Data preprocessing

Дубликаты

1. Выбрасывать
2. Группировать

```
data = data.drop_duplicates()
```

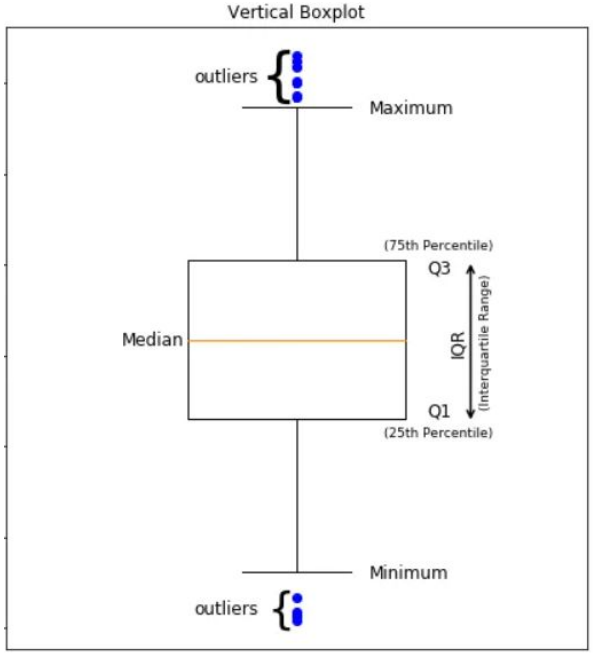
Пропущенные значения

1. Константное значение
2. Среднее
3. Предсказания другого алгоритма(knn, линейка, матрицы)
4. Удалить столбец

```
data[num_cols] = data[num_cols].fillna(data[num_cols].mean())
```

Выбросы

Чаще всего стоит удалять ориентируюсь на распределения признаков, ошибки, целевой переменной



Категориальные

1. LabelEncoding
2. OHE
3. TargetEncoding

original dataset

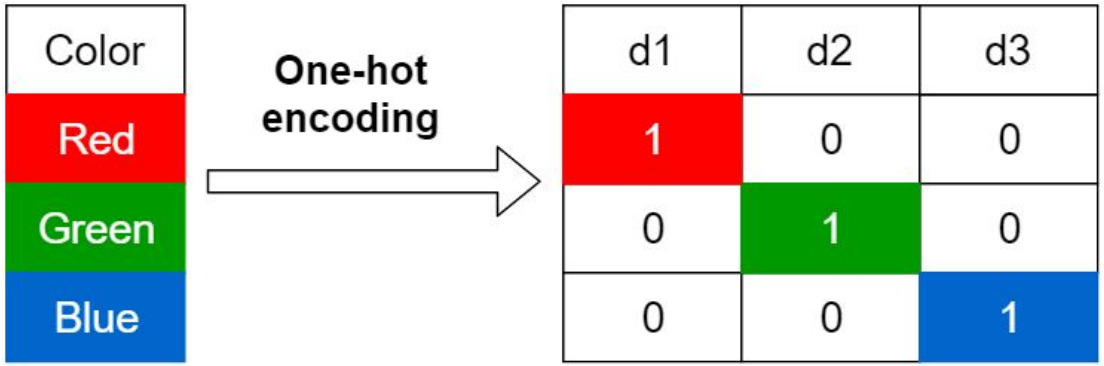
x_1	x_2	y
5	8	calabar
9	3	uyo
8	6	owerri
0	5	uyo
2	3	calabar
0	8	calabar
1	8	owerri

LabelEncoder

```
{
  "calabar" ---> 0
  "owerri"  ---> 1
  "uyo"     ---> 2
}
```

dataset with encoded labels

x_1	x_2	y
5	8	0
9	3	2
8	6	1
0	5	2
2	3	0
0	8	0
1	8	1


Color
Red
Green
Blue
One-hot encoding
d1
d2
d3
1
0
0
0
1
0
0
0
1

original dataset

x_1	x_2	y
5	8	calabar
9	3	uyo
8	6	owerri
0	5	uyo
2	3	calabar
0	8	calabar
1	8	owerri

LabelEncoder

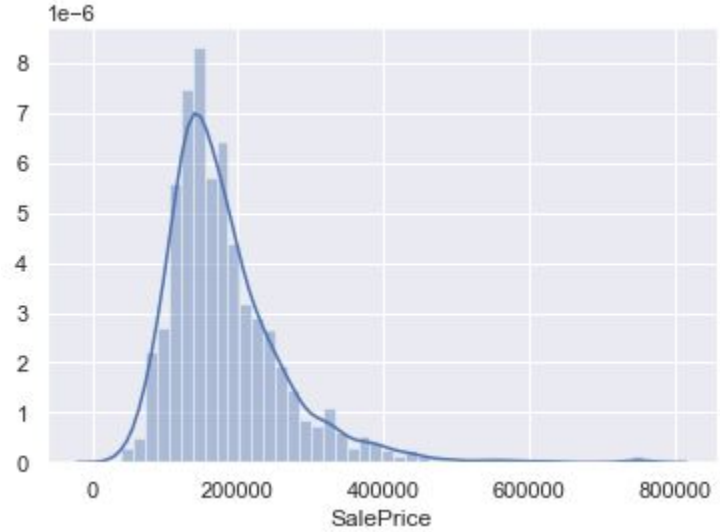
```
{
  "calabar" ---> 0
  "owerri"  ---> 1
  "uyo"     ---> 2
}
```

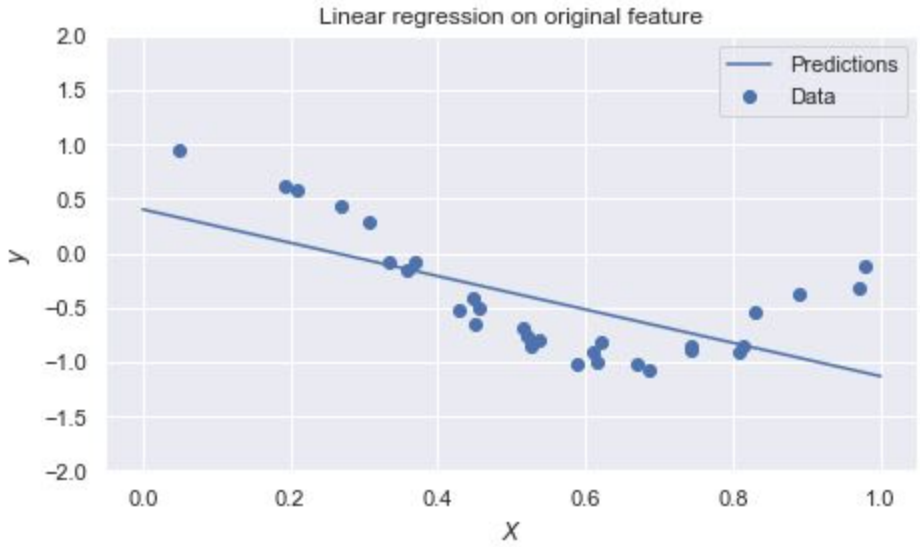
dataset with encoded labels

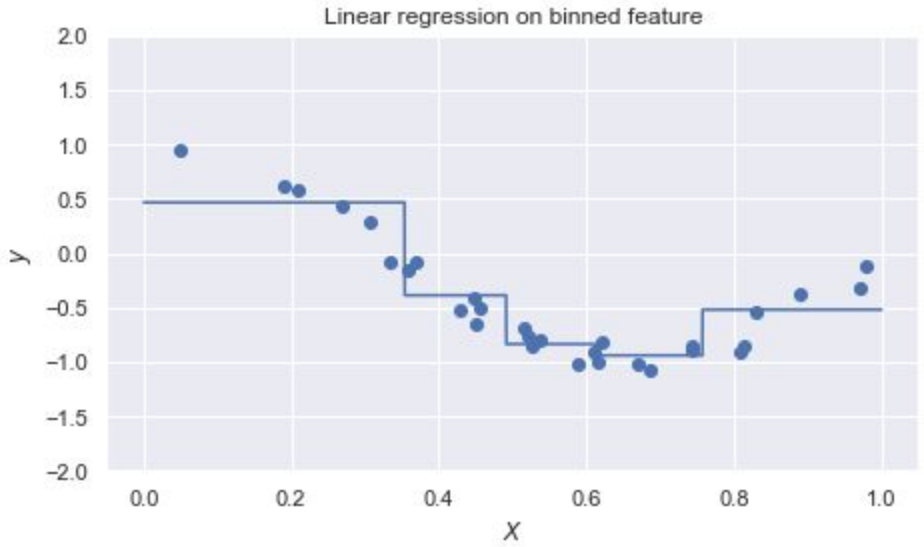
x_1	x_2	y
5	8	0
9	3	2
8	6	1
0	5	2
2	3	0
0	8	0
1	8	1

Числовые

1. Нормировка
2. Логарифмирование
3. Бинаризация


1e−6
8
7
6
5
4
3
2
1
0
0
200000
400000
600000
800000
SalePrice


Linear regression on original feature
Predictions
Data
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
0.0
0.2
0.4
0.6
0.8
1.0
X


Linear regression on binned feature
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
0.0
0.2
0.4
0.6
0.8
1.0
X

```
num_cols = data.select_dtypes([np.number]).columns
cat_cols = data.select_dtypes(object).columns
```

```
mask = (error < np.quantile(error, 0.95))
X_train = X_train[mask]
y_train = y_train[mask]
```

```
column_transformer = ColumnTransformer([
    ('ohe', OneHotEncoder(handle_unknown="ignore"), categorical),
    ('scaling', StandardScaler(), numeric_features)
])

pipeline = Pipeline(steps=[
    ('ohe_and_scaling', column_transformer),
    ('regression', Ridge())
])

model = pipeline.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Test RMSE = %.4f" % mean_squared_error(y_test, y_pred, squared=False))
```

Какие признаки оставлять?

1. Корреляция
2. Веса
3. Feature_imporance
4. Удалять признаки и обучать модель
5. Методы фильтрации

Как придумать новые признаки?

1. Посмотреть на интересные признаки(pickup_datetime)
2. AutoFeat

Все зависит от модели и метрики