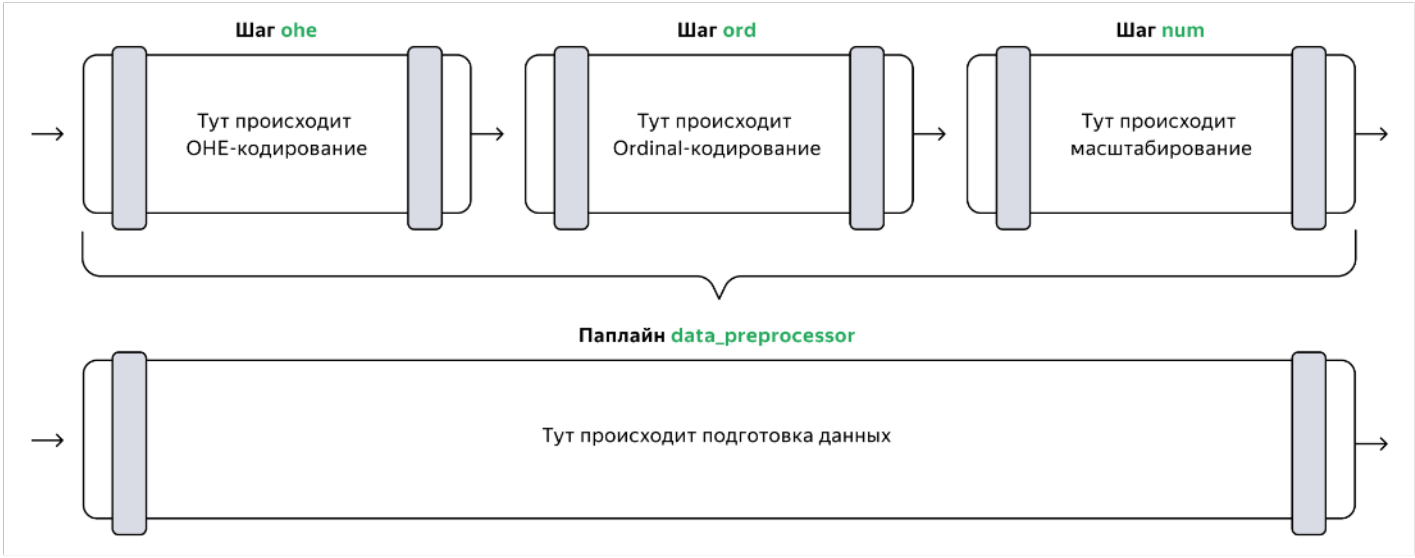
Пайплайн обучения

Пайплайн, конвейер, — это инструмент, который позволяет автоматизировать действия по подготовке данных, обучению моделей и оценке их качества. Схема объединения шагов по подготовке данных в один пайплайн:



Создание пайплайна для подготовки данных

```
from sklearn.pipeline import Pipeline
# класс ColumnTransformer работает с данными разного типа в одном наборе
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder, StandardScaler
from sklearn.impute import SimpleImputer

# признаки для OneHotEncoder
ohe_columns = ['Пол', 'Путешествует по работе']

# признаки для OrdinalEncoder
ord_columns = ['Оценка качества wifi']

# признаки для масштабирования
num_columns = ['Возраст', 'Расстояние']
```

```
# SimpleImputer + OHE
ohe_pipe = Pipeline(
    [
        (
            'simpleImputer_ohe',
            SimpleImputer(missing_values=np.nan, strategy='most_frequent')
        ),
```

Пайплайн обучения

```
        (
            'ohe',
            OneHotEncoder(drop='first', handle_unknown='ignore', sparse=False)
        )
    ]
)

# SimpleImputer + ORD + SimpleImputer
ord_pipe = Pipeline(
    [
        (
            'simpleImputer_before_ord',
            SimpleImputer(missing_values=np.nan, strategy='most_frequent')
        ),
        (
            'ord',
            OrdinalEncoder(
                categories=[
                    ['нормально', 'хорошо', 'плохо', 'отсутствует'],
                ],
                handle_unknown='use_encoded_value', unknown_value=np.nan
            )
        ),
        (
            'simpleImputer_after_ord',
            SimpleImputer(missing_values=np.nan, strategy='most_frequent')
        )
    ]
)

# объединение шагов подготовки
data_preprocessor = ColumnTransformer(
    [
        ('ohe', ohe_pipe, ohe_columns), # применение пайплайна ohe_pipe к данным
ohe_columns
        ('ord', ord_pipe, ord_columns), # применение пайплайна ord_pipe к данным
ord_columns
        ('num', StandardScaler(), num_columns) # масштабирование num_columns
    ],
    remainder='passthrough' # не применять шаги к данным вне списков
)
```

Пайплайн обучения

Создание итогового пайплайна

```
from sklearn.pipeline import Pipeline
from sklearn.tree import DecisionTreeClassifier

# итоговый пайплайн: подготовка данных и модель
pipe_final = Pipeline(
    [
        ('preprocessor', data_preprocessor),
        ('models', DecisionTreeClassifier(random_state=RANDOM_STATE))
    ]
)
```

Применение пайплайна к данным

```
# обучение модели на тренировочной выборке
pipe_final.fit(X_train, y_train)

# вывод предсказанных значений тренировочной выборки
y_train_pred = pipe_final.predict(X_train)
print(f'Предсказание на обучающей выборке: {y_test_pred}')

# применение обученной модели на тестовой выборке
y_test_pred = pipe_final.predict(X_test)
print(f'Метрика ROC-AUC на тестовой выборке: {roc_auc_score(y_test, y_test_pred)}')
```