Министерство науки и высшего образования Российской Федерации

Федеральное государственное образовательное учреждение

высшего образования

«Волгоградский государственный технический университет»

Факультет «Электроники и вычислительной техники»

Кафедра «Системы автоматизированного проектирования и поискового конструирования»

Контрольная работа

По дисциплине «Технологии анализа данных»

На тему: «Анализ данных с использованием метода линейной регрессии»

Выполнил:

Студент группы ЭВМ 1.2,

Галоян А. М.

Проверила:

Профессор кафедры САПР,

Садовникова Н. П.

Волгоград, 2023

Оглавление

1. Постановка задачи

В качестве набора данных для решения задачи регрессии был выбран датасет Spotify Song Popularity, который содержит информацию о 41 099 уникальных песнях, найденных на популярном музыкальном стриминговом сервисе Spotify. Данные, описывающие эти песни, были получены из API Spotify и объединены с данными из API Billboard. Все песни в наборе данных были выпущены в период с 1960-х по 2010-е годы. Spotify алгоритмически генерирует рейтинги для таких характеристик трека, как темп, акустичность, валентность и т. д. Данный датасет имеет следующие поля:

- *song_name*: название песни;
- *song_popularity*: популярность песни от 0 до 100;
- *song_duration_ms*: длительность песни в мс;
- *acousticness*: мера достоверности от 0,0 до 1,0 того, является ли дорожка акустической;
- *danceability*: танцевальность описывает, насколько трек подходит для танцев на основе комбинации музыкальных элементов, включая темп, стабильность ритма, силу бита и общую регулярность;
- *energy*: мера энергичности от 0,0 до 1,0 описывает интенсивность и активность;
- *instrumentalness:* предсказывает, содержит ли дорожка вокал;
- *key:* тональность от 0 до 11;
- *liveness:* определяет присутствие аудитории в записи;
- *loudness:* общая громкость трека в децибелах*;*

Источником, откуда был взят данный набор данных, является репозиторий Kaggle

В ходе исследования необходимо проанализировать зависимость между характеристиками песен и их популярностью. Результатом

исследования должна стать модель, позволяющая на основании характеристик песен предсказать, какие факторы способствуют популярности.

2. Описание используемого метода

Линейная регрессия представляет собой метод анализа данных, который исследует взаимосвязь между двумя переменными, где одна из них выступает в роли зависимой, а другая — независимой. Этот метод помогает выявить математическую модель, наилучшим образом отражающую данную связь.

Применение линейной регрессии весьма разнообразно. Ее можно использовать для прогнозирования будущих значений переменных, выявления взаимосвязей между различными факторами, оценки воздействия этих факторов на результаты и многих других областей. Например, в бизнесе линейная регрессия может помочь определить, как изменение цены на продукт влияет на его продажи, а в медицине — выявить факторы, влияющие на здоровье человека.

Основная задача регрессии заключается в том, чтобы на основе доступных данных предсказывать значения числовой переменной. Например, по собранным данным о цене на недвижимость и характеристикам домов можно построить модель, которая предсказывает цену на дом в зависимости от этих характеристик.

Решение задачи регрессии включает различные методы, включая линейную регрессию, которая предполагает линейную зависимость между переменными и строит прямую линию, наилучшим образом соответствующую данным. Для оценки качества модели используются различные метрики, в том числе коэффициент детерминации (R-квадрат), который измеряет, насколько хорошо модель объясняет изменчивость данных.

Задачи регрессии широко применяются в разных областях, таких как экономика (прогнозирование цен), медицина (предсказание заболеваний) и многие другие.

3. Практическая часть

3.1 Инструменты анализа

Использование Python и Jupyter Notebook предоставляет удобный подход для создания прототипов и проведения экспериментов с кодом. В Jupyter Notebook вы можете непосредственно писать код и мгновенно видеть результаты его выполнения, что делает его идеальным для изучения новых концепций и проведения экспериментов. Кроме того, Jupyter Notebook облегчает обмен идеями с другими людьми, что делает его отличным инструментом для совместной работы.

Для проведения анализа используется язык программирования Python, а также интерактивный блокнот Jupyter.

В качестве используемых дополнительных библиотек представлены: «pandas», «matplotlib» и «sklearn».

- pandas — это библиотека Python для работы с данными, которая позволяет легко и быстро обрабатывать, и анализировать большие объемы данных.
- matplotlib — это библиотека Python для создания графиков и визуализации данных. Она позволяет создавать различные типы графиков, включая линейные, круговые, гистограммы и т.д.
- sklearn.model_selection — это модуль библиотеки scikit-learn, который предоставляет набор инструментов для выбора наилучших моделей машинного обучения. Он включает в себя функции для кросс-валидации, выбора размера выборки и других параметров модели.
- sklearn.linear_model — это модуль библиотеки scikit-learn, который предоставляет различные алгоритмы линейной регрессии для решения задач классификации и регрессии.
- sklearn.metrics.mean_squared_error — это функция из библиотеки scikit-learn, которая вычисляет среднеквадратичную ошибку

между предсказанными и фактическими значениями целевой переменной. Она часто используется для оценки качества моделей машинного обучения.

3.2 Процесс решения поставленной задачи

Загрузим данные и посмотрим первые данные из набора

```python
data = pd.read_csv("song_data.csv")
data.head()
```

Создаем отдельную целевую переменную popularity(популярность) и матрицу функции x, убирав первый столбец, отвечающий за название песни, так как при регрессионном анализе данный параметр нам необходим не будет

```python
# Целевая переменная
y = data['song_popularity']
# Матрица функции x
x = data.drop(columns=['song_name','song_popularity'])
```

Перед проведение исследования разделим набор данных на обучающую и тестовую выборку.

```python
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=50)
```

```

Предположим, что популярность песен однозначно определяется только одним из параметров, имеющих численное значение: темпа ударов в минуту или значением позитивности. Проверим два этих предположения.

Предположение 1. Популярность песен зависит только от ее темпа ударов в минуту .

Построим корреляционное поле (рис.1) и попробуем визуально определить вид зависимости.

```python
plt.plot(y_train, x_train['danceability'], 'o', markersize=1)
plt.ylabel('danceability')
plt.xlabel('song_popularity')
plt.show()
```

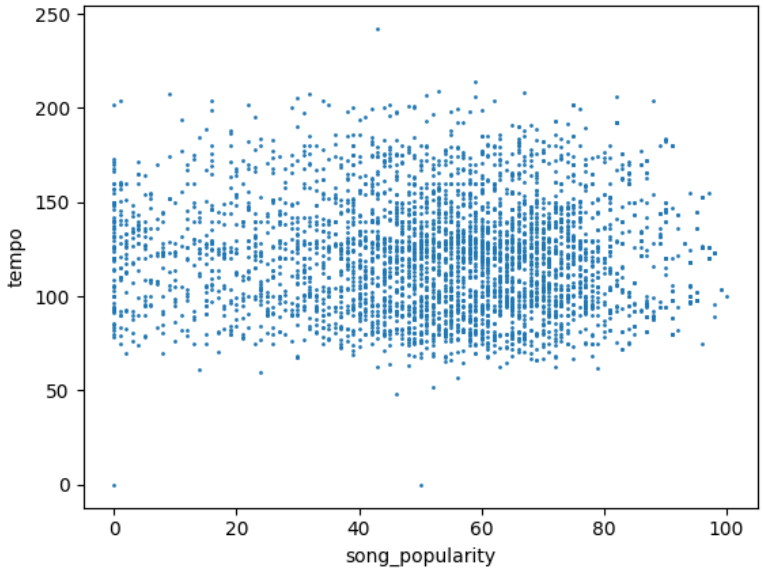


Рисунок 1 – Диаграмма зависимости популярности песни от уровня темпа

Как мы можем судить по полученной диаграмме, зависимость между темпа ударов в минуту и популярностью конфеты не прослеживается; предположение отвергнуто.
```

Предположение 2. Популярность песни зависит только от значения позитивности в ней.

Построим корреляционное поле (рис.2) и попробуем визуально определить вид зависимости.

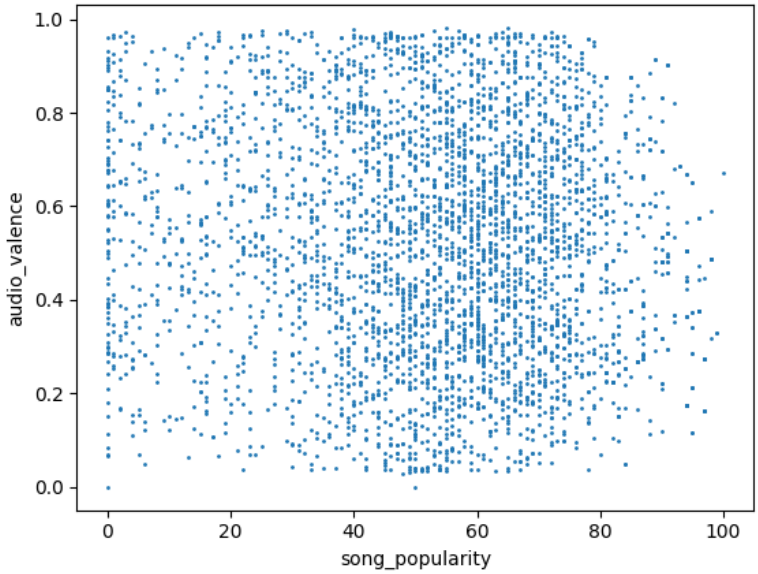


Рисунок 2 – Диаграмма зависимости популярности песни от значения позитивности в ней

Так же, как и в предыдущем предположении, не обнаружено связи между уровнем позитивности и популярностью песни. Предложенная гипотеза отклонена.

Теперь допустим, что влияние популярности песен зависит от нескольких параметров в совокупности. В начале определим, не существует ли взаимосвязи между этими параметрами, то есть исключим наличие мультиколлинеарности, прежде чем приступить к созданию регрессионной модели.

```python
corr = x_train.corr()
corr.style.background_gradient(cmap='coolwarm')
```

```

| | song_duration_ms | acousticness | danceability | energy | instrumentalness | key | liveness | loudness | audio_mode | speechiness | tempo | time_signature | audio_valence |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| song_duration_ms | 1.000000 | -0.094874 | -0.101165 | 0.093081 | -0.012531 | -0.001998 | 0.020380 | 0.017492 | -0.024520 | -0.082298 | 0.015132 | -0.000684 | -0.056156 |
| acousticness | -0.094874 | 1.000000 | -0.185686 | -0.661887 | 0.173816 | 0.001348 | -0.085496 | -0.556643 | 0.058522 | -0.083789 | -0.129319 | -0.158070 | -0.124809 |
| danceability | -0.101165 | -0.185686 | 1.000000 | 0.040581 | -0.136227 | 0.004437 | -0.090096 | 0.177824 | -0.108964 | 0.212231 | -0.123852 | 0.128823 | 0.329055 |
| energy | 0.093081 | -0.661887 | 0.040581 | 1.000000 | -0.209052 | 0.016636 | 0.173073 | 0.755517 | -0.045737 | 0.053230 | 0.157895 | 0.141264 | 0.316472 |
| instrumentalness | -0.012531 | 0.173816 | -0.136227 | -0.209052 | 1.000000 | -0.014001 | -0.029773 | -0.390838 | -0.004569 | -0.079957 | -0.038975 | -0.069474 | -0.182732 |
| key | -0.001998 | 0.001348 | 0.004437 | 0.016636 | -0.014001 | 1.000000 | -0.010985 | 0.008841 | -0.175796 | 0.031030 | 0.001768 | -0.003766 | 0.018772 |
| liveness | 0.020380 | -0.085496 | -0.090096 | 0.173073 | -0.029773 | -0.010985 | 1.000000 | 0.105524 | -0.000131 | 0.094426 | 0.033875 | 0.008342 | 0.013258 |
| loudness | 0.017492 | -0.556643 | 0.177824 | 0.755517 | -0.390838 | 0.008841 | 0.105524 | 1.000000 | -0.058171 | 0.069719 | 0.129403 | 0.114181 | 0.199856 |
| audio_mode | -0.024520 | 0.058522 | -0.108964 | -0.045737 | -0.004569 | -0.175796 | -0.000131 | -0.058171 | 1.000000 | -0.109972 | 0.026741 | -0.027223 | -0.005217 |
| speechiness | -0.082298 | -0.083789 | 0.212231 | 0.053230 | -0.079957 | 0.031030 | 0.094426 | 0.069719 | -0.109972 | 1.000000 | 0.059671 | 0.062756 | 0.008310 |
| tempo | 0.015132 | -0.129319 | -0.123852 | 0.157895 | -0.038975 | 0.001768 | 0.033875 | 0.129403 | 0.026741 | 0.059671 | 1.000000 | -0.002129 | 0.038256 |
| time_signature | -0.000684 | -0.158070 | 0.128823 | 0.141264 | -0.069474 | -0.003766 | 0.008342 | 0.114181 | -0.027223 | 0.062756 | -0.002129 | 1.000000 | 0.090526 |
| audio_valence | -0.056156 | -0.124809 | 0.329055 | 0.316472 | -0.182732 | 0.018772 | 0.013258 | 0.199856 | -0.005217 | 0.008310 | 0.038256 | 0.090526 | 1.000000 |

Рисунок 3 – Корреляционная матрица параметров песен

В представленной корреляционной матрице (рис. 3) выделяются несколько высоких по модулю значений коэффициентов парной корреляции, указывающих на значительную взаимосвязь между определенными предикторами. Наибольшее по модулю значение демонстрирует коэффициент парной корреляции между “energy” и “acousticness”. Отрицательный знак коэффициента обуславливается тем, что эти 2 показателя часто исключают друг друга. Еще одним значительным по абсолютному значению является коэффициент парной корреляции между “energy” и “ loudness, что логично, учитывая, что самые энергичные песни обычно исполнятся и воспроизводятся на высокой громкости для большей динамики. Остальные параметры, в среднем, демонстрируют умеренную или слабую взаимосвязь. В плане построения и оценки регрессионных моделей мы рассмотрим как включение указанных параметров, так и их исключение.

Построим сначала полную регрессионную модель на основе полного набора параметров

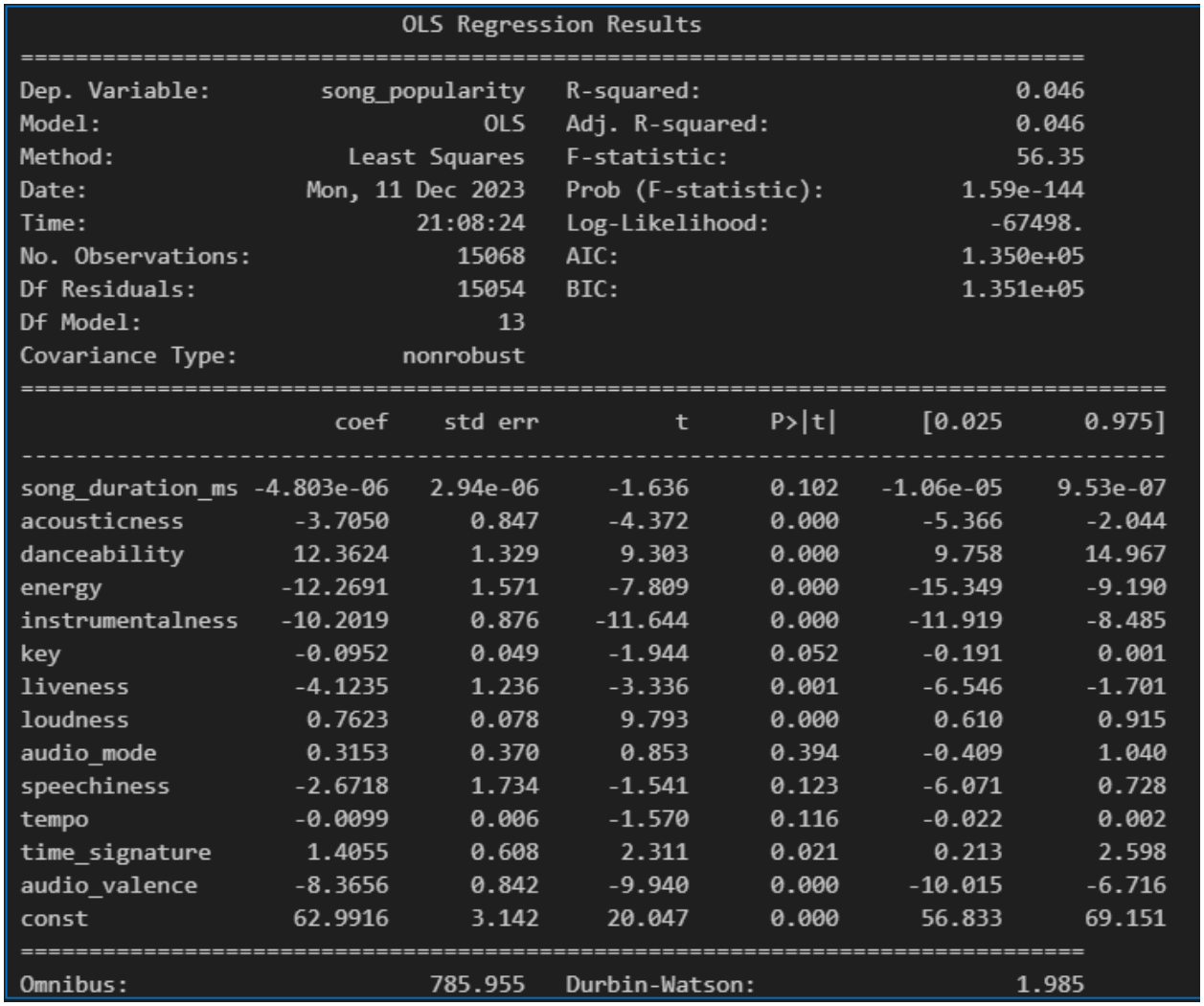
```python

model = OLS(y, x)

res = model.fit()

print(res.summary())

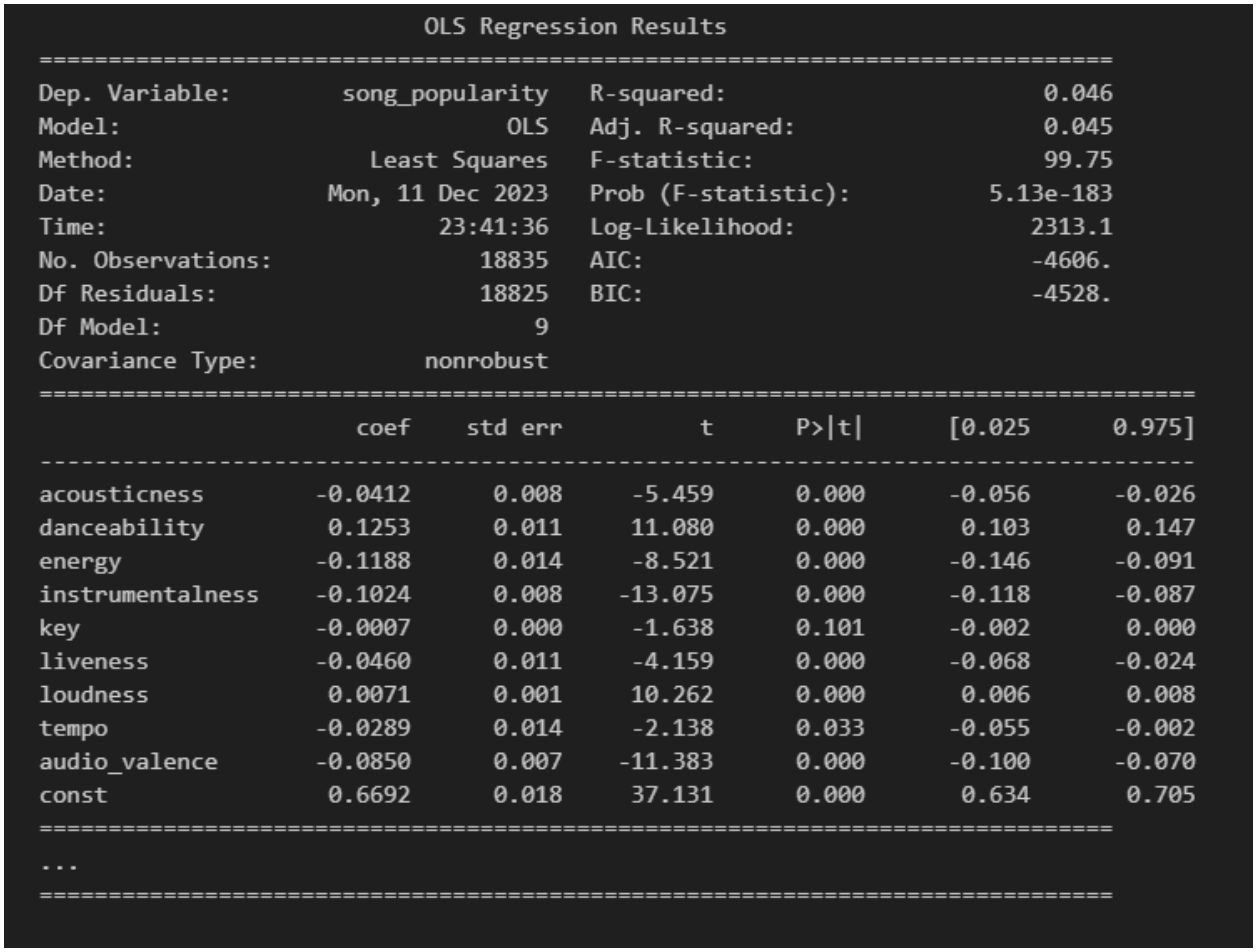
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:        song_popularity   R-squared:                       0.046
Model:                            OLS   Adj. R-squared:                  0.046
Method:                 Least Squares   F-statistic:                     56.35
Date:                Mon, 11 Dec 2023   Prob (F-statistic):          1.59e-144
Time:                        21:08:24   Log-Likelihood:                -67498.
No. Observations:               15068   AIC:                         1.350e+05
Df Residuals:                   15054   BIC:                         1.351e+05
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
song_duration_ms -4.803e-06   2.94e-06     -1.636      0.102   -1.06e-05    9.53e-07
acousticness        -3.7050      0.847     -4.372      0.000      -5.366      -2.044
danceability        12.3624      1.329      9.303      0.000       9.758      14.967
energy             -12.2691      1.571     -7.809      0.000     -15.349      -9.190
instrumentalness   -10.2019      0.876    -11.644      0.000     -11.919      -8.485
key                 -0.0952      0.049     -1.944      0.052      -0.191       0.001
liveness            -4.1235      1.236     -3.336      0.001      -6.546      -1.701
loudness             0.7623      0.078      9.793      0.000       0.610       0.915
audio_mode           0.3153      0.370      0.853      0.394      -0.409       1.040
speechiness         -2.6718      1.734     -1.541      0.123      -6.071       0.728
tempo               -0.0099      0.006     -1.570      0.116      -0.022       0.002
time_signature       1.4055      0.608      2.311      0.021       0.213       2.598
audio_valence       -8.3656      0.842     -9.940      0.000     -10.015      -6.716
const               62.9916      3.142     20.047      0.000      56.833      69.151
==============================================================================
Omnibus:                      785.955   Durbin-Watson:                   1.985
```

Рисунок 5 – Результат построения регрессионной модели

Как можно видеть многие коэффициенты (рис. 5) оценены как статистически незначимые. В связи с этим мы провели пошаговую регрессию, результаты которой представлены на рисунке 6. В этой модели остались только те предикторы, которые имеют статистическую значимость.
```



OLS Regression Results

Dep. Variable:	song_popularity	R-squared:	0.046
Model:	OLS	Adj. R-squared:	0.045
Method:	Least Squares	F-statistic:	99.75
Date:	Mon, 11 Dec 2023	Prob (F-statistic):	5.13e-183
Time:	23:41:36	Log-Likelihood:	2313.1
No. Observations:	18835	AIC:	-4606.
Df Residuals:	18825	BIC:	-4528.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
acousticness	-0.0412	0.008	-5.459	0.000	-0.056	-0.026
danceability	0.1253	0.011	11.080	0.000	0.103	0.147
energy	-0.1188	0.014	-8.521	0.000	-0.146	-0.091
instrumentalness	-0.1024	0.008	-13.075	0.000	-0.118	-0.087
key	-0.0007	0.000	-1.638	0.101	-0.002	0.000
liveness	-0.0460	0.011	-4.159	0.000	-0.068	-0.024
loudness	0.0071	0.001	10.262	0.000	0.006	0.008
tempo	-0.0289	0.014	-2.138	0.033	-0.055	-0.002
audio_valence	-0.0850	0.007	-11.383	0.000	-0.100	-0.070
const	0.6692	0.018	37.131	0.000	0.634	0.705

...

Рисунок 6 – Результат пошаговой регрессии

Видим, что после удаления таких предикторов как song_duration_ms, speechiness, time_signature, audio_mode коэффициент детерминации совсем не изменился. Это показывает, что данные предикторы совсем никак не объясняли изменчивость нашей зависимой переменной.

Разделим набор данных на обучающую и тестовую выборку.

```python
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=50)
```

Далее создадим модель линейной регрессии и обучим данную модель на части датасета

```python
model = LinearRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
y_pred_all = model.predict(x)
```

Визуализируем данные

```python
plt.scatter(y, y_pred_all, label="All data", color='blue', alpha=0.5)
plt.scatter(y_test, y_pred, label="'Test data", color='green', alpha=0.5)
plt.xlabel("Actual total length")
plt.ylabel("Predicted total length")
plt.legend()
plt.show()
```

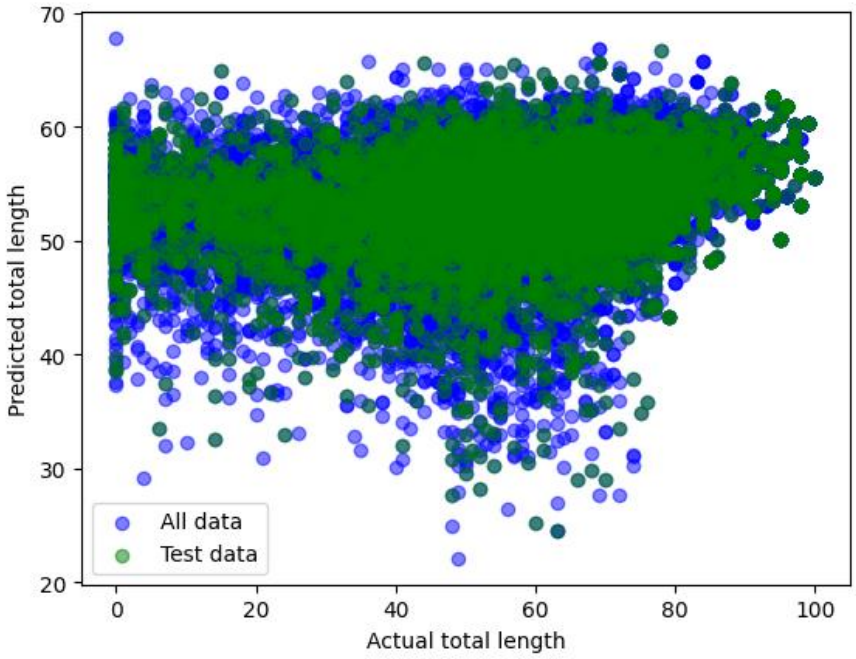


Рисунок 7 – Тестирование модели на контрольной выборке данных

Вывод коэффициентов модели. Каждый коэффициент модели указывает, насколько изменяется целевая переменная при изменении соответствующих.

```python
mse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)
coefficients = model.coef_
intercept = model.intercept_
print("Model coefficients:")
for feature, coef in zip(x.columns, coefficients):
    print(f"{feature}: {coef}")
print(f"Intercept: {intercept}")
print(f"Mean square error (MSE): {mse}")
```

```
print(f"Coefficient of determination (R2): {r2}")
```

3.3 Результаты решения

В результате тестирования линейной регрессии с сокращенным набором предикторов мы получили следующие коэффициенты модели, которые влияют на общую популярность песен:

- acousticness: -3.9832063479783506
- danceability: 12.400041991338792
- energy: -13.491392610509521
- instrumentalness: -10.158490767126912
- liveness: -5.010485299849293
- loudness: 0.7929651916140532
- audio_valence: -7.824046905177627
- const: 0.0
- Intercept: 66.66020041226525
- Mean square error (MSE): 21.671602644343537
- Coefficient of determination (R2): 0.04021417340330269

Intercept определяет значение общей популярности песен, когда все характеристики равны нулю. В данном случае, среднеквадратическая ошибка является весьма надежным показателем, поскольку она основывается на среднем значении целевой переменной. Коэффициент детерминации же отражает процентное отклонение целевой переменной.

Заключение

В моей работе был проведен регрессионный анализ для предсказания длины туловища особей опоссумов на основе признаков. С помощью библиотеки scikit-learn была построена модель линейной регрессии.

Результаты коэффициента детерминации показали, что показатель популярности песни хоть и имеет свои значимые предикторы, но на общую картину это мало как влияет. На популярность песни играют сразу несколько факторов, которые взаимосвязаны. Также стоит учитывать, что зачастую при выборе песни люди зачастую опираются на субъективные чувства.

Список литературы

1. Song Popularity Dataset [Электронный ресурс]. – Режим доступа: https://www.kaggle.com/datasets/yasserh/song-popularity-dataset/data (дата обращения: 9.12.2020).

2. Линейная регрессия в Python [Электронный ресурс]. – Режим доступа: https://education.yandex.ru/handbook/data-analysis/article/pandan-linejnaya-regressiya-v-python (дата обращения: 9.12.2020).

3. Регрессионные модели в Python [Электронный ресурс]. – Режим доступа: https://nagornyy.me/it/regressionnye-modeli-v-python/ (дата обращения: 9.12.2020).

4. Как создать корреляционную матрицу в Python [Электронный ресурс]. – Режим доступа: https://www.codecamp.ru/blog/correlation-matrix-in-python/ (дата обращения: 9.12.2020).

4. Основы регрессионного анализа [Электронный ресурс]. – Режим доступа: https://pro.arcgis.com/ru/pro-app/latest/tool-reference/spatial-statistics/regression-analysis-basics.htm (дата обращения: 9.12.2020).