Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №0 по курсу «Искусственный интеллект»

Тема: Анализ и подготовка данных

Студент: А.С. Федоров

Преподаватель: Самир Ахмед

Группа: М8О-307Б-19

Дата:

Оценка:

Подпись:

Москва, 2022

Задача:

Требуется определить задачу и найти под нее соответствующие данные. Проанализировать и подготовить данные, визуализировать зависимости.

Описание

Подготовка данных для обучения модели – важный этап в ее разработке. Анализировать данные проще всего с помощью визуализаций, на которых можно быстро определить необходимые характеристики и предпринять соответствующие меры.

Ход работы

Загрузка данных

В качестве задачи выберу актуальную проблему предсказания оттока абонентов. Формулировка: на основе косвенных данных (активное время абонента, количество проговоренных минут, количество звонков в тех поддержку и т.д.) предсказать, откажется ли абонент от услуг компании. Использую датасет с сайта Kuggle. Ссылка: https://www.kaggle.com/datasets/barun2104/telecom-churn. В датасете 8 числовых (активность абонента в неделях, количество используемого трафика, количество звонков в техподдержку, количество израсходованных минут за день, количество звонков за день, средний месячный счет, самая большая плата за перерасход за последние 12 месяцев) и два категориальных признака (есть ли тарифный план, продлил ли абонент договор недавно).

Загрузка и последующая проверка общих сведений о данных не выявило ничего необычного.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Churn            3333 non-null   int64  
 1   AccountWeeks     3333 non-null   int64  
 2   ContractRenewal  3333 non-null   int64  
 3   DataPlan         3333 non-null   int64  
 4   DataUsage        3333 non-null   float64
 5   CustServCalls    3333 non-null   int64  
 6   DayMins          3333 non-null   float64
 7   DayCalls         3333 non-null   int64  
 8   MonthlyCharge    3333 non-null   float64
 9   OverageFee       3333 non-null   float64
 10  RoamMins         3333 non-null   float64
dtypes: float64(5), int64(6)
memory usage: 286.6 KB
```

Обычно, если с каким-то признаком что-то не так его тип – это object. Так происходит, если в столбце встречаются строки, что было бы странно

для фчисловых признаков. Также по ссыке было обозначено, что категориальные признаки уже закодированны One-Hot кодированием. Примем это к сведенью. Тип категориальных прихзнаков целочисленный, что и ожидалось.

Возможно, в каких-то полях таблицы присутствуют NaN-ы. Проверю это.

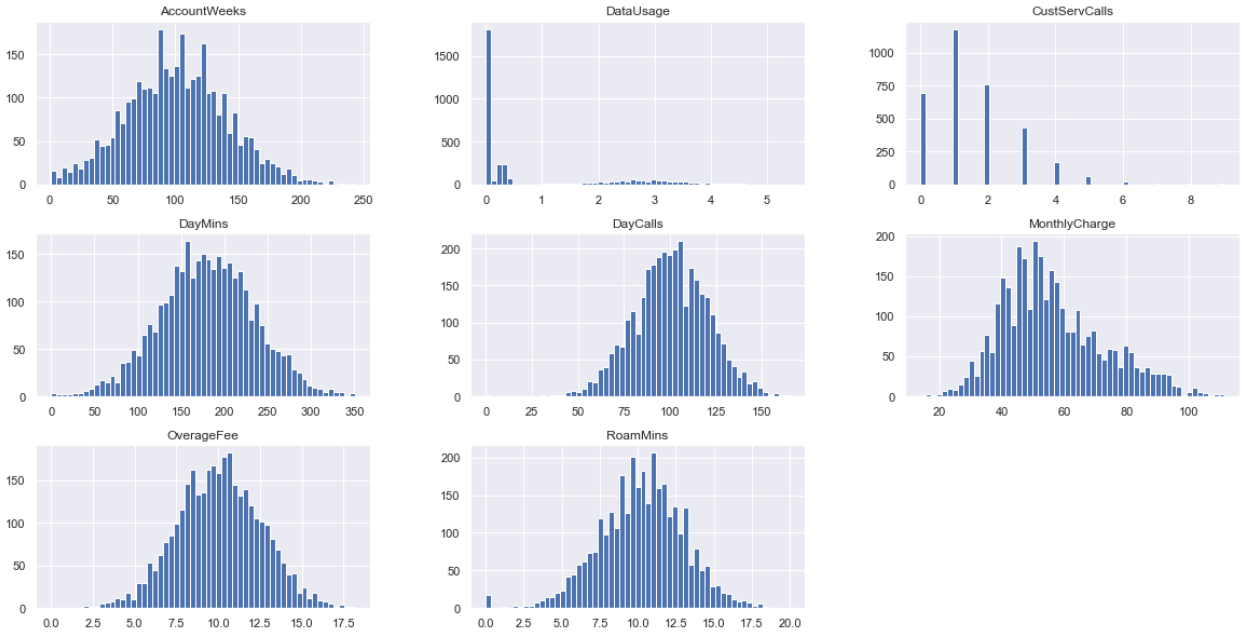
```
# Проверю, если NaN-ы
data.isnull().values.any()
```

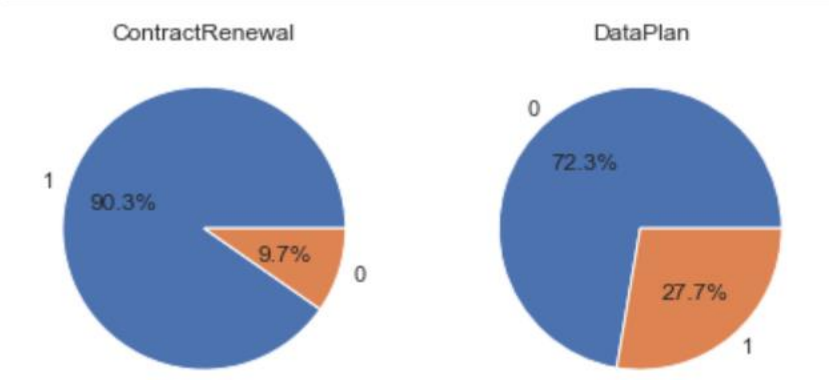
```
False
```

NaN-в нет, отлично. Никаких дополнительных действий по этому поводу делать не нужно.

Визуализация

Визуализирую данные, чтобы иметь более четкое представление о них. Числовые признаки визуализирую гистограммами, а категориальные круговыми диаграммами.





Никаких аномалий в распределении числовых признаков не обнаружено. Можно удивиться, почему на второй гистограмме такой высокий первый столбец, но так как это количество израсходованного трафика в гигабайтах, можно предположить, что многие абоненты которые не имеют тарифного плана не могут, как следствие, расходовать трафик.

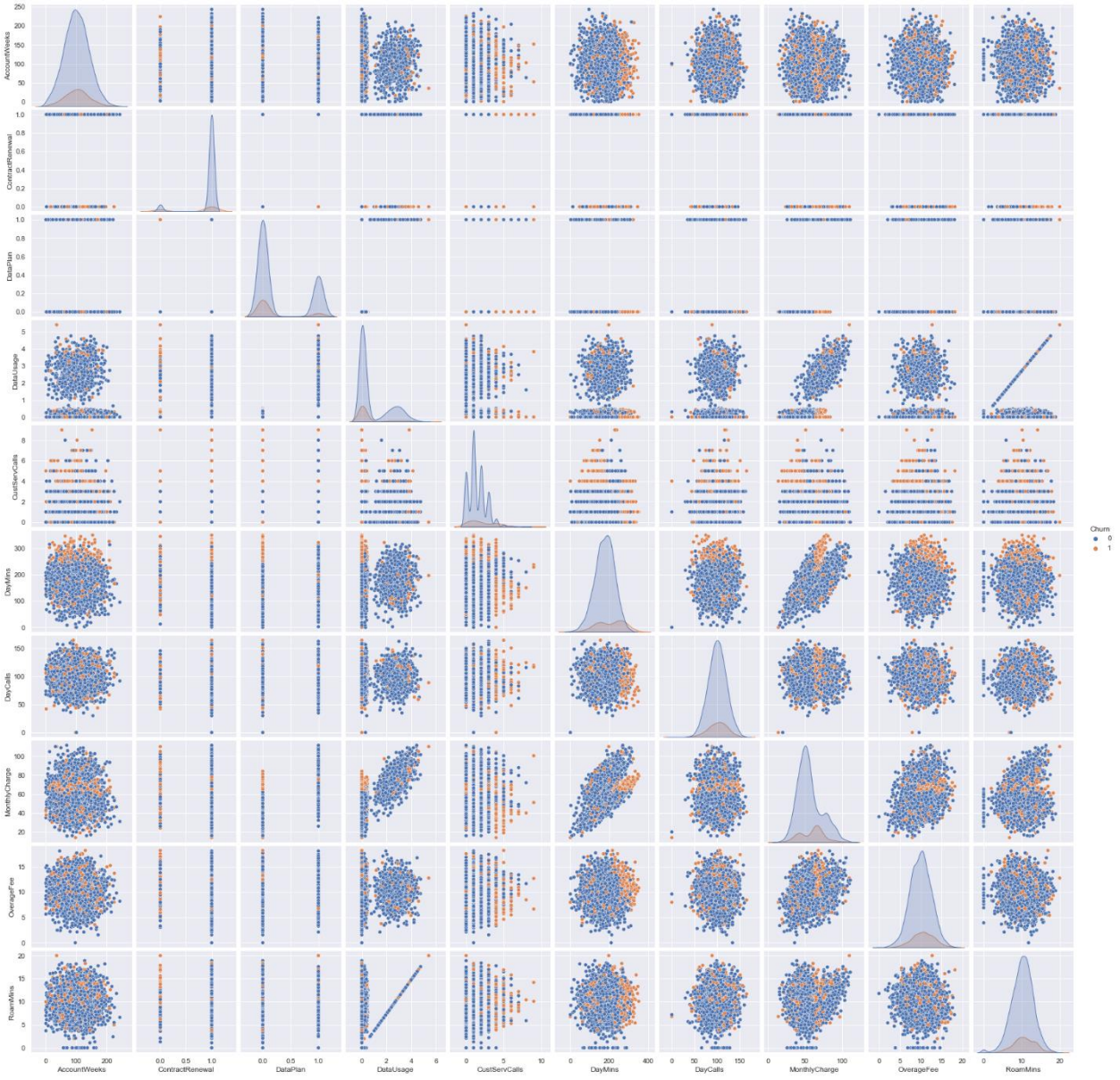
Так как в бедующем я планирую обучать линейную модель, то хорошо понять сразу, на сколько признаки коррелируют с предсказываемым результатом. Подсчитаю корреляцию.

```
data.corrwith(data[target_col])
```

```
Churn              1.000000
AccountWeeks       0.016541
ContractRenewal   -0.259852
DataPlan          -0.102148
DataUsage         -0.087195
CustServCalls      0.208750
DayMins            0.205151
DayCalls           0.018459
MonthlyCharge      0.072313
OverageFee         0.092812
RoamMins           0.068239
dtype: float64
```

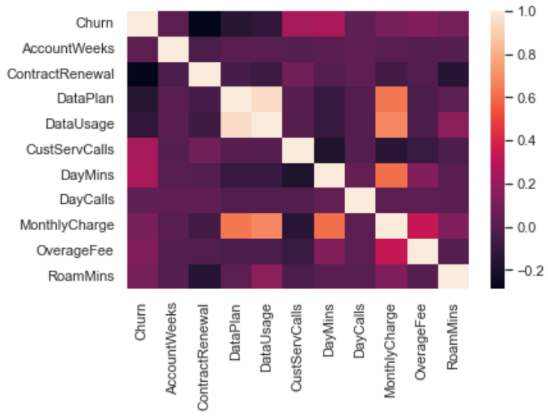
Похоже, что ни один признак явно не коррелирует с предсказываемым признаком. Однако, стоит отметить относительно хороший результат для ContactRenewal, CustServCalls, DayMins. Значит можно их считать наиболее важными.

Также, для более ясного представления о данных, построю парные графики для всех пар признаков.



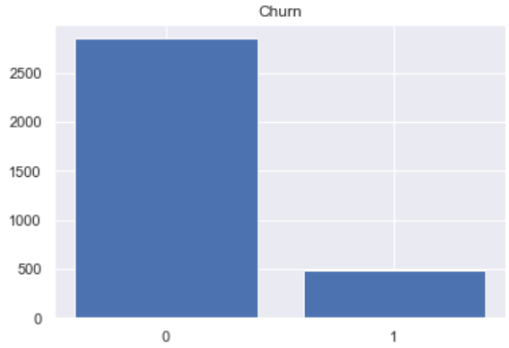
В большинстве пар провести достаточно хорошую разделяющую кривую не представляется возможным. Однако несколько пар дали довольно хорошую группировку, хорошо, это значит, что есть шансы чего-то достичь линейной моделью.

Посмотрю на матрицу корреляции для признаков.



Признаки, имеющие высокую корреляцию, можно считать бесполезными, так как один признак может выражаться посредством другого. Из визуализации видно, что высокую корреляцию имеют признаки DataPlan и DataUsage. Однако не буду считать это критичным, так как один является категориальным, а второй числовым. Также есть довольно высокая корреляция DataPlan с другими числовыми характеристиками, отражающими объем потребляемых услуг. Также не считаю это критичным.

Напоследок, взгляну на соотношение классов, которые нужно предсказать.



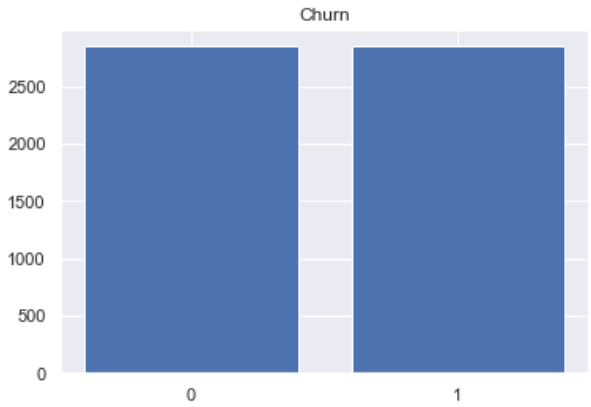
Наблюдается некоторый дисбаланс классов, что может плохо сказаться на результатах модели. Выровняю выборку с помощью оверсемплинга.

```
sample = data[data[target_col] == 1]
sample.shape
```

```
(2850, 11)
```

```
while data[data[target_col] == 1].shape[0] + sample.shape[0] < data[data[target_col] == 0].shape[0]:
    data = data.append(sample)
data = data.append(sample.iloc[:data[data[target_col] == 0].shape[0]- \
                               data[data[target_col] == 1].shape[0]])
```

Взгляну на соотношение теперь.



Так как изначальные признаки дают слабую корреляцию, можно выдумать искусственные, скомбинировав изначальные. Однако, эта задача довольна сложна, для человека, не разбирающегося в сфере, к которой относятся данные. Поэтому на этом считаю данные готовыми для обучения на них линейной модели.

Вывод

В ходе выполнения лабораторий работы был проведет анализ данных с целью подготовки их для обучения линейной модели. Дисбаланс классов был устранён методом оверсемплинга. Визуализация данных различными способами позволили сделать некоторые выводы. Таким образом был обнаружен дисбаланс классов и была произведена поверхностная оценка выразительности бедующей линейной модели.