Московский государственный технический университет им. Н. Э. Баумана

Курс «Технологии машинного обучения»

Отчёт по лабораторной работе

Выполнил
Бокатуев М. С.
группа ИУ5-62Б

Проверил:
Гапанюк Ю.Е.

Дата: 04.06.25

Дата:

Подпись:

Подпись:

Москва, 2025 г.

Задание:

Выбрать набор данных (датасет), содержащий категориальные признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.)

Для выбранного датасета (датасетов) на основе материалов лекции решить следующие задачи:

- обработку пропусков в данных;
- кодирование категориальных признаков;
- масштабирование данных.

Загрузка и первичный анализ данных

Используем данные из датасета Titanic

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
from sklearn.impute import KNNImputer
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Lasso
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from IPython.display import Image
%matplotlib inline
sns.set(style="ticks")
```

In [ ]:
```
hdata_loaded = pd.read_csv('sample_data/Titanic-Dataset.csv', sep=",")
hdata_loaded.shape
```

Out[ ]:
```
(891, 12)
```

In [ ]:
```
hdata = hdata_loaded
```

In [ ]:
```
hdata.head()
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/ O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450

In [ ]:
```
list(zip(hdata.columns, [i for i in hdata.dtypes]))
```

Out[ ]:
```
[('PassengerId', dtype('int64')),
 ('Survived', dtype('int64')),
 ('Pclass', dtype('int64')),
 ('Name', dtype('O')),
 ('Sex', dtype('O')),
 ('Age', dtype('float64')),
 ('SibSp', dtype('int64')),
 ('Parch', dtype('int64')),
 ('Ticket', dtype('O')),
 ('Fare', dtype('float64')),
 ('Cabin', dtype('O')),
 ('Embarked', dtype('O'))]
```

In [ ]:
```
hcols_with_na = [c for c in hdata.columns if hdata[c].isnull().sum() > 0]
hcols_with_na
```

Out[ ]:
```
['Age', 'Cabin', 'Embarked']
```

In [ ]:
```
[(c, hdata[c].isnull().sum()) for c in hcols_with_na]
```

Out[ ]:
```
[('Age', 177), ('Cabin', 687), ('Embarked', 2)]
```

In [ ]:
```
[(c, hdata[c].isnull().mean()) for c in hcols_with_na]
```

Out[ ]:
```
[('Age', 0.19865319865319866),
 ('Cabin', 0.7710437710437711),
 ('Embarked', 0.002244668911335578)]
```

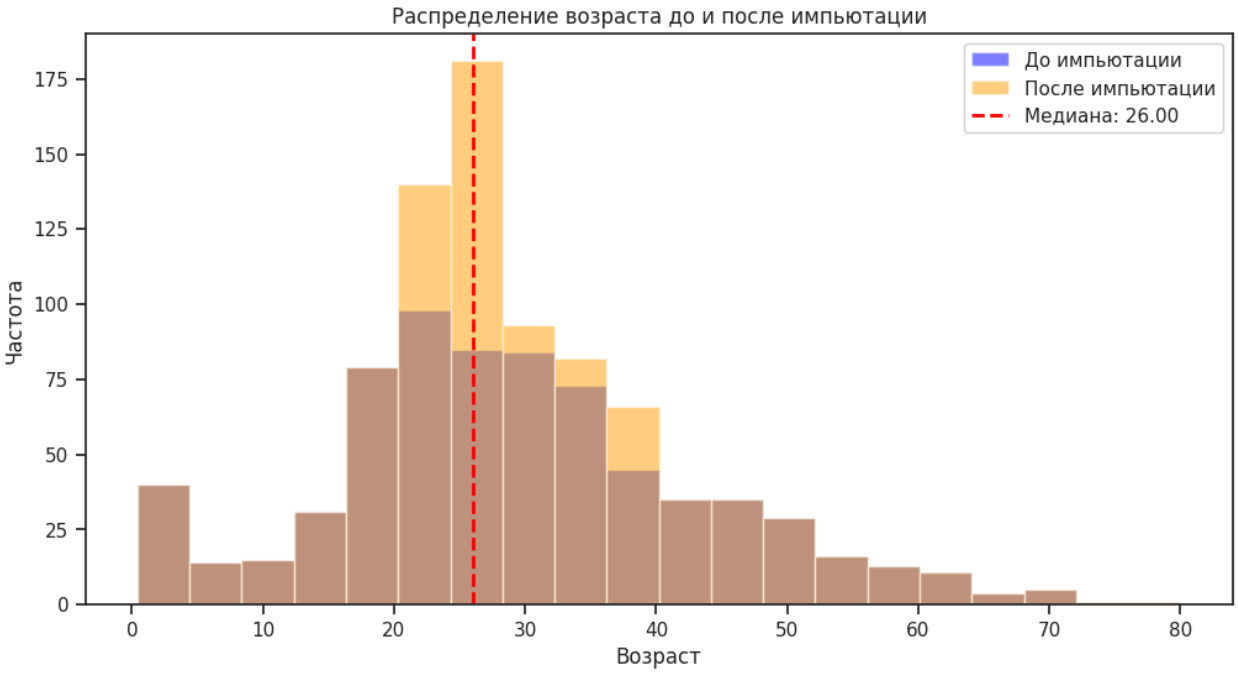
Обработка пропусков в Age

Пропуски составляют около 19.87% данных.

In [ ]:
```
def fill_age_groupwise(df):
    # Для каждого значения Pclass и Sex заполняем пропуски медианой
    hdata['Age'] = df.groupby(['Pclass', 'Sex'])['Age'].transform(lambda x: x.
    return df['Age'].isnull().sum()
before_imputation = hdata["Age"].copy()
# Заполнение пропусков
fill_age_groupwise(hdata)
```

Out[ ]:
```
0
```

In [ ]:
```
# Рассчитываем процент пропущенных значений в колонке Age
missing_percentage_before = hdata_loaded["Age"].isnull().mean() * 100
missing_percentage_after = hdata["Age"].isnull().mean() * 100

# Построение гистограммы с выделением изменений
plt.figure(figsize=(12, 6))

# До импьютации
plt.hist(before_imputation.dropna(), bins=20, alpha=0.5, label='До импьютации'

# Отображаем только импьютированные значения
plt.hist(hdata["Age"][hdata["Age"].isnull() == False], bins=20, alpha=0.5, lab

# Добавляем информацию о медиане
plt.axvline(hdata["Age"].median(), color='red', linestyle='dashed', linewidth=

# Подписи и легенда
plt.title(f'Распределение возраста до и после импьютации')
plt.xlabel('Возраст')
plt.ylabel('Частота')
plt.legend()

# Отображаем график
plt.show()
```



Обработка пропусков в Cabin

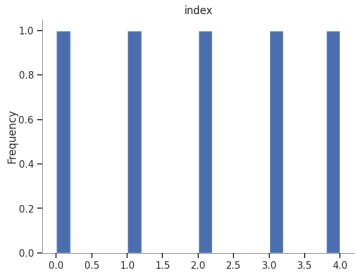
Пропуски составляют около 77.10% данных.

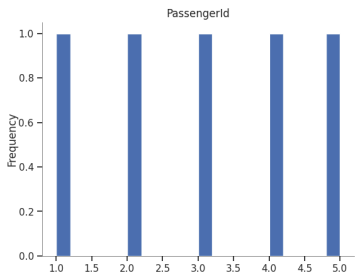
In [ ]:
```
hdata['Cabin'] = hdata['Cabin'].apply(lambda x: x[0] if pd.notna(x) else 'U')
hdata.head()
```

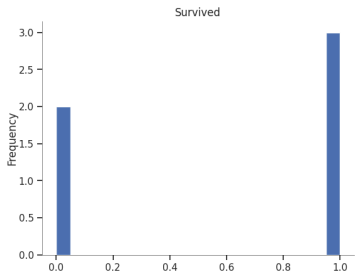
Out[ ]:

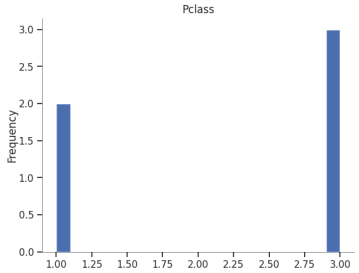
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/ O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450

Distributions

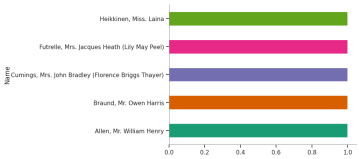


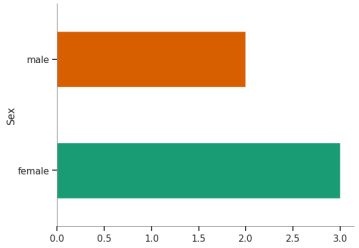


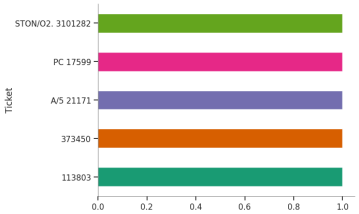


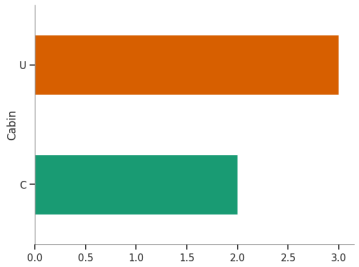


Categorical distributions

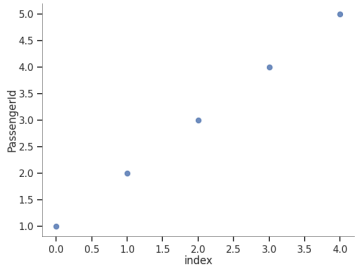


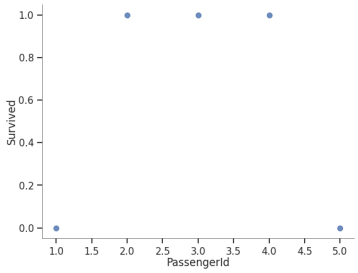


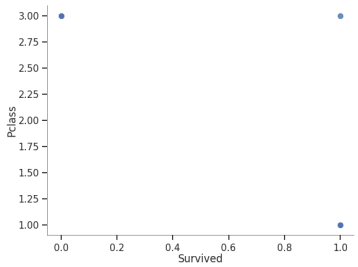


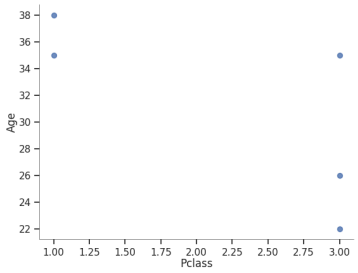


2-d distributions

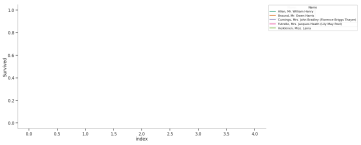


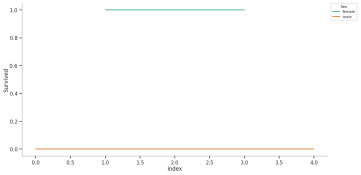

Survived
PassengerId

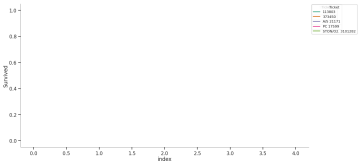

Pclass
Survived

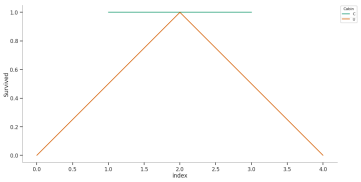

Age
Pclass

Time series

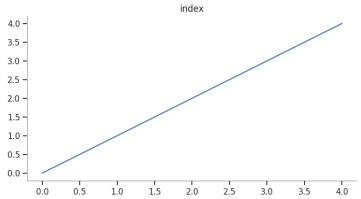


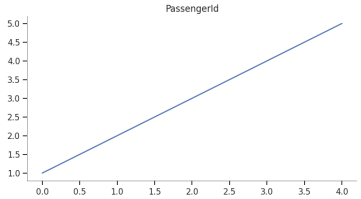

Survived
index

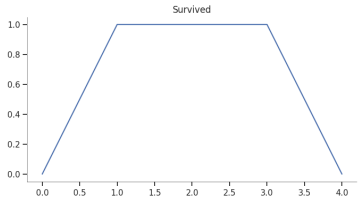

Survived
index

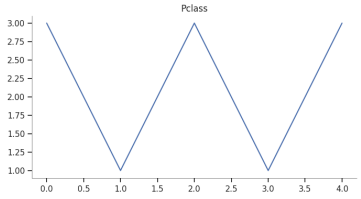

Survived
index

Values

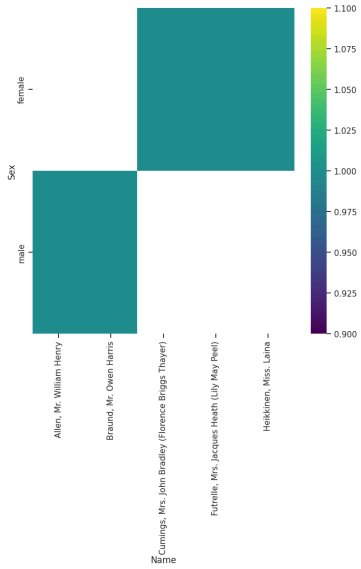


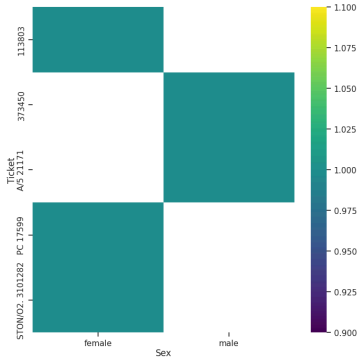


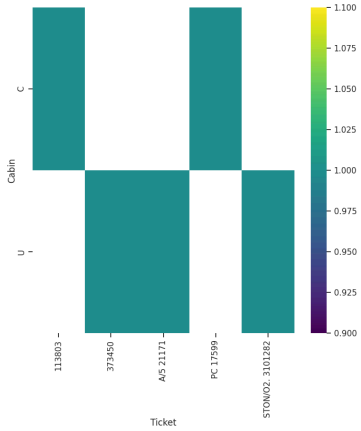


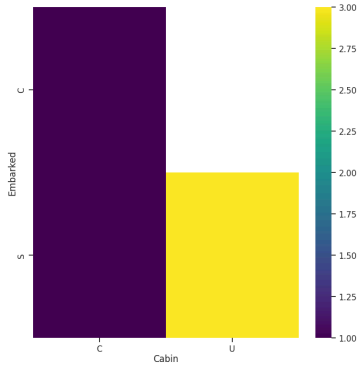


2-d categorical distributions



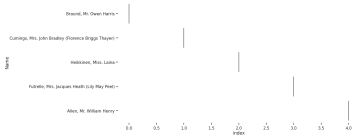




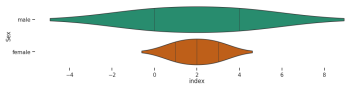


Faceted distributions

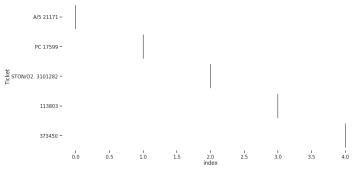
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.
```



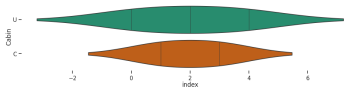
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.
```




```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.
```




```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.
```



Обработка пропусков в Embarked

Пропуски составляют около 0.22% данных.

In [ ]:
```
hdata['Embarked'] = hdata['Embarked'].fillna(hdata['Embarked'].mode()[0])
print(hdata['Embarked'].isnull().sum())
```

```
0
```

In [ ]:
```
[(c, hdata[c].isnull().sum()) for c in hcols_with_na]
```

Out[ ]:
```
[('Age', 0), ('Cabin', 0), ('Embarked', 0)]
```

Кодирование категориальных признаков

One-Hot Encoding (для признаков без порядка)

In [ ]:
```
cdata = hdata
cdata = pd.get_dummies(hdata, columns=['Embarked', 'Sex'])
```

Label Encoding (для признаков с порядком)

In [ ]:
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
cdata['Cabin'] = label_encoder.fit_transform(hdata['Cabin'])
```

In [ ]:
```
dict(zip(label_encoder.classes_, label_encoder.transform(label_encoder.classes
```

Out[ ]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'T': 7, 'U': 8}
```

In [ ]:
```
cdata.head()
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.100
4	5	0	3	Allen, Mr. William Henry	35.0	0	0	373450	8.050

Масштабирование данных.

In [ ]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
df = cdata
# Выбираем числовые признаки для масштабирования
num_features = ["Age", "Fare"]
min_max_scaler = MinMaxScaler()
df_minmax = df.copy()
df_minmax[num_features] = min_max_scaler.fit_transform(df[num_features])
```

```
standard_scaler = StandardScaler()
df_standard = df.copy()
df_standard[num_features] = standard_scaler.fit_transform(df[num_features])

fig, axes = plt.subplots(3, 2, figsize=(18, 12))
fig.suptitle("Распределение признаков до и после масштабирования", fontsize=16

for i, feature in enumerate(num_features):
    # Оригинальные данные
    sns.histplot(df[feature], bins=30, kde=True, ax=axes[0, i], color="blue")
    axes[0, i].set_title(f"Оригинал: {feature}")

    # Min-Max Scaling
    sns.histplot(df_minmax[feature], bins=30, kde=True, ax=axes[1, i], color="
    axes[1, i].set_title(f"Min-Max: {feature}")

    # Standard Scaling
    sns.histplot(df_standard[feature], bins=30, kde=True, ax=axes[2, i], color
    axes[2, i].set_title(f"Standard Scaler: {feature}")

plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```

Распределение признаков до и после масштабирования