ЧНУ

Аналіз та прогноз донорства крові

Виконала:
студентка 4 курсу, 444А групи
Мельничук Анна Геннадіївна
Керівник:
Кандидат фізико-математичних наук,
доцент кафедри комп’ютерних наук
Ковальчук М.Л.

Мета

- аналіз даних про донорів та їхні донації за певний період часу
- прогнозування донорської активності

Проблеми/Вирішення

Проблеми:

- нестабільність здач крові
- залежність від кількості нових донорів
- немає моніторингу за кількістю доступних донорів до здачі крові на місяць

Вирішення

- аналіз відомих даних
- використання прогнозування для можливості відслідковування здач крові

Стек використаних технологій

Процес розробки

 # 1

Аналіз даних та статистика

 # 2

Проектування та реалізація моделі

 # 3

Візуалізація результатів

 # 4

Подальші вдосконалення

Попередній аналіз даних

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)
0	2	50	12500	98
1	0	13	3250	28
2	1	16	4000	35
3	2	20	5000	45
4	1	24	6000	77

- R (Останній час - місяці з моменту останньої пожертви)
- F (Частота - загальна кількість пожертвувань)
- M (Кров – загальна кількість донорської крові в к.к.)
- T (Час - місяці з моменту першої донації)

Статистика

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)
count	748.000000	748.000000	748.000000	748.000000
mean	9.506684	5.514706	1378.676471	34.282086
std	8.095396	5.839307	1459.826781	24.376714
min	0.000000	1.000000	250.000000	2.000000
25%	2.750000	2.000000	500.000000	16.000000
50%	7.000000	4.000000	1000.000000	28.000000
75%	14.000000	7.000000	1750.000000	50.000000
max	74.000000	50.000000	12500.000000	98.000000

Нормалізація даних

До

```
Recency (months)              66.929
Frequency (times)             33.830
Monetary (c.c. blood)    2114363.700
Time (months)                611.147
dtype: float64
```

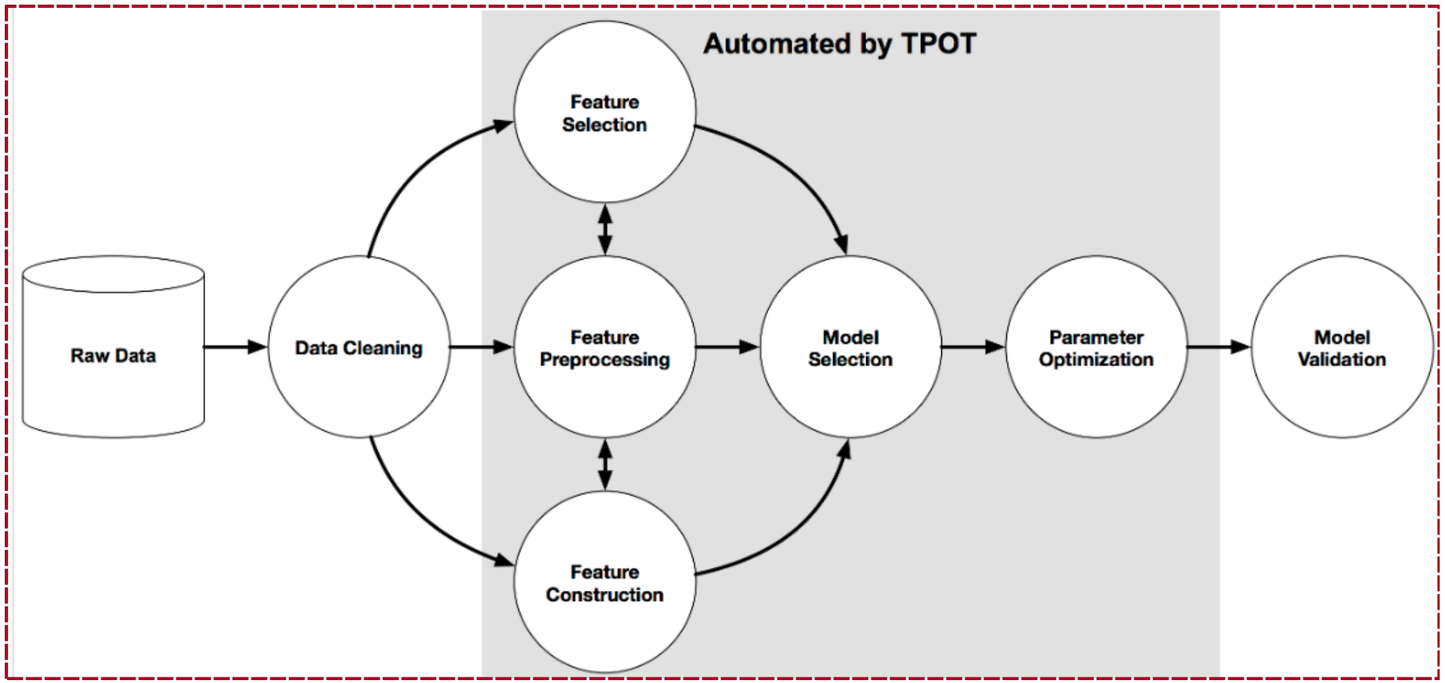
Після

```
Recency (months)      66.929
Frequency (times)     33.830
Time (months)        611.147
monetary_log           0.837
dtype: float64
```

Поділ на тестову та навчальну вибірки

```python
# Import train_test_split method
from sklearn.model_selection import train_test_split

# Розділити переливання DataFrame на
# Набори даних X_train, X_test, y_train і y_test,
# розшарування на стовпець `target`
X_train, X_test, y_train, y_test = train_test_split(
    transfusion.drop(columns='target'),
    transfusion.target,
    test_size=0.25,
    random_state=42,
    stratify=transfusion.target
)

# Роздрукуйте перші 2 рядки X_train
X_train.head(2)
```

Застосування TPOT



TPOT — це інструмент автоматизованого машинного навчання Python.
Ми використовуємо його, щоб допомогти нам зосередитися на одній моделі, яку потім можемо досліджувати та оптимізувати далі.

```
Generation 1 - Current best internal CV score: 0.7422459184429089

Generation 2 - Current best internal CV score: 0.7422459184429089

Generation 3 - Current best internal CV score: 0.7422459184429089

Generation 4 - Current best internal CV score: 0.7422459184429089

Generation 5 - Current best internal CV score: 0.7423330644124078

Best pipeline: LogisticRegression(RobustScaler(input_matrix), C=25.0, dual=False, penalty=l2)

AUC score: 0.7858

Best pipeline steps:
1. RobustScaler()
2. LogisticRegression(C=25.0, random_state=42)
```

Вибір найкращої моделі для нашого набору даних

```python
# Importing modules
from sklearn import linear_model
from sklearn.metrics import roc_auc_score

# Створення екземпляра логістичної регресії
logreg = linear_model.LogisticRegression(
    solver='liblinear',
    random_state=42
)

# Тренування моделі
logreg.fit(X_train_normed, y_train)

# Оцінка AUC для моделі tpot
logreg_auc_score = roc_auc_score(y_test, logreg.predict_proba(X_test_normed)[:, 1])
print(f'\nAUC score: {logreg_auc_score:.4f}')
```

```
AUC score: 0.7891
```

Навчання моделі логістичної регресії на підготовлених даних та оцінка її якості

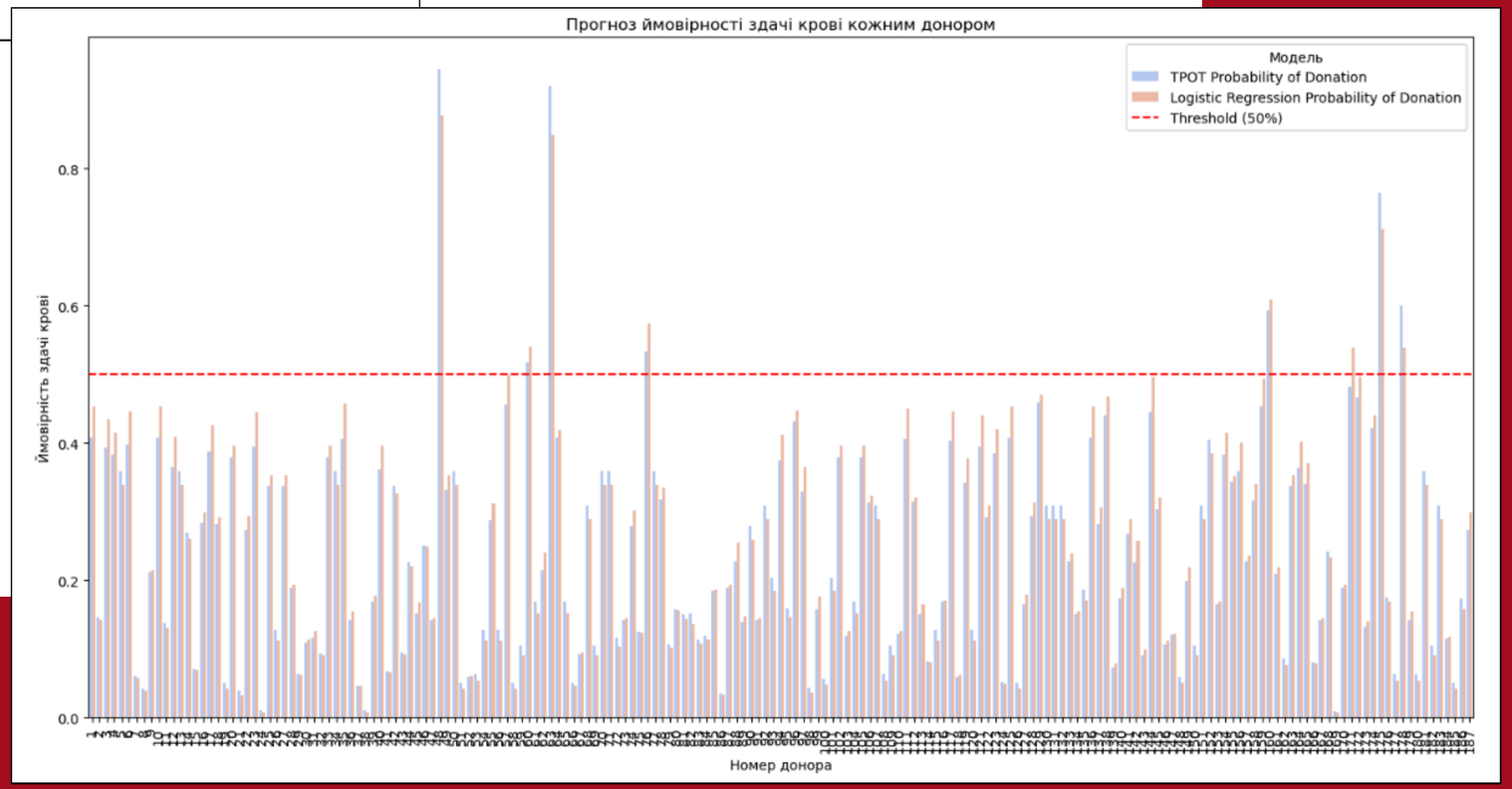
Після навчання модель оцінює ймовірність того, що конкретний донор прийде знову

```python
# Прогноз для кожного донора в тестовому наборі за допомогою логістичної регресії
logreg_predictions = logreg.predict_proba(X_test_normed)[:, 1]

# Створення таблиці з номерами донорів і їх прогнозами
predictions_df = pd.DataFrame({
    'Donor Number': range(1, len(X_test) + 1),
    'Logistic Regression Probability of Donation': logreg_predictions
})

# Виведення таблиці
print(predictions_df)
```

Результат

	Donor Number	Logistic Regression Probability of Donation
0	1	0.452876
1	2	0.142197
2	3	0.434499
3	4	0.414502
4	5	0.339023
..	...	...
182	183	0.288582
183	184	0.117598
184	185	0.041835
185	186	0.157384
186	187	0.299046

Загальний прогноз (Logistic Regression): 24.58% донорів



Висновки

Отже, у ході роботи було проаналізовано дані про донорів та їхні донації за певний період часу, що дозволяє нам побачити загальну картину співпраці донорів з мед закладом.

Завдяки аналізу історичних даних вдалося створити модель, яка прогнозує активність донорів, допомагаючи уникати дефіциту крові в критичних ситуаціях.

Дана модель може бути інтегрована в існуючі інформаційні системи медичних закладів

Можливості подальшого розвитку

Для подальшого розвитку системи планується реалізація наступних покращень:

- Під’єднати цей аналіз до вже існуючої версії додатку
- Навчання моделі для підрахунку потреби в кожній групі крові окремо, що дозволить забезпечити ефективне управління запасами для кожної групи.
- Інтеграція з іншими медичними системами для автоматизованого отримання оновлених даних та підвищення швидкості реагування на потреби у крові.

Дякую за увагу

Основні цілі :

Mission

- Побудувати модель для реалізації прогнозування
- Забезпечити точність прогнозування потреб у крові для ефективного управління запасами