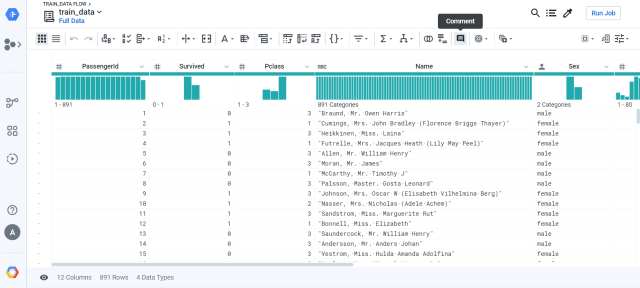
**Trifacta для машинного обучения**

Dataprep (Trifacta в Google Cloud Platform), недооцененный продукт облачной платформы Google, который можно использовать для подготовки данных для анализа и машинного обучения.

Trifacta с легкостью интеллектуально исследует, очищает и преобразует данные за пару минут с минимальными усилиями; по сравнению с Pandas и другими инструментами предварительной обработки вам не нужно писать код для подготовки данных с помощью Trifacta.

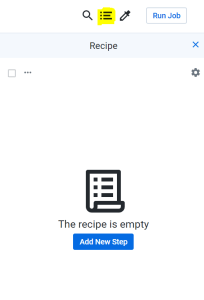
Чтобы понять возможности Trifacta, мы используем самый популярный набор обучающих данных Kaggle [Titanic](https://www.kaggle.com/c/titanic/data) в графическом интерфейсе следующим образом.



Вот как это выглядит интерфейс Trifacta, когда мы импортируем данные и начинаем их обрабатывать.

Мы видим строки нашего датасета, как показано выше, они содержат 12 столбцов отображаемых гистограммой, 891 строку и 4 типа данных. Со временем мы изменим схему по необходимости.

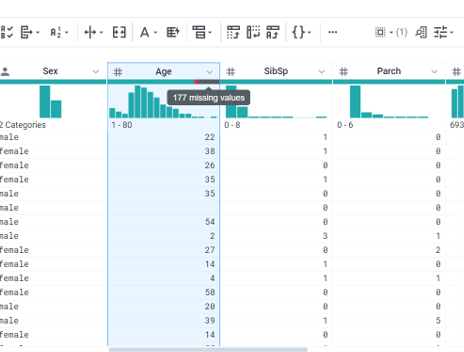
В Trifacta в Рецепт записываются все шаги по очистке, преобразованию данных и другие операции, чтобы Рецепт можно было экспортировать и использовать для подготовки аналогичных данных с той же схемой без необходимости создавать рецепт снова с нуля.



Рецепт в Trifacta

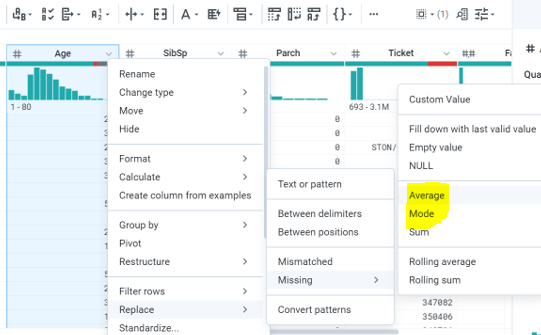
**Отсутствующие значения**

Определите недостающие значения и замените их на среднее или moda. В нашем наборе данных пропущенные значения в столбцах (Возраст) обозначены серым цветом в столбце, как показано ниже.



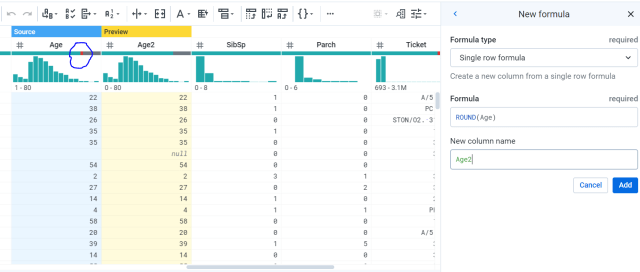
Исследование данных в Trifacta

Затем щелкните раскрывающийся список столбца Возраст и замените отсутствующие значения, используя среднее значение или моду (mode), как показано ниже.

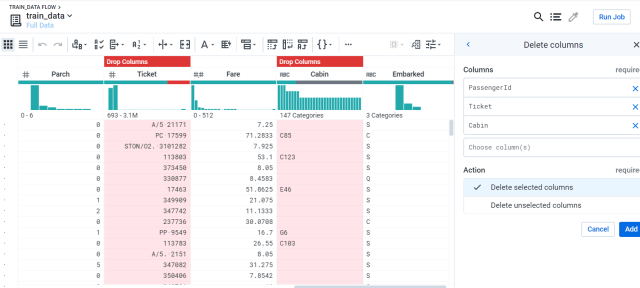


Замена отсутствующе значений в Trifacta

Продолжайте выполнять аналогичные действия всех столбцов с пропущенными значениями. В случае несовпадения значений, как показано ниже красным цветом, используйте функцию округления, чтобы округлить значения и добавить их в рецепт.

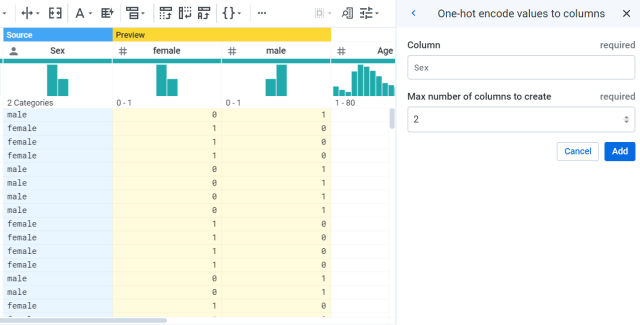
Округление значений в Trifacta

Если столбец содержит много пропущенных значений или это не очень важно для обучения данных, мы можем удалить столбцы, как показано ниже.

удаление столбцов с пропущенными значениями в Trifacta

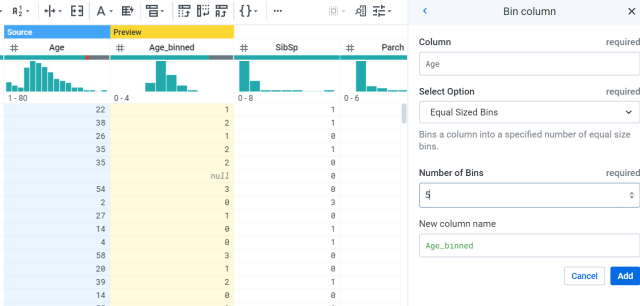
**One-Hot Encoding**

Чтобы применить одну горячую кодировку, мы можем найти One-Hot Encoding в строке поиска рецептов и указать имя столбца и количество новых создаваемых столбцов, как показано ниже для столбца *пол,* и выполнить ту же процедуру для остальных категориальных столбцов.

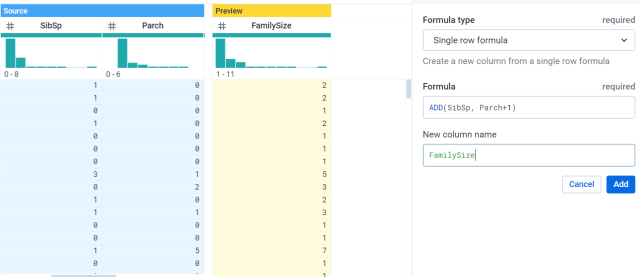
 One-Hot Encoding в Trifacta

**Биннинг**

Если нам нужно разделить значения на диапазоны равного или нестандартного размера? найдите Bin в преобразованиях и предоставьте опцию равных по размеру или по кастомизированных участков, а также укажите количество ячеек, если участки (бины) равного размера выбраны в соответствии с требованиями, показанными ниже.

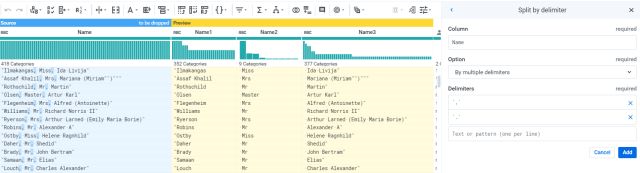
Биннинг в Trifacta

Чтобы получить новые фичи из существующих столбцов, найдите рецепт добавления и примените формулу. В нашем случае нам нужно добавить столбцы (sibsp, parch), чтобы получить новый столбец FamilySize, как показано ниже.

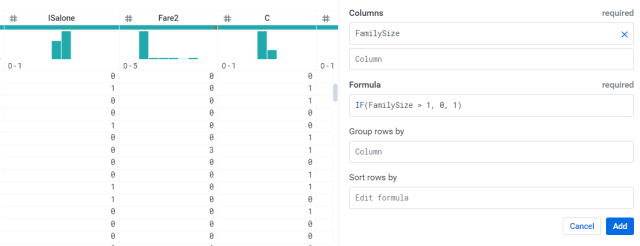
Создание новых признаков в Trifacta

**Split**

Нам нужно разделить *обращение* от *имени* в столбце *Name*, найдем **Split** в строке поиска рецептов, выберем несколько разделителей по мере необходимости, как показано ниже

Разделение по разделителю в Trifacta

Мы можем применить формулу с условием **IF,** как *показано* ниже, чтобы получить новый столбец.

Преобразование столбца в Trifacta

Вот и все, мы рассмотрели почти все преобразования (биннинг, One-Hot Encoding, пропущенные значения, несовпадающие значения и др.), необходимые для подготовки данных для машинного обучения с использованием набора данных Titanic.

Теперь пора загрузить подготовленные данные в Bigquery или другое облачное хранилище. Нажмите «Выполнить задание» в правом верхнем углу, чтобы запустить задание Trifacta в фоновом режиме, используя созданный нами рецепт преобразования данных.

Если вы хотите повторно использовать рецепт, создайте копию потока, содержащего рецепт, и измените входной набор данных и при необходимости измените рецепт.

**Прогноз выживания на Титанике в BigQuery**

BigQuery ML - это функция внутри BigQuery, бессерверного хранилища данных от Google, которое позволяет аналитикам данных создавать и развертывать модели машинного обучения на массивных структурированных или полуструктурированных наборах данных с помощью простых операторов SQL.

Мы используем самый популярный набор данных [Titanic](https://www.kaggle.com/c/titanic/data) из конкурса Kaggle Competitions для новичков в области бинарной классификации. Проблема двоичной классификации - это не что иное, как предсказать, произойдет ли конкретное событие или нет. В этой задаче «Титаник» нужно предсказать, выживет ли пассажир корабля «Титаник».

Здесь мы не будем вдаваться в основы машинного обучения и исследовательского анализа данных, поскольку наша главная цель - понять функциональность Bigquery ML. Мы выполним в основном три шага

* Создание модели.
* Оцениваем модель.
* Прогнозирование с использованием модели.

**Создание модели:**

Нам нужно создать набор данных в Bigquery для хранения наших моделей перед созданием модели, загрузить файлы обучающего и тестового датасета из kaggle в набор данных Bigquery как train\_data и test\_data, мы будем использовать train\_data для создания и оценки модели.

Учитывая необходимые функции/столбцы, необходимые для обучения модели, мы запустим следующий запрос в пользовательском интерфейсе Bigquery.

CREATE OR REPLACE MODEL projectname.datasetname.modelname

OPTIONS (

model\_type = 'logistic\_reg' ,labels=['survived']

) AS (

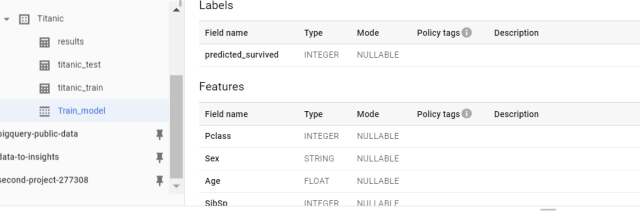
SELECT

Pclass,Sex, Age, SibSp, Parch, Fare, Embarked ,survived

FROM

`projectname.datasetname.tablename` )

В действительности это всего лишь пара операторов SQL для создания модели машинного обучения в Bigquery ML. Схема модели будет показана ниже.

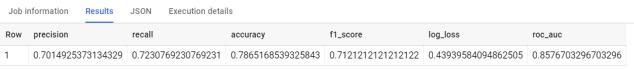


**Оценка модели:**

Это просто строка кода на SQL для оценки модели, которая оценивает предсказанные значения с фактическими значениями.

SELECT \* FROM ML.evaluate(MODEL `projectname.datasetname.tablename`)

Результаты оценки будут такими, как показано ниже.

Модель Оценила, что 78% прогнозов верны.

**Прогнозирование с использованием модели:**

Мы будем использовать ML.predict для прогнозирования значений из обученной модели для тестового набора данных test\_data

SELECT

\*

FROM

ML.PREDICT(MODEL `projectname.datasetname.modelname`, (

SELECT

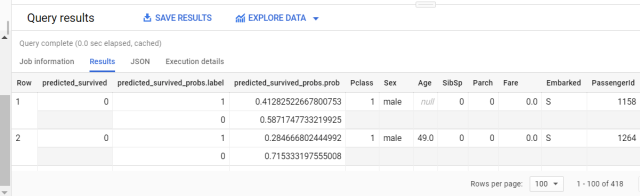
PassengerId

FROM

`projectname.datasetname.modelname`

))

После завершения запроса результат будет показан ниже.

Предсказание выживания

Это все! мы создали, оценили модель бинарной логистической регрессии с набором данных Titanic и спрогнозировали выживаемость с помощью Bigquery ML.

CREATE OR REPLACE MODEL `big-query-exam-283517.Titanic.Titanic\_model`

OPTIONS(

model\_type = 'logistic\_reg' ,labels=['survived']

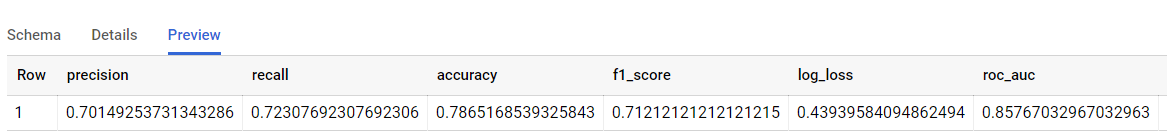
) AS (

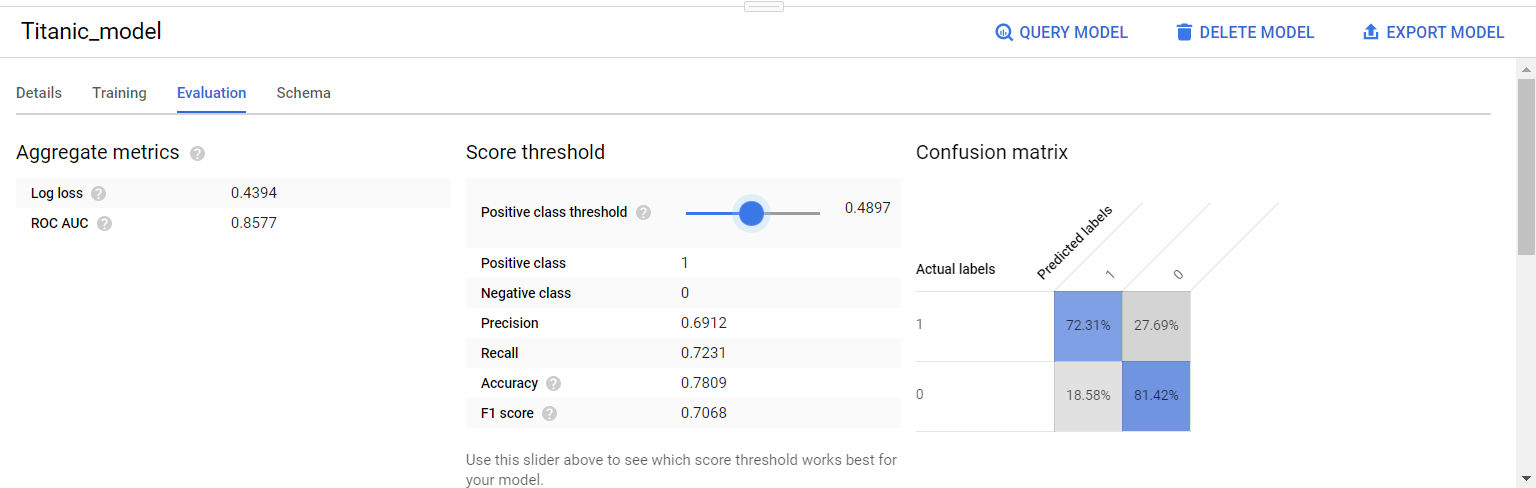
SELECT

Pclass,Sex, Age, SibSp, Parch, Fare, Embarked ,survived

FROM `big-query-exam-283517.Titanic.Titanic\_train`)

SELECT \* FROM ML.evaluate(MODEL `big-query-exam-283517.Titanic.Titanic\_model`)





SELECT

\*

FROM

ML.PREDICT(MODEL `big-query-exam-283517.Titanic.Titanic\_model`,

(

SELECT

Pclass,Sex, Age, SibSp, Parch, Fare, Embarked

FROM

`big-query-exam-283517.Titanic.Titanic\_test`

))

CREATE OR REPLACE MODEL `big-query-exam-283517.Titanic.Titanic\_model\_xgb`

OPTIONS(

model\_type = 'BOOSTED\_TREE\_CLASSIFIER' ,

AUTO\_CLASS\_WEIGHTS = TRUE,

MAX\_TREE\_DEPTH=4,

MAX\_ITERATIONS=10,

labels=['survived']

) AS (

SELECT

Pclass,Sex, Age, SibSp, Parch, Fare, Embarked ,survived

FROM `big-query-exam-283517.Titanic.Titanic\_train`)

CREATE OR REPLACE MODEL `big-query-exam-283517.Titanic.Titanic\_model\_xgb`

OPTIONS(

model\_type = 'BOOSTED\_TREE\_CLASSIFIER' ,

#AUTO\_CLASS\_WEIGHTS = TRUE,

MAX\_TREE\_DEPTH=3,

MAX\_ITERATIONS=5,

labels=['survived']

) AS (

SELECT

Pclass,Sex, Age, SibSp, Parch, Fare, Embarked ,survived

FROM `big-query-exam-283517.Titanic.Titanic\_train`)

