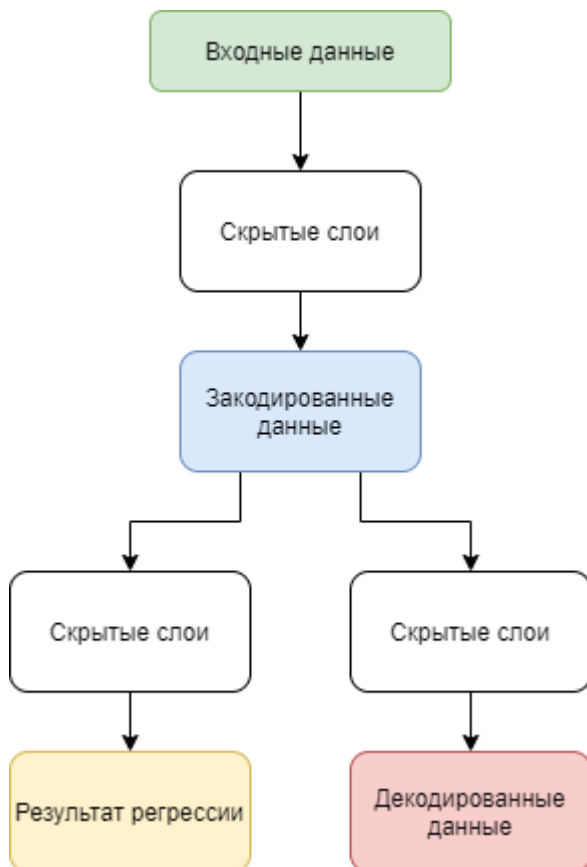


Задание

Необходимо в зависимости от варианта сгенерировать датасет и сохранить его в формате csv.

Построить модель, которая будет содержать в себе [автокодировщик](#) и регрессионную модель. Схематично это должно выглядеть следующим образом:



Обучить модель и разбить обученную модель на 3: Модель кодирования данных (Входные данные -> Закодированные данные), модель декодирования данных (Закодированные данные -> Декодированные данные), и регрессионную модель (Входные данные -> Результат регрессии).

В качестве результата представить исходный код, сгенерированные данные в формате csv, кодированные и декодированные данные в формате csv, результат регрессии в формате csv (что должно быть и что выдает модель), и сами 3 модели в формате h5.

Выполнение

Был реализован метод getData:

```
def getData(nrow):
    data = np.zeros((nrow, 6))
    targets = np.zeros(nrow)
    for i in range(nrow):
        x = np.random.normal(-5, 10)
        e = np.random.normal(0, 0.3)
        data[i, :] = (
            (-x) ** 3 + e, np.log(np.fabs(x)) + e, np.sin(3 * x) + e, x + 4 + e, -x + np.sqrt(np.fabs(x)) + e,
            x + e)
        targets[i] = np.exp(x) + e
    return data, targets
```

Реализована функция инициализации структуры НС:

```
def create():
    main_input = Input(shape=(6,), name='mainInput')

    enc = Dense(64, activation='relu')(main_input)
    enc = Dense(32, activation='relu')(enc)
    enc = Dense(6, activation='relu', name="encode")(enc)

    input2 = Input(shape=(6,), name='input_encoded')

    dec = Dense(32, activation='relu')(input2)
    dec = Dense(64, activation='relu')(dec)
    dec = Dense(6, name='decode')(dec)

    pred = Dense(64, activation='relu')(enc)
    pred = Dense(32, activation='relu')(pred)
    pred = Dense(16, activation='relu')(pred)
    pred = Dense(1, name="predict")(pred)

    enc = Model(main_input, enc, name="encoder")
    dec = Model(input2, dec, name="decoder")
    pred = Model(main_input, pred, name="autoencoder")
    return enc, dec, pred, main_input
```

Функция записи в csv:

```
def write_csv(path, data):
    with open(path, 'w', newline='') as file:
        my_csv = csv.writer(file, delimiter=',', quoting=csv.QUOTE_MINIMAL)
        try:
            for i in data:
                my_csv.writerow(i)
        except Exception as ex:
            my_csv.writerow(data)
```