Министерство образования Республики Беларусь

Учреждение Образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

Лабораторная работа №5

Статистические основы индуктивного вывода

Проверил:

Выполнили: Гаевой Александр(*121731*)

Минск

2023 г.

Содержание:

1. Цель ……………………………………………………………………………………3
2. Код программы………………………………………………………………………..4
3. Вывод программы……………………………………………………………………..7

Код программы

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.tree import DecisionTreeClassifier, plot\_tree, export\_text

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report

# Загружаем датасет

df = pd.read\_csv("dts.csv")

# Разделяем датасет на признаки и целевую переменную

X = df.drop('DEATH\_EVENT', axis=1)

y = df['DEATH\_EVENT']

# Разделяем датасет на обучающую и тестовую выборки

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.1, random\_state=42)

# Создаем и обучаем модель дерева решений

dt = DecisionTreeClassifier(max\_depth=3, random\_state=42)

dt.fit(X\_train, y\_train)

# Предсказываем результаты на тестовой выборке

y\_pred = dt.predict(X\_test)

# Оцениваем точность модели построив матрицу сходства

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()

# Вычисляем метрики точности, полноты, F-меры и AUC

print('Accuracy:', accuracy\_score(y\_test, y\_pred))

print('Classification Report:\n', classification\_report(y\_test, y\_pred))

# Получаем интерпретацию построенной модели

# Визуализируем дерево решений

plt.figure(figsize=(12,8))

plot\_tree(dt, feature\_names=X.columns, class\_names=['Survived', 'Died'], filled=True, rounded=True)

plt.show()

# Выводим текстовое представление дерева решений

print('Decision Tree:\n', export\_text(dt, feature\_names=X.columns, show\_weights=True))

# Указываем какие знания можно получить из набора

# Анализируем важность признаков для модели

feature\_importances = pd.Series(dt.feature\_importances\_, index=X.columns).sort\_values(ascending=False)

sns.barplot(x=feature\_importances, y=feature\_importances.index)

plt.xlabel('Feature Importance Score')

plt.ylabel('Features')

plt.title('Important Features')

plt.show()

# Выводим корреляционную матрицу признаков

corr = df.corr()

sns.heatmap(corr, annot=True, cmap='RdBu')

plt.title('Correlation Matrix')

plt.show()







