

затребуваним на ринку праці. Університет створить розуміння того, що потрібно вивчати та надасть напрямок руху заклавши фундамент та умовний рельєф для перетворення студента, у справжнього фахівця, але знань, які ви там зможете отримати, очевидно не достатньо. Що стосується університетської практики, то там все неоднозначно.

Способи підвищення зацікавленості, для студента під час навчання в університеті

1. Місцеві студентські тусовки!
2. Знайдіть свого сенсея!
3. Влаштуйтеся на роботу в університеті або знайдіть підробіток!
4. Змініть відношення до університету!

Список літератури

1. https://brdo.com.ua/wp-content/uploads/2021/02/Analiz_IT_osvity_u_vyshah_Ukrai-ny_Print.pdf
2. <https://er.nau.edu.ua/handle/NAU/41153>
3. <https://dou.ua/lenta/articles/reform-in-it-education/>
4. Metanit. <https://metanit.com/>
5. code.mu. <http://code.mu/ru/javascript/book/prime/>

УДК 004.82:004:85

*Нескородєва А. Р., студентка 3 курсу
СО Бакалавр
Данильчук О. М., к. пед. наук, доцент,
кафедри прикладної математики*

ВИКОРИСТАННЯ БІБЛІОТЕКИ MATPLOTLIB ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Matplotlib — це бібліотека для створення статичних, анімованих та інтерактивних візуалізацій на Python[1]. Matplotlib робить легкі речі легкими, а важкі можливими. Бібліотека підтримує 2d та 3d графіку. Вона написана і підтримується в основному Джоном Хантером і поширюється на умовах BSD-подібної ліцензії. В документації автор зізнається, що Matplotlib починався з імітування графічних команд MATLAB, але є незалежним від нього проектом.

Інтерфейс ruIab дозволяє легко використовувати matplotlib досвідченими користувачами MATLAB. Нижче наведені деякі переваги використання matplotlib, як аналогу MATLAB:

- вбудована підтримка Scalable Vector Graphics (масштабована векторна графіка);
- є відкритим програмним забезпеченням;
- безкоштовний.

Роздивимось використання бібліотеки на прикладі набору даних «Титанік»[2]. В ньому записані відомості про пасажирів та вижили вони чи ні. Для початку необхідно імпортувати бібліотеку та зчитати дані з файлу:

```
import matplotlib.pyplot as plt
```

```
train = pd.read_csv('titanic_train_data.csv')
```

Далі використавши цей код побудуємо точкову діаграму яка відображає вижив пасажир чи ні в використовуючи значення віку та ціни квитка:

```
import matplotlib.patches as mpatches
```

```
plt.figure(figsize=(10,5)) # ініціалізуємо об'єкт
```

```
# позначимо пасажирів які врятувались
```

```
plt.scatter(train["Age"][train["Survived"] == 0], train["Fare"][train["Survived"] == 0], c="red", marker="x")
```

```
# позначимо пасажирів які не врятувались
```

```
plt.scatter(train["Age"][train["Survived"] == 1], train["Fare"][train["Survived"] == 1], c="blue", marker="o")
```

```
plt.grid()
```

```
blue_patch = mpatches.Patch(color='blue', label='survived')
```

```
red_patch = mpatches.Patch(color='red', label='died')
```

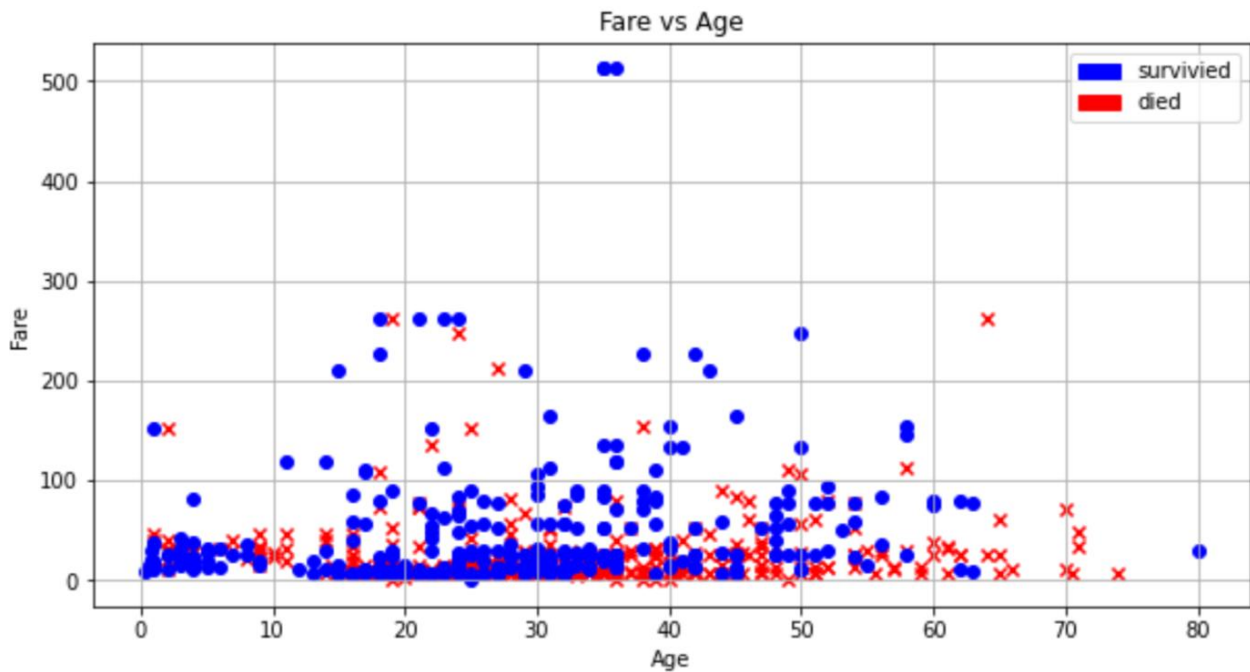
```
plt.legend(handles=[blue_patch, red_patch])
```

```
plt.title("Fare vs Age")
```

```
plt.xlabel("Age")
```

```
plt.ylabel("Fare")
```

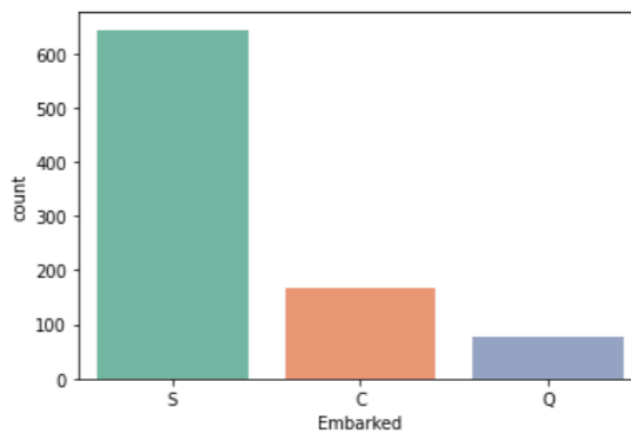
```
plt.show()
```



Діаграма 1.Кореляційне поле

Для того щоб подивитись на діаграму розподілу місця посадки пасажирів використаєм цю команду:

```
ax = sns.countplot(x='Embarked', data=train, palette='Set2')
plt.show()
```



Діаграма 2. Розподіл кількості пасажирів за містом(портом) посадки, де C = Cherbourg, Q = Queenstown, S = Southampton).

За допомогою цього коду побудуємо матрицю кореляції 8 ознак до ознаки 'survived' для цього також використаєм допоміжну бібліотеку seaborn[3]:

```
import seaborn as sns

correlation_matrix = train.iloc[:, 1:].corr()
```

```

top_features = correlation_matrix.nlargest(correlation_matrix.shape[0], 'Survived')['Survived'].index

fig, ax = plt.subplots(figsize=(12,12))

sns.heatmap(train[top_features].corr(), square=True, annot=True);

```

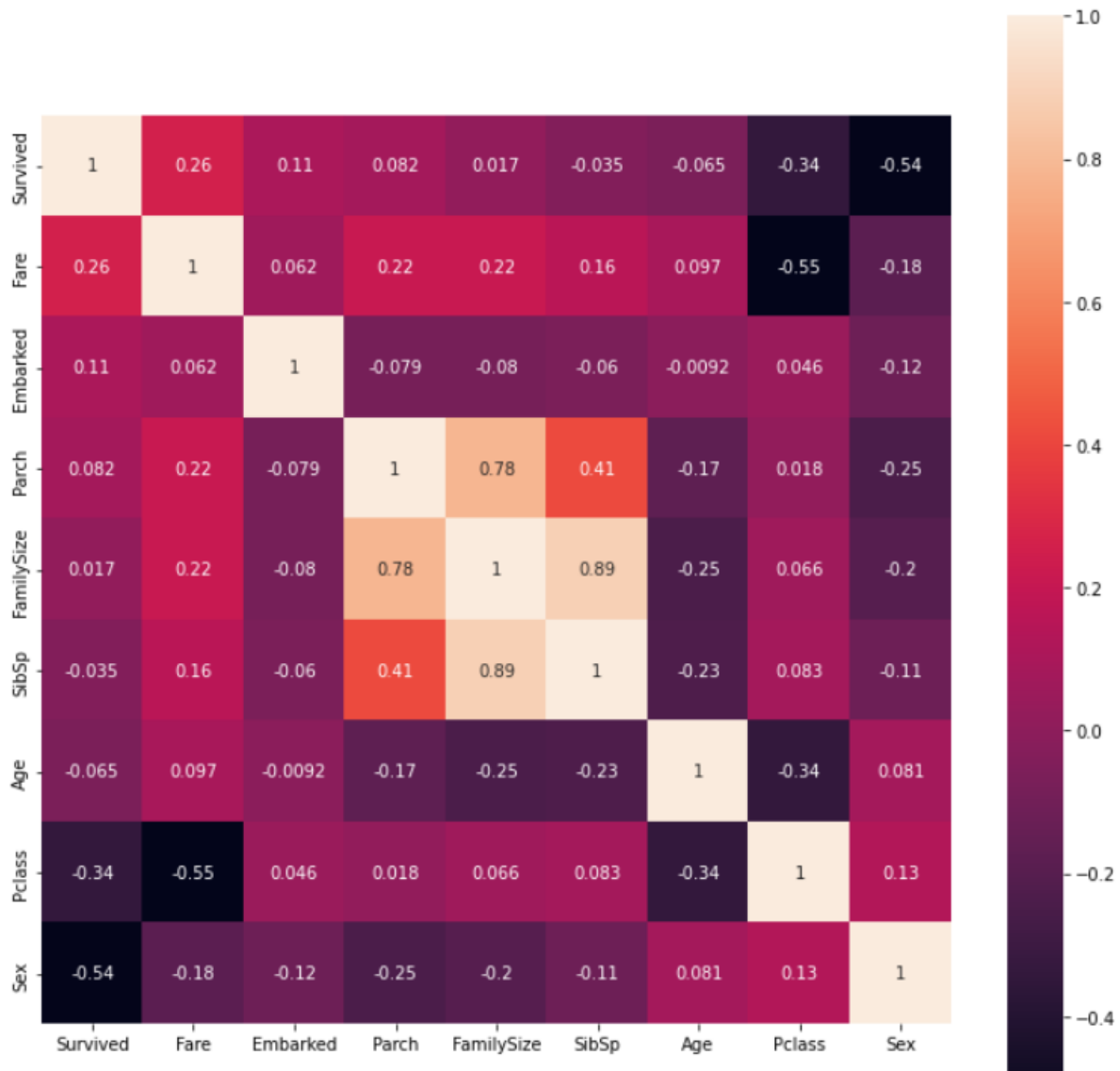


Рисунок 1. Кореляційна таблиця

Для того щоб більш детального подивитись на співвідношення ознак які більше за все корелюють з ознакою 'survived' використаєм цей код:

```

sns.set(style='ticks')

plots_in_row = 4

rows_number = 2

for i in range(1, rows_number + 1):
    index = (i-1) * plots_in_row + 1

```

```

sns.pairplot(data=train,

              y_vars='Survived',

              x_vars=top_features[index: index + plots_in_row],

              kind="reg",

              height = 3.2)

plt.show()

```

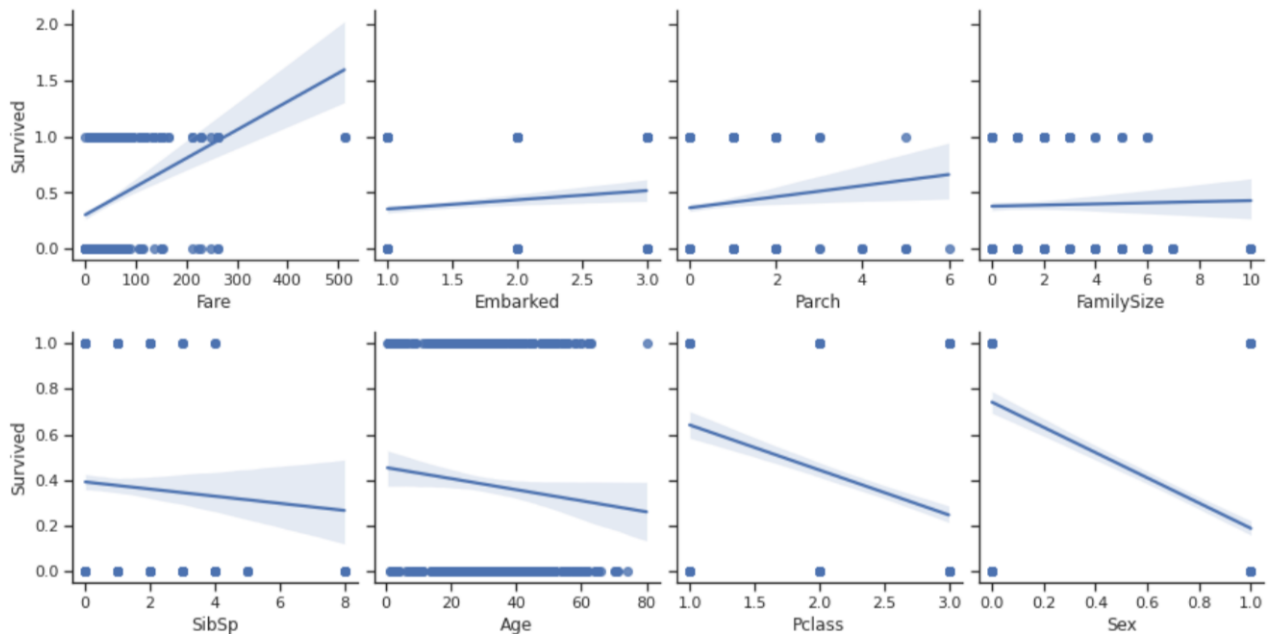


Рисунок 2. Графік співвідношень ознак з ознакою 'survived'

Висновок. За допомогою бібліотеки `matplotlib` на мові програмування `python` можна побудувати векторні графіки, діаграми та багато іншого. Бібліотека підтримується та оновлюється, а її використання є безкоштовним.

Список літературних джерел:

1. Сайт бібліотеки `Matplotlib` – [Електронний ресурс]. Режим доступу: <https://matplotlib.org/>
2. Сайт `Kaggle` – [Електронний ресурс]. Режим доступу: <https://www.kaggle.com/competitions/titanic/data>
3. Сайт бібліотеки `Seaborn` – [Електронний ресурс]. Режим доступу: <https://seaborn.pydata.org/>

УДК 004.942+656.052.1

*Nikolyuk P.K., professor of information
technology department of Vasil Stus
Donetsk National department
Olena-Maria Pryamukhina, student
Artem Perepelytsia, student*