Лабораторная работа №1 – Линейная регрессия

Линейная регрессия

Одна из главных областей статистики касается возможности делать прогнозы. С помощью регрессии можно прогнозировать одну переменную на основе другой. Прогнозирование - это процесс оценки значения одной переменной при известном значении другой.

Далее мы рассмотрим простую линейную регрессию (одна зависимая и одна независимая переменная), где связь между ними можно описать прямой линией в системе координат.

Регрессия тесно связана с концепцией корреляции. Сильная ассоциация между двумя переменными позволяет делать более точные прогнозы.

Пример проекта по созданию линейной модели:

Компания электронной торговли, расположенная в Нью-Йорке, продает одежду онлайн, но также предлагает консультации по стилю в своих фирменных магазинах. Клиенты могут записаться на личную встречу со стилистом, после чего они могут купить одежду через мобильное приложение или сайт.

Компания пытается решить, на чем сосредоточить усилия: на мобильном приложении или веб-сайте. Они наняли вас по контракту, чтобы помочь им разобраться! Давайте начнем!

Просто следуйте шагам ниже, чтобы проанализировать данные клиентов (это фальшивые данные, не беспокойтесь, мы не передали реальные номера кредитных карт или электронные почты).

1. Импорт данных

Шаг 1: Импортируйте pandas, numpy, matplotlib и seaborn. Затем установите %matplotlib inline (позже при необходимости импортируйте sklearn).

2. Получение данных

Мы будем работать с файлом Ecommerce Customers CSV от компании. Он содержит информацию о клиентах, такую как e-mail, адрес и цвет аватара, а также числовые значения:

- ***Avg. Session Length:*** средняя продолжительность сеансов консультации в магазине.
- ***Time on App:*** среднее время, проведенное в приложении (в минутах).
- ***Time on Website:*** среднее время, проведенное на сайте (в минутах).
- ***Length of Membership:*** сколько лет клиент является участником программы.

Шаг 2: Прочитайте файл Ecommerce Customers CSV в DataFrame под названием customers.

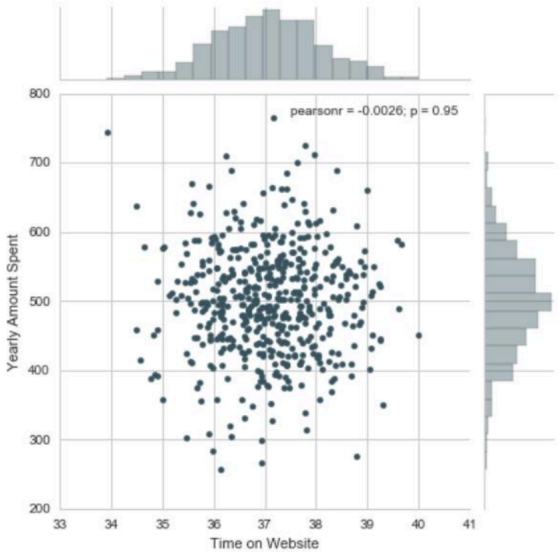
Шаг 3: Выведите первые строки (head) данных о клиентах и получите информацию о них с помощью функций info() и describe().

3. Исследовательский анализ данных

В оставшейся части упражнения мы будем работать только с числовыми данными из CSV-файла.

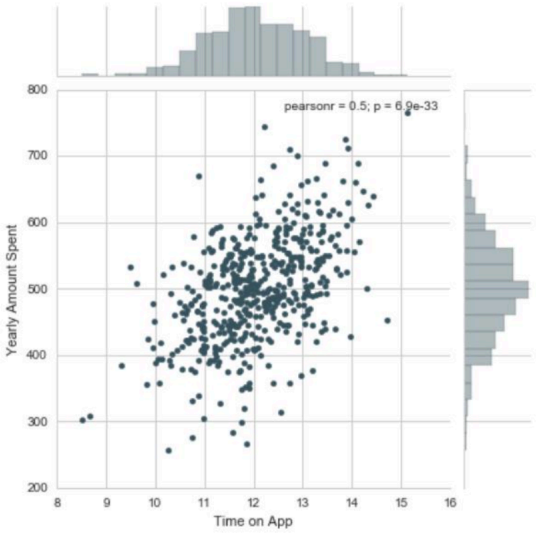
Используйте seaborn для создания jointplot (совместной диаграммы) для сравнения колонок Time on Website и Yearly Amount Spent. Есть ли корреляция?

```
<seaborn.axisgrid.JointGrid at 0x120bfcc88>
```



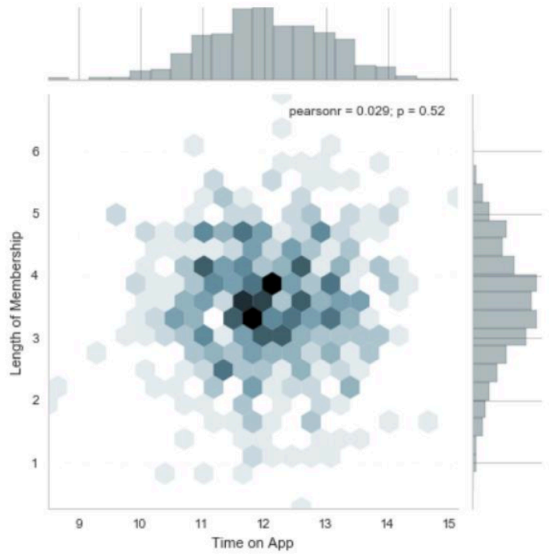
Шаг 4: Повторите этот анализ для Time on App.

```
<seaborn.axisgrid.JointGrid at 0x132db5908>
```



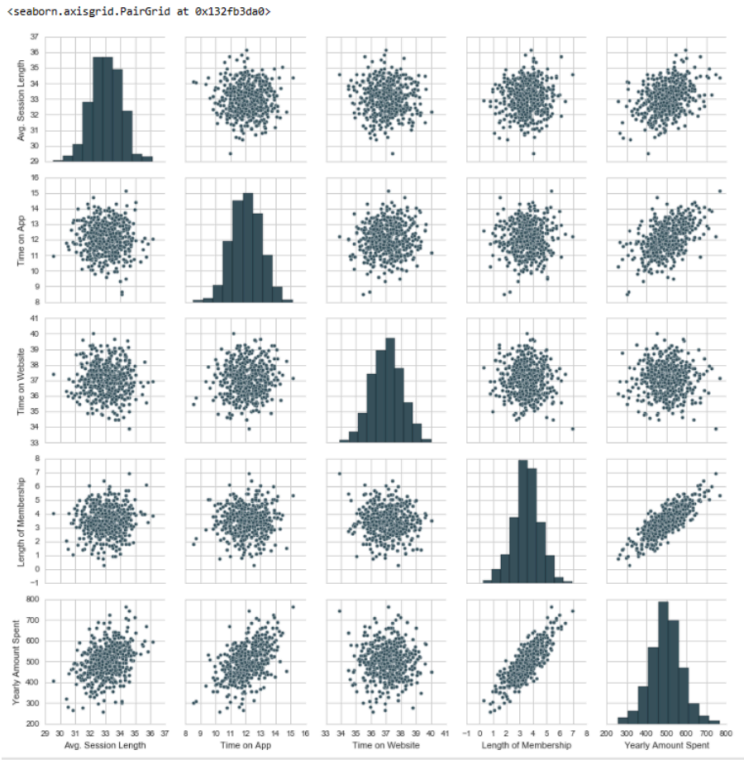
Шаг 5: Используйте ***jointplot()*** для построения 2D-диаграммы, сравнивающей Time on App и Length of Membership.

```
<seaborn.axisgrid.JointGrid at 0x130edac88>
```



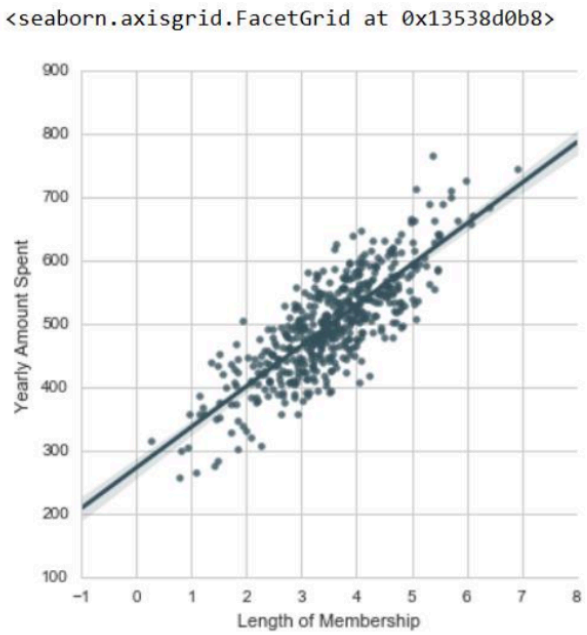
(kind='hex')

Шаг 6: Исследуйте взаимосвязи во всем наборе данных. Используйте pairplot, чтобы воссоздать приведенный ниже график. (Не беспокойтесь о цветах.)



На основе этого графика, какая характеристика кажется наиболее коррелированной с Yearly Amount Spent?

Шаг 7: Создайте линейную модель графически (используя lmplot из seaborn) для данных Amount Spent vs. Length of Membership.



4. Обучение и тестирование данных

Разделим данные на обучающий и тестовый наборы.

Шаг 8: Определите переменную X, содержащую числовые характеристики клиентов, и переменную y, содержащую колонку Yearly Amount Spent.

Шаг 9: Используйте model_selection.train_test_split из sklearn для разделения данных на обучающий и тестовый наборы. Установите test_size=0.3 и random_state=101.

5. Обучение модели

Теперь обучим модель на обучающем наборе данных!

Шаг 10: Импортируйте LinearRegression из sklearn.linear_model.

Шаг 11: Создайте экземпляр модели LinearRegression() с именем lm.

Шаг 12: Обучите lm на обучающих данных.

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

Шаг 13: Выведите коэффициенты модели.

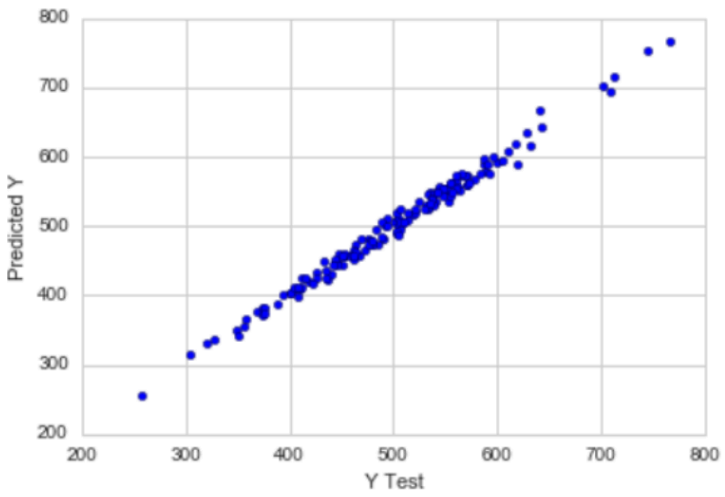
6. Прогнозирование тестовых данных

Оценим производительность модели, прогнозируя тестовые значения!

Шаг 14: Используйте lm.predict() для предсказания значений для X_test.

Шаг 15: Постройте диаграмму рассеяния реальных значений теста против предсказанных значений.

```
<matplotlib.text.Text at 0x135546320>
```



7. Оценка модели

Оценим производительность модели, вычислив сумму квадратов остатков и коэффициент детерминации (R^2).

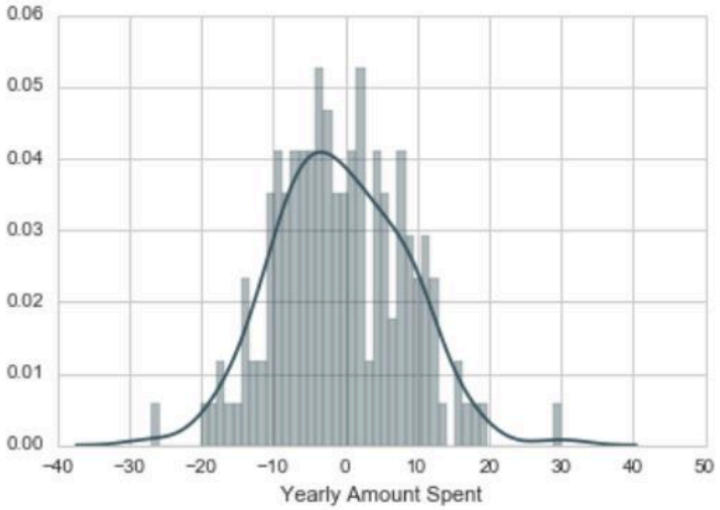
Шаг 16: Вычислите среднюю абсолютную ошибку (MAE), среднеквадратичную ошибку (MSE) и корень среднеквадратичной ошибки (RMSE). Обратитесь к лекции или Wikipedia для формул.

```
MAE: 7.22814865343
MSE: 79.813051651
RMSE: 8.93381506698
```

8. Остатки

Быстро исследуем остатки, чтобы убедиться, что с данными все в порядке.

Шаг 17: Постройте гистограмму остатков и убедитесь, что они распределены нормально. Используйте либо seaborn distplot, либо plt.hist().



9. Выводы

Все еще хотим ответить на изначальный вопрос: стоит ли сосредоточиться на мобильном приложении или веб-сайте? Или, возможно, наиболее важным фактором является продолжительность членства клиента?

Шаг 18: Воссоздайте приведенный ниже набор данных.

Шаг 19: Ответьте на вопрос: Как можно интерпретировать эти коэффициенты?

	Coeffecient
Avg. Session Length	25.981550
Time on App	38.590159
Time on Website	0.190405
Length of Membership	61.279097

Шаг 20: Ответьте на вопрос: Должна ли компания сосредоточиться на мобильном приложении или веб-сайте?

Индивидуальное задание: Посетите https://www.kaggle.com/datasets, выберите и загрузите набор данных для создания линейной модели. При необходимости очистите данные и аргументируйте выбранный метод. Пройдите шаги 1-20, как в примере выше, обоснуйте корреляцию между колонками выбранного набора данных и оцените предсказание.