Тулегулов Амандос Добысович
канд. физ.-мат. наук, профессор
Ешпанов Владимир Сарсембаевич
д-р ист. наук, профессор
Исмаилов Асылхан
магистрант
Серикпай Айнур Талгаткызы
магистр, преподаватель
Абдикеримова Айнур Абдикадировна
магистр, преподаватель
Казахский университет технологии и бизнеса
г. Нур-Султан, Республика Казахстан
Сарсембай Милена Владимировна
учитель
Школа-гимназия №22
г. Нур-Султан, Республика Казахстан

ПРАКТИЧЕСКИЙ ОПЫТ ОБУЧЕНИЯ МЕТОДАМ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА НА ПЛАТФОРМЕ PYTHON ANACONDA

Аннотация*: в статье представлены краткие результаты практического опыта обучения методам интеллектуального анализа на платформе Pyton Anaconda. Приведены конкретные скрипты, на основе которых сделаны выводы, которые позволяют утверждать об эффективности применения методов интеллектуального анализа на практических занятиях по дисциплине «Информатика». В результате авторы получили достаточно полную модель, на основе которой возможно дальнейшее прогнозирование поведения объектов исследования в различных ситуациях, что в свою очередь доказывает, что в процессе формирования понимания значимости развития цифровых навыков огромную роль играют практики, демонстрирующие конкретные результаты.*

***Ключевые слова**: Pyton Anaconda, скрипты, интеллектуальный анализ, модель, цифровые навыки.*

Помимо теоретического обучения важную роль в успешном освоении дисциплины «Информатика» играют практические навыки. В данной статье мы хотим поделиться опытом практического применения методов интеллектуального анализа на открытой платформе Anaconda, которая является дистрибутивом языков программирования Python и R, включает в себя набор популярных свободных библиотек, объединенных проблематиками науки о данных и машинного обучения.

Платформа Anaconda Navigator является GUI, включенный в дистрибутив Anaconda, который позволяет запускать приложения и легко управлять пакетами, средами и каналами conda без использования команд командной строки.

В нашей практике использовались следующие библиотеки и модули ввода, обработки и визуализации данных платформы Anaconda, как:

– Pandas – которая предназначена для предварительной обработки и анализа данных. Работа pandas с данными строится на основе библиотеки NumPy, являющейся инструментом более низкого уровня. Модуль предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами;

– Numpy – имеет возможности поддержки многомерных массивов и высокоуровневых математических функций, предназначенных для работы с многомерными массивами;

– Matplotlib – библиотека на Python для визуализации данных двумерной и трехмерной графикой. Получаемые изображения могут быть использованы в качестве иллюстраций в программном коде;

– Scikit-learn – библиотека для Python. Включает различные алгоритмы классификации, регрессии и кластеризации, включая SVM, случайные леса, усиление градиента, k -средства и DBSCAN, предназначен для взаимодействия с числовыми и научными библиотеками NumPy и SciPy;

– Lime – модуль объяснения прогнозов произвольно выбранного классификатора машинного обучения;

– Seaborn – модуль предназначен для визуализации статистических данных [1, с. 115].

Проект применения машинного обучения направлен на решение полной проблемы машинного обучения с использованием реального набора данных.

Цель данной статьи заключается в том, чтобы показать как на практике можно эффективно и доступно использовать преимущества методов интеллектуального анализа для проведения экспериментально-исследовательской работы:

– использование имеющихся большие данные для построения модели прогнозирования значения энергоэффективности (количество баллов Energy Star) для отдельно взятого здания в городе;

– преобразовать результаты для поиска факторов, влияющих на итоговый балл [2].

В открытых данных имеются здания, которым уже присвоены указанные баллы, и в дальнейшей работе будем реализовать регрессионное машинное обучение с учетом следующих условий:

– ставим задачу обучения модели, которая сможет самостоятельно сопоставить признаки и цель;

– баллы энергоэффективности являются непрерывным переменным.

Строящаяся модель должна соответствовать критериям:

а) точности – она должна спрогнозировать значение баллов Energy Star максимально близко к реальному;

б) интерпретируемости – прогнозы модели должны быть понятными. Зная целевые данные, можно использовать их по мере детального изучения данных и создания модели [3, с. 201].

Для достижения цели в экспериментально-исследовательской работе последовательно реализуем «этапы машинного анализа данных:

– очистка и форматирование данных;

– разведочный анализ данных;

– конструирование и выбор признаков;

– сравнение метрик нескольких моделей машинного обучения;

– гиперпараметрическая настройка лучшей модели;

– оценка лучшей модели на тестовой выборке данных;

– интерпретирование результатов работы модели;

– выводы по применению инструментов и отчет о результатах» [4].

После запуска необходимых модулей ввода и преобразования исходных данных проводим их очистку и форматирование [5].

```
# Загрузить модули Pandas и Numpy для преобразования данных
import pandas as pd
import numpy as np

# Не предупреждать о значении настройки на копии среза данных
pd.options.mode.chained_assignment = None

# Отобразить до 60 столбцов данных
pd.set_option('display.max_columns', 60)

# Загрузить модуль визуализации Matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

```
# Установить размер шрифта по умолчанию
plt.rcParams['font.size'] = 24

# Внутренний инструмент ipython для настройки размера фигур
from IPython.core.pylabtools import figsize

# Загрузить модуль визуализации статистики Seaborn
import seaborn as sns
sns.set(font_scale = 2)

# Разделить данные на тренировочную и тестовую
from sklearn.model_selection import train_test_split
```

Затем даем команду загрузки данных в датафрейм и проверим загруженные данных.

```
# Загрузить данные в датафрейм
data = pd.read_csv('Energy_and_Water_Data.csv')

# Показать верхнюю строку датафрейма
data.head()
```

Набор данных представляет собой множество наблюдений, однако в нем присутствуют аномалии и пропущенные значения, которые очистим и приведем к нужному формату.

Просматриваем типы данных и пропущенные значения в датафрейме.

```
# Просмотреть столбец типов данных и непропущенных значений
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11746 entries, 0 to 11745
Data columns (total 60 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   Order                                           11746 non-null  int64
 1   Property Id                                     11746 non-null  int64
 2   Property Name                                   11746 non-null  object
 3   Parent Property Id                              11746 non-null  object
 4   Parent Property Name                            11746 non-null  object
 5   BBL - 10 digits                                 11695 non-null  float64
 6   NYC Borough, Block and Lot (BBL) self-reported  11746 non-null  object
 7   NYC Building Identification Number (BIN)        11746 non-null  object
 8   Address 1 (self-reported)                       11746 non-null  object
 9   Address 2                                       11746 non-null  object
 10  Postal Code                                     11746 non-null  object
 11  Street Number                                   9560 non-null   float64
 12  Street Name                                     11624 non-null  object
 13  Borough                                         11628 non-null  object
 14  DOF Gross Floor Area                            11628 non-null  float64
```

Таким образом, в результате проведенных операций, мы можем получить достаточно полную модель, на основе которой возможно дальнейшее прогнозирование поведения объектов исследования в различных ситуациях.

В заключение хотелось бы отметить, что в процессе формирования понимания значимости развития цифровых навыков огромную роль играют практики, демонстрирующие конкретные результаты, что в свою очередь повышают заинтересованность и активизируют процессы обучения.

Список литературы

1. Барсегян А.А. Методы и модели анализа данных: OLAP и Data Mining / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко [и др.]. – СПб.: БХВ-Петербург, 2004. – 336 с.

2. Делаем проект по машинному обучению на Python. Ч. 1 / NIX Solutions corporate blog / Habr [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/nix/blog/425253/

3. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. – СПб.: Питер, 2018. – 576 с.

4. A Complete Machine Learning Walk-Through in Python: Part Three [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/

5. Witten I.H. Data Mining: practical machine learning tools and techniques / I.H. Witten, Eibe Frank. – 2nd ed. p. cm. – (Morgan Kaufmann series in data management systems).