МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий
Кафедра прикладной математики

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 5

«Классификация»

по дисциплине

«Технологии и инструментарий анализа больших данных»

Выполнил студент группы ИВБО-03-21 — Лазарев А. В.

Принял преподаватель кафедры прикладной математики — Тетерин Н.Н.

Практическая работа выполнена « __ » _____ 2024 г.

«Зачтено» « __ » _____ 2024 г.

Москва 2024

СОДЕРЖАНИЕ

1 РЕШЕНИЕ ЗАДАЧ

1.1 Задача №1

Решение программы представлено на Рисунке 1.1.

```
[ ]  url = 'https://raw.githubusercontent.com/InspectorJelly/BigDataMirea/refs/heads/main/datasets/insurance.csv'
     data = pd.read_csv(url)
```

Рисунок 1.1 – Программа

1.2 Задача №2

Решение и результат программы представлены на Рисунке 1.2, 1.3.

```
column_to_plot = 'region'
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x=column_to_plot, palette="viridis")
plt.title(f"Распределение по классу '{column_to_plot}'")
plt.xlabel(column_to_plot.capitalize())
plt.ylabel("Частота")
plt.show()
```

Рисунок 1.2 – Программа

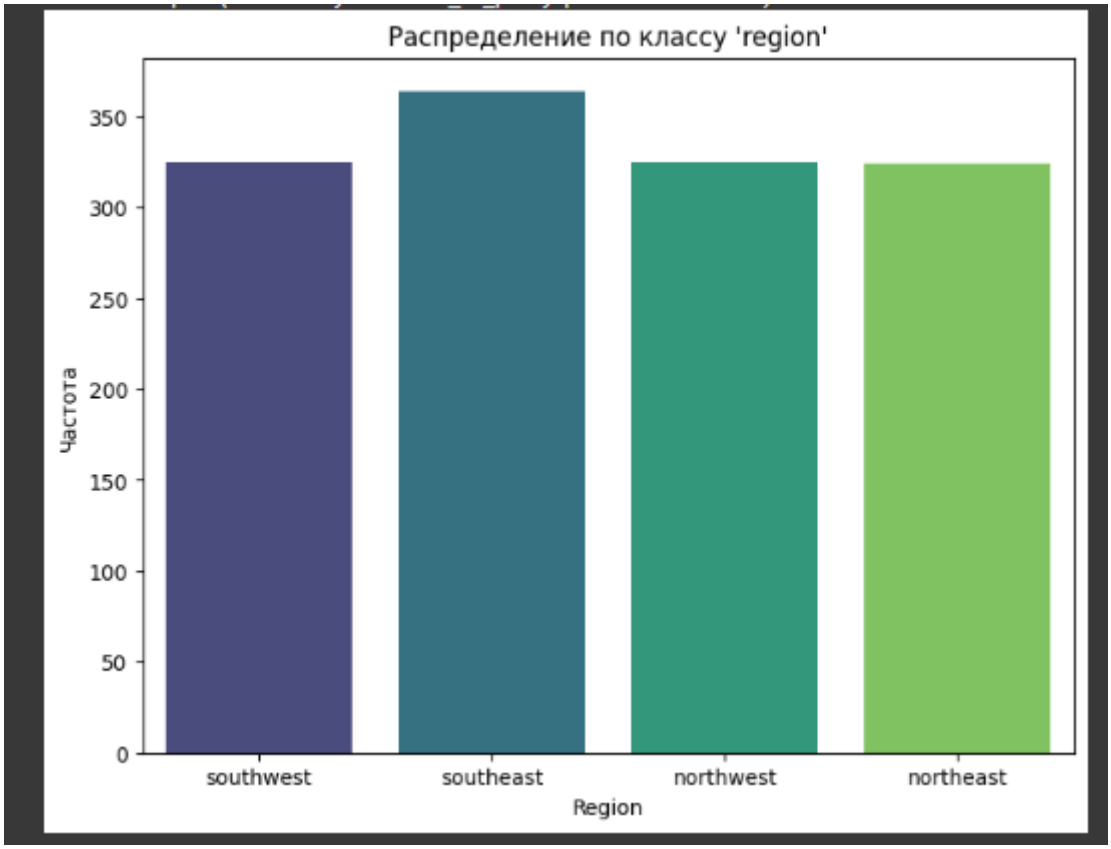

Рисунок 1.3 – Результат выполнения программы

1.3 Задача №3

Решение и результат программы представлены на Рисунке 1.4.

```
[ ]  X = data.drop(columns='charges')
     y = data['charges']

     #(80% - train, 20% - test)
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     print(f"Размер тренировочной выборки: {X_train.shape}, {y_train.shape}")
     print(f"Размер тестовой выборки: {X_test.shape}, {y_test.shape}")

     Размер тренировочной выборки: (1070, 6), (1070,)
     Размер тестовой выборки: (268, 6), (268,)
```

Рисунок 1.4 – Программа и результат ее выполнения

1.4 Задача №4

Решение и результат программы представлены на Рисунках 1.5, 1.6.

```
X = data.drop(columns='sex')
X = pd.get_dummies(X, drop_first=True)
y = data['sex'].apply(lambda x: 1 if x == 'male' else 0)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

models = {
    "Logistic Regression": LogisticRegression(max_iter=200),
    "SVM": SVC(),
    "KNN": KNeighborsClassifier(n_neighbors=5)
}

plt.figure(figsize=(15, 5))
for i, (model_name, model) in enumerate(models.items(), 1):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    cm = confusion_matrix(y_test, y_pred)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Female', 'Male'])
    plt.subplot(1, 3, i)
    disp.plot(cmap='Blues', ax=plt.gca(), values_format='d')
    plt.title(f"{model_name} - Confusion Matrix")

plt.tight_layout()
plt.show()
```

Рисунок 1.5 – Программа

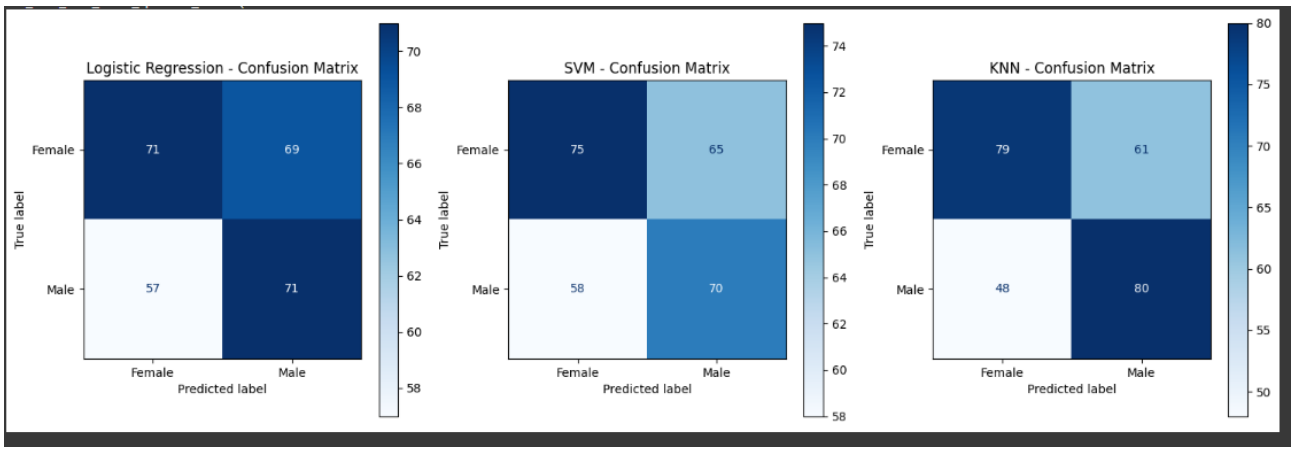

Рисунок 1.6 – Результат выполнения программы

1.5 Задача №5

Решение и результат программы представлены на Рисунке 1.7.

```
for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(f"Метрики для {model_name}:")
    print(classification_report(y_test, y_pred, target_names=['Female', 'Male']))
    print("="*50)
```

Рисунок 1.7 – Программа

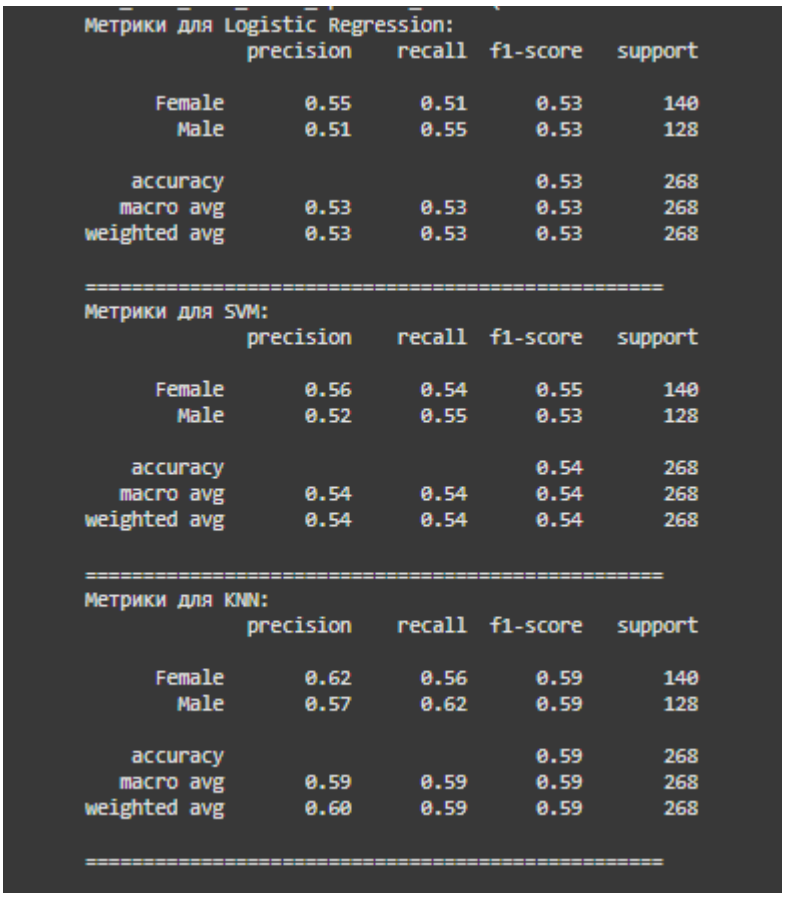

Рисунок 1.8 – Результат выполнения программы