# Прогнозируем отток клиентов с помощью алгоритмов машинного обучения в БД Oracle (OML4SQL)

# Подготовка

1. Зайдите в Oracle Cloud с вашей учетной записью.

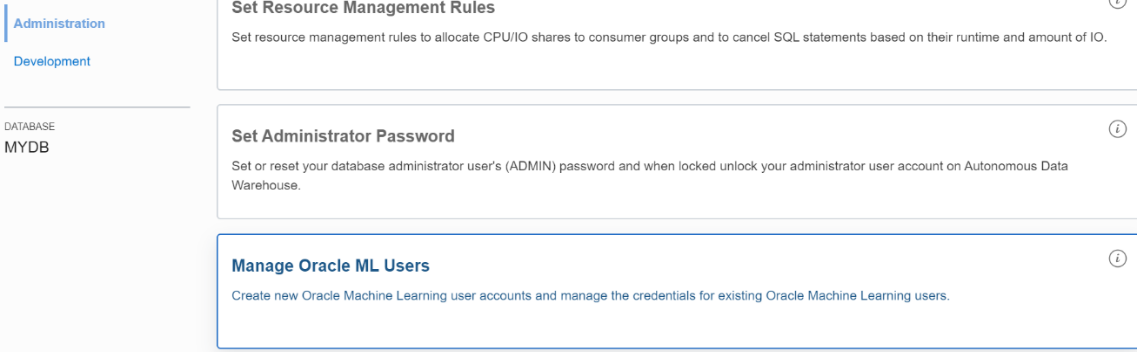
2. Создайте ADW БД с параметрами по умолчанию.



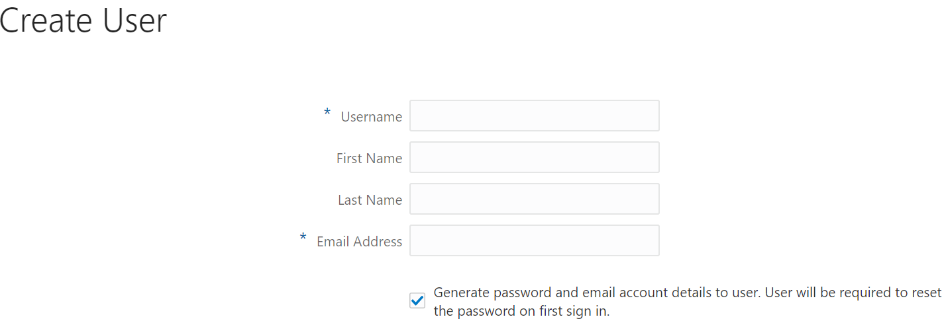
3. Дождитесь пока загорится зеленый индикатор (Available) и кликните на Service Console

## 

5. Выберите раздел Administration -> Manage Oracle ML Users

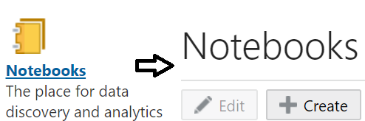


6. Создайте нового пользователя ML.



7. Откройте среду разработки нажатием на  в правом верхнем углу окна администрирования пользователей.

8. Введите имя пользователя и пароль, созданные на этапе 6.

9. Создайте новый ноутбук 

# Загрузка данных, расчет модели и анализ результатов

1. Создадим таблицу CHURN

%**sql**

**CREATE** **TABLE** CHURN

( "ROWNUMBER" **NUMBER**(38,0),

"CUSTOMERID" **NUMBER**(38,0),

"SURNAME" **VARCHAR2**(26 BYTE),

"CREDITSCORE" **NUMBER**(38,0),

"GEOGRAPHY" **VARCHAR2**(26 BYTE),

"GENDER" **VARCHAR2**(26 BYTE),

"AGE" **NUMBER**(38,0),

"TENURE" **NUMBER**(38,0),

"BALANCE" **NUMBER**(38,2),

"NUMOFPRODUCTS" **NUMBER**(38,0),

"HASCRCARD" **NUMBER**(38,0),

"ISACTIVEMEMBER" **NUMBER**(38,0),

"ESTIMATEDSALARY" **NUMBER**(38,2),

"EXITED" **VARCHAR2**(26 BYTE)

);

2. Импортируем данные из файла с разделителями

**%script**

**BEGIN**

**DBMS\_CLOUD.copy\_data**(

table\_name => 'CHURN',

file\_uri\_list => 'https://objectstorage.eu-frankfurt-1.oraclecloud.com/n/cissandbox/b/osir-ml-workshops/o/Bank\_Churn\_Modelling.csv',

format => json\_object('type' **VALUE** '', 'delimiter' **VALUE** ';',

'ignoremissingcolumns' **VALUE** 'true', 'skipheaders' **VALUE** '1',

'numericcharacters' **VALUE** ',.')

);

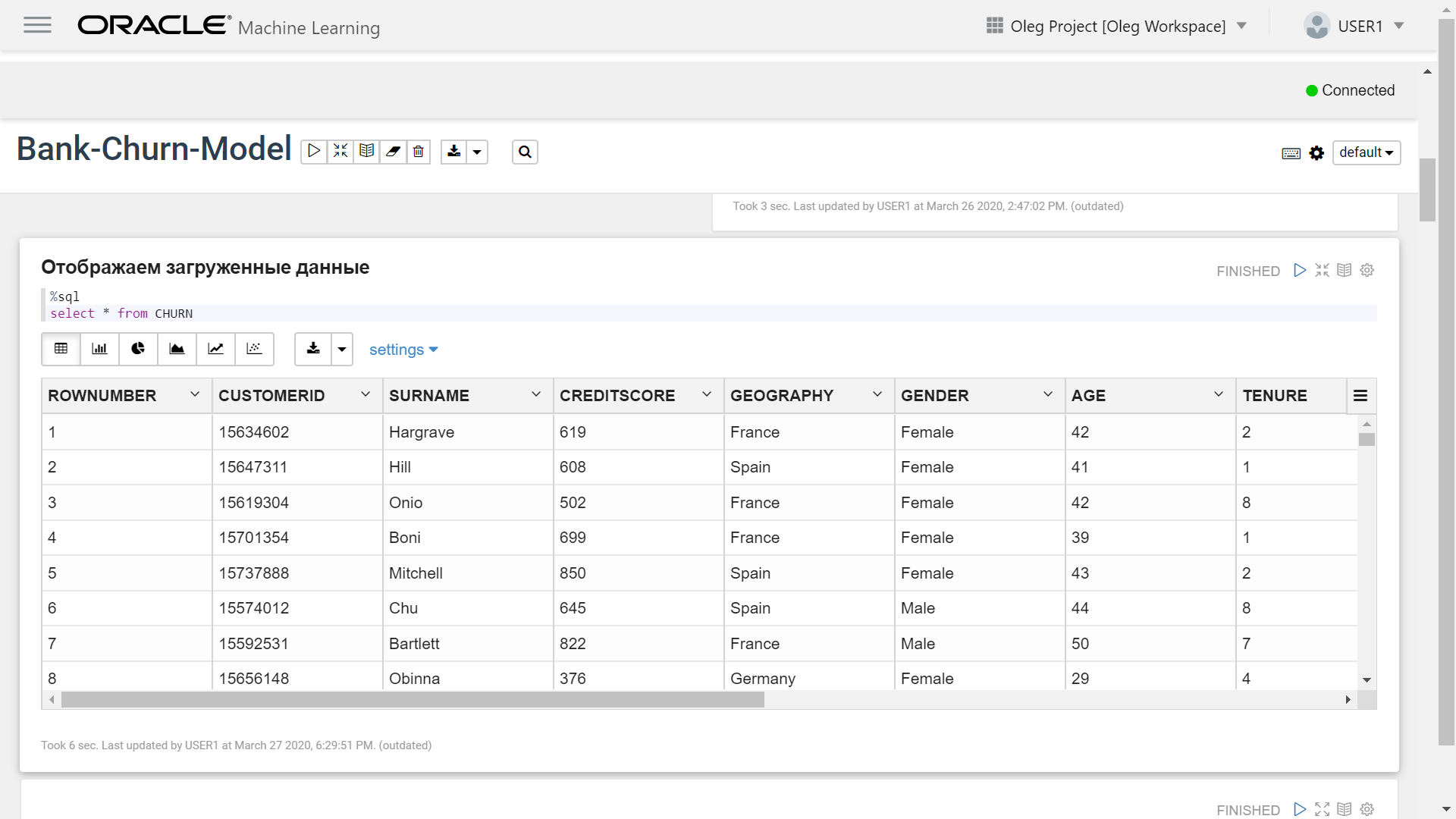
**END**;

/

3. Исследуем данные в таблице CHURN

%**sql**

**SELECT** \* **FROM** CHURN;

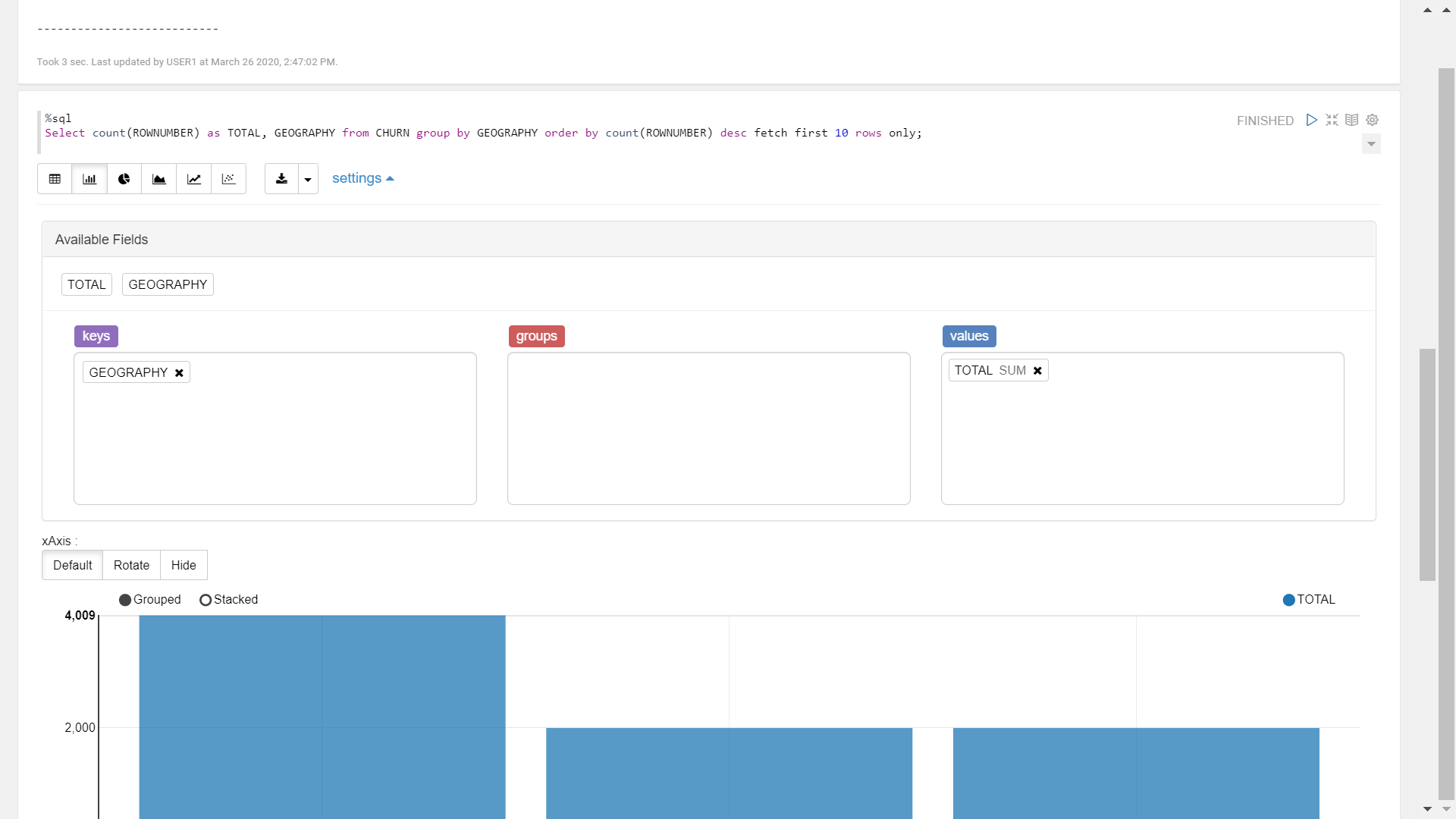


4. Посмотрим распределение записей по географиям

%**sql**

**SELECT** **COUNT**(ROWNUMBER) **AS** TOTAL, GEOGRAPHY **FROM** CHURN

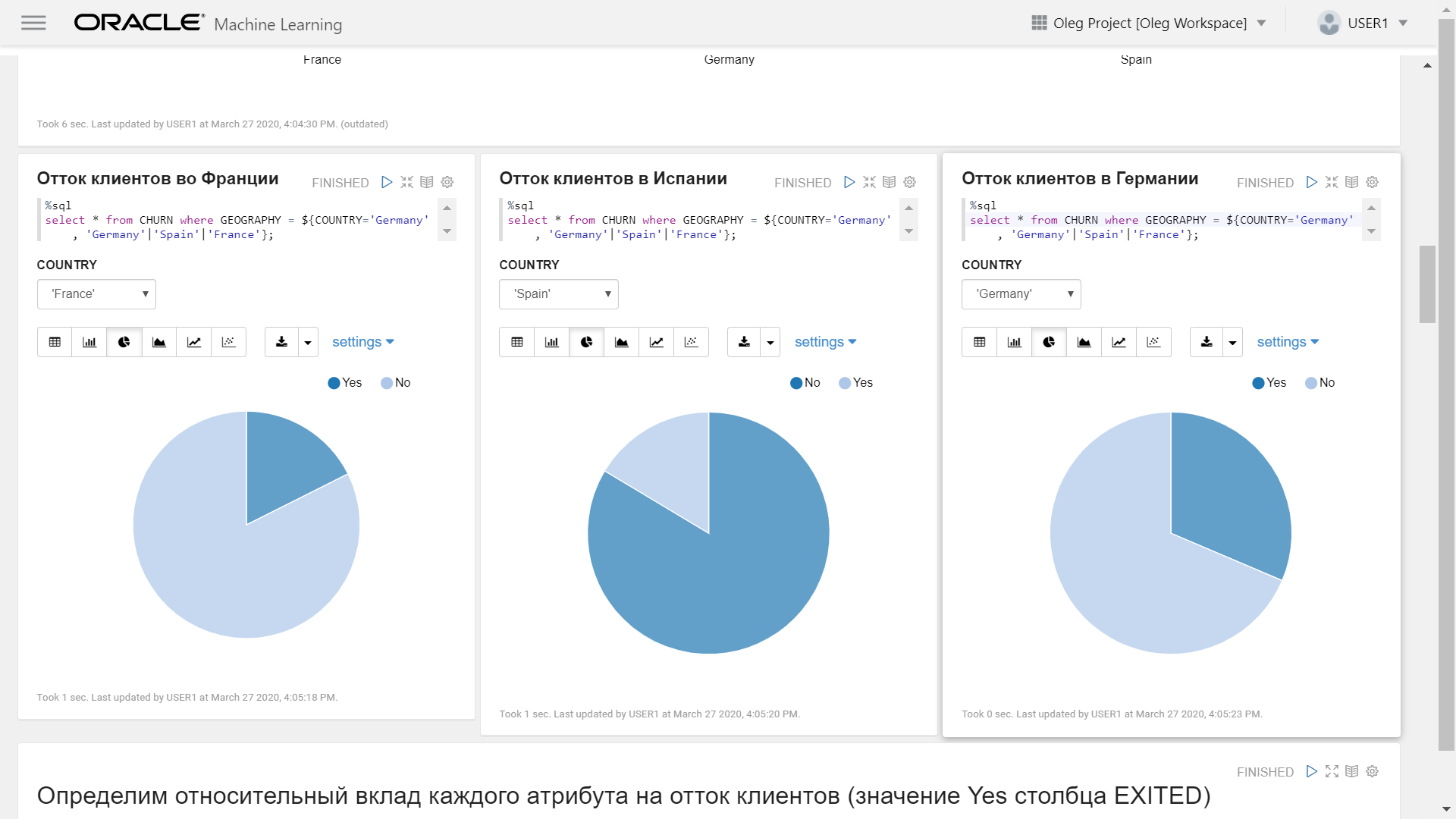
**GROUP BY** GEOGRAPHY **ORDER BY** **COUNT**(ROWNUMBER) **DESC**;



5. Изучаем отток клиентов в разных странах (3 параграфа в одной строке)

%**sql**

**SELECT** \* **FROM** CHURN **WHERE** GEOGRAPHY = ${COUNTRY='Germany', 'Germany'|'Spain'|'France'};



6. Создаем таблицу настроек

**%script**

--DROP TABLE Att\_Import\_Mode\_Settings PURGE;

**CREATE** **TABLE** Att\_Import\_Mode\_Settings (setting\_name **VARCHAR2**(30), setting\_value **VARCHAR2**(30));

7. Настраиваем и рассчитываем модель (алгоритм Minimum Description Length)

**%script**

**BEGIN**

--DBMS\_DATA\_MINING.DROP\_MODEL('CHURN\_\_AI');

**DBMS\_DATA\_MINING.CREATE\_MODEL**(

model\_name => 'CHURN\_\_AI',

mining\_function => DBMS\_DATA\_MINING.ATTRIBUTE\_IMPORTANCE,

data\_table\_name => 'CHURN',

case\_id\_column\_name => 'ROWNUMBER',

target\_column\_name => 'EXITED',

settings\_table\_name => 'Att\_Import\_Mode\_Settings');

**END**;

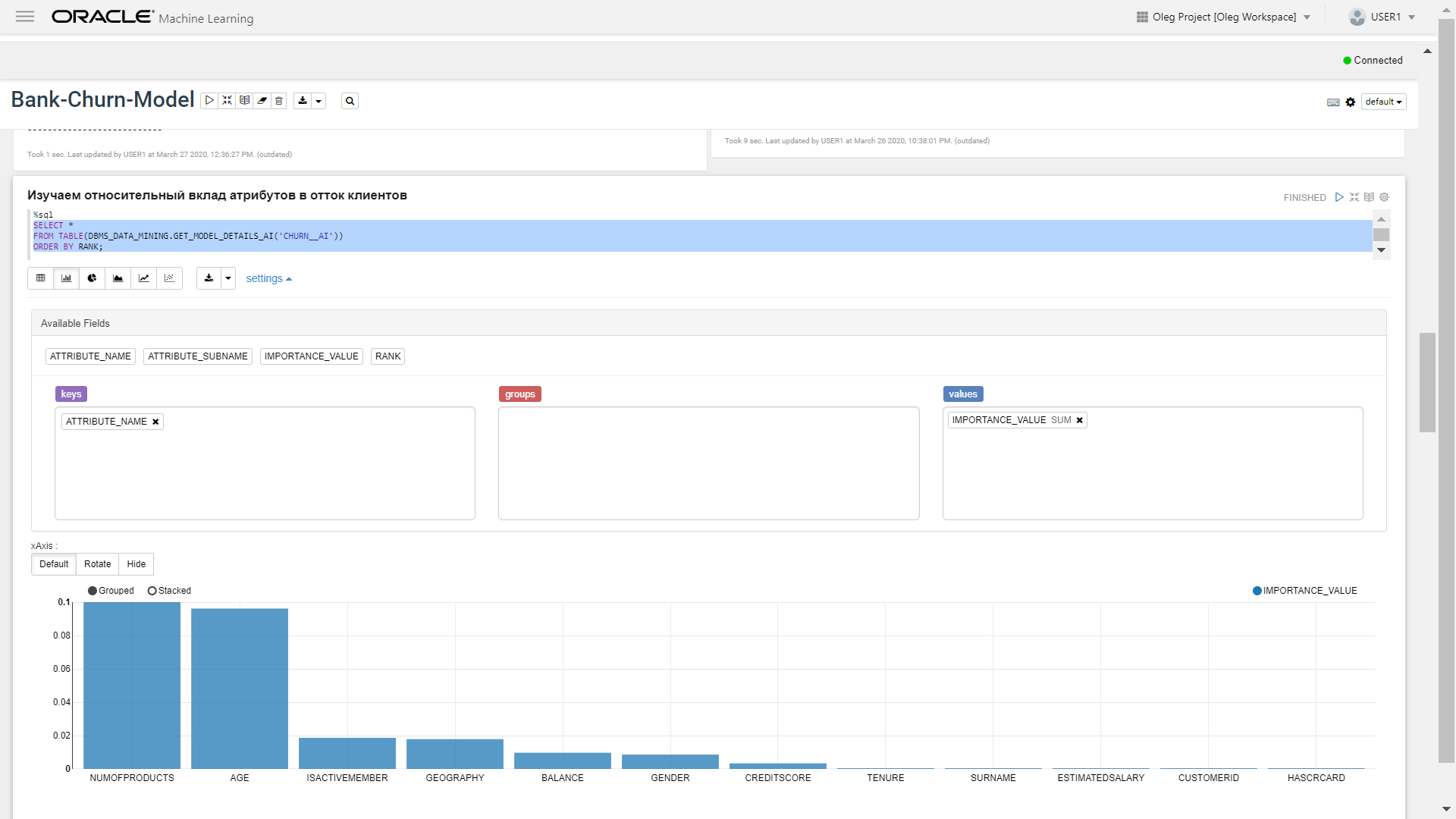
/

8. Изучаем относительный вклад атрибутов в отток клиентов

%**sql**

**SELECT** \* **FROM** **TABLE**(DBMS\_DATA\_MINING.GET\_MODEL\_DETAILS\_AI('CHURN\_\_AI'))

**ORDER** **BY** RANK;



9. Создадим набор для обучения и тестирования

%**script**

-- используем sql операторы SAMPLE и SEED для рандомизированной выборки 60% записей

**CREATE** **OR** **REPLACE** **VIEW** CHURN\_TRAIN\_DATA **AS** **SELECT** \* **FROM** CHURN **SAMPLE** (60) SEED (1);

-- используем sql оператор MINUS для отображения оставшихся записей в CHURN\_TRAIN\_DATA

**CREATE** **OR** **REPLACE** **VIEW** CHURN\_TEST\_DATA **AS** **SELECT** \* **FROM** CHURN **MINUS** **SELECT** \* **FROM** CHURN\_TRAIN\_DATA;

10. Обучим модель для прогноза оттока клиентов (алгоритм Random Forest)

**%script**

-- если вы запускаете скрипт повторно, убедитесь, что старые модели и таблицы удалены

--DROP TABLE CHURN\_build\_settings PURGE;

--CALL DBMS\_DATA\_MINING.DROP\_MODEL('CHURN\_CLASS\_MODEL');

--DROP TABLE CHURN\_APPLY\_RESULT PURGE;

-- создим таблицу настроек и укажем два параметра:

-- ALGO\_NAME - алгоритм, который будет использоваться для обучения модели

-- PREP\_AUTO - автоматическая подготовка данных (технология Oracle ADP)

**CREATE** **TABLE** CHURN\_build\_settings (setting\_name **VARCHAR2**(30), setting\_value **VARCHAR2**(4000));

**INSERT** **INTO** CHURN\_build\_settings (setting\_name, setting\_value)

**VALUES** ('ALGO\_NAME', 'ALGO\_RANDOM\_FOREST');

**INSERT** **INTO** CHURN\_build\_settings (setting\_name, setting\_value)

**VALUES** ('PREP\_AUTO', 'ON');

-- создадим и обучим модель также как и при расчете значимых атрибутов выше

**CALL** DBMS\_DATA\_MINING.CREATE\_MODEL('CHURN\_CLASS\_MODEL', 'CLASSIFICATION', 'CHURN\_TRAIN\_DATA', 'ROWNUMBER','EXITED', 'CHURN\_build\_settings');

-- применим обученную модель к тестовому набору данных для оценки ее качества

**CALL** DBMS\_DATA\_MINING.APPLY('CHURN\_CLASS\_MODEL','CHURN\_TEST\_DATA','ROWNUMBER','CHURN\_APPLY\_RESULT');

11. Рассчитаем матрицу ошибок (Confusion matrix)

**%script**

**DECLARE**

v\_accuracy **NUMBER**;

**BEGIN**

--EXECUTE IMMEDIATE 'DROP TABLE CHURN\_CM\_TABLE';

**DBMS\_DATA\_MINING.COMPUTE\_CONFUSION\_MATRIX**(v\_accuracy, 'CHURN\_APPLY\_RESULT',

'CHURN\_TEST\_DATA', 'ROWNUMBER','EXITED','CHURN\_CM\_TABLE','PREDICTION','PROBABILITY');

**DBMS\_OUTPUT**.**PUT\_LINE**('\*\*\*\* ТОЧНОСТЬ МОДЕЛИ \*\*\*\*: ' || **ROUND**(v\_accuracy,4));

**END**;

/

12. Отобразим матрицу ошибок (Confusion matrix)

%**sql**

**SELECT** \* **FROM** CHURN\_CM\_TABLE

13. Рассчитаем ROC-кривую

**%script**

**DECLARE**

v\_area\_under\_curve **NUMBER**;

**BEGIN**

--EXECUTE IMMEDIATE 'DROP TABLE CHURN\_ROC\_TABLE';

**DBMS\_DATA\_MINING.COMPUTE\_ROC**(v\_area\_under\_curve, 'CHURN\_APPLY\_RESULT','CHURN\_TEST\_DATA',

'ROWNUMBER','EXITED','CHURN\_ROC\_TABLE','Yes','PREDICTION','PROBABILITY');

**DBMS\_OUTPUT**.**PUT\_LINE**('\*\*\* ПЛОЩАДЬ ПОД ROC КРИВОЙ \*\*\*:' || ROUND(v\_area\_under\_curve,4));

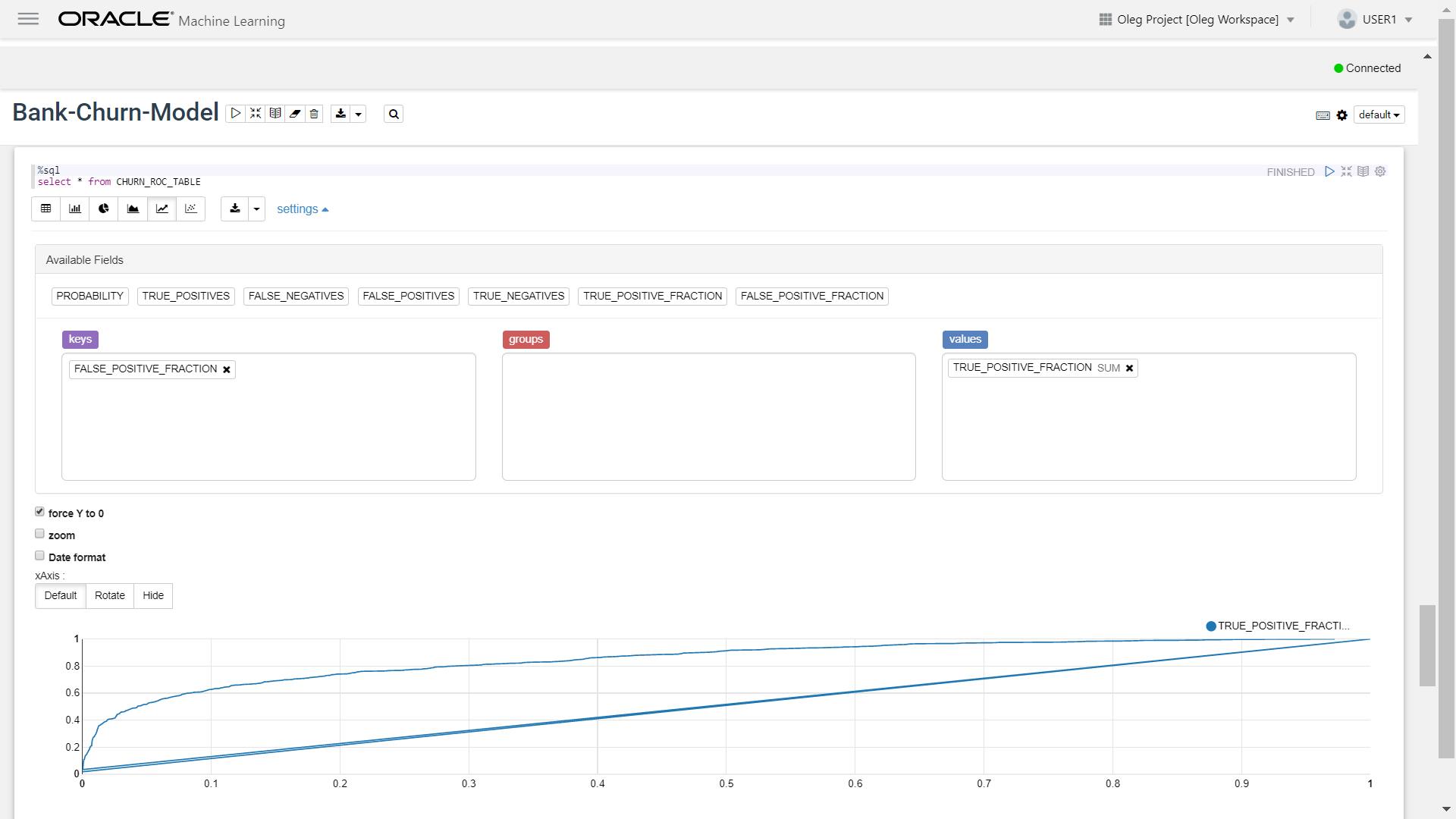
**END;**

      /

14. Отобразим ROC-кривую

%**sql**

**SELECT** \* **FROM** CHURN\_ROC\_TABLE



15. Изучим какие значения расчитала наша модель для тестового набора и сравним с исходными

%**sql**

**SELECT** **A**.PREDICTION, B.EXITED **FROM** CHURN\_APPLY\_RESULT **A**, CHURN\_TEST\_DATA B

**WHERE** PREDICTION = ${PREDICTION='Yes','Yes'|'No'} **AND a**.ROWNUMBER = b.ROWNUMBER;

16. Применим обученную модель для новых данных

**%script**

-- Drop table CHURNPREDICTION PURGE;

-- Применяем модель на лету и загружаем результаты в таблицу ChurnPredictons

**CREATE TABLE** CHURNPREDICTION **AS**

**SELECT** ROWNUMBER, CUSTOMERID, SURNAME, CREDITSCORE, GEOGRAPHY, GENDER, BALANCE, NUMOFPRODUCTS,

HASCRCARD, prediction, probability

**FROM** (**SELECT** ROWNUMBER, CUSTOMERID, SURNAME, CREDITSCORE, GEOGRAPHY, GENDER, BALANCE,

NUMOFPRODUCTS, HASCRCARD, PREDICTION\_SET(CHURN\_CLASS\_MODEL **USING** \*) pset

**FROM** CHURN) T, **TABLE**(pset) **S**

**ORDER** **BY** prediction, probability **DESC**;