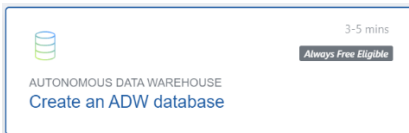


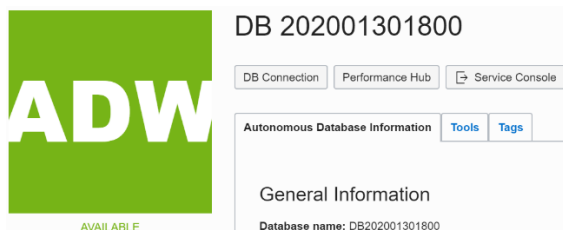
ПРОГНОЗИРУЕМ ОТТОК КЛИЕНТОВ С ПОМОЩЬЮ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ В БД ORACLE (OML4SQL)

Подготовка

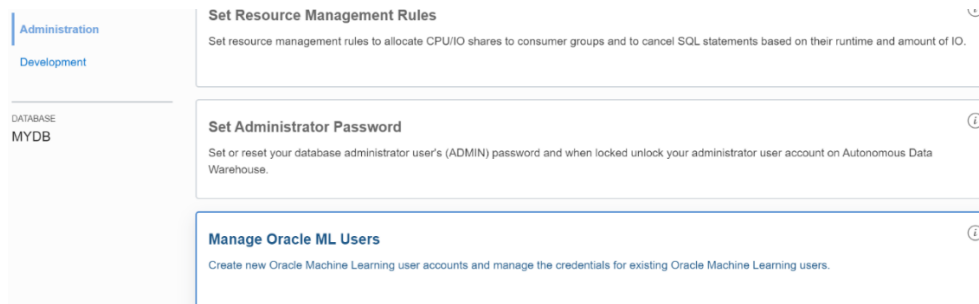
1. Зайдите в Oracle Cloud с вашей учетной записью.
2. Создайте ADW БД с параметрами по умолчанию.



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.

Create User


* Username

First Name

Last Name

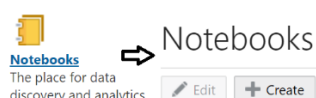
* Email Address

☒ Generate password and email account details to user. User will be required to reset the password on first sign in.

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;
```

ORACLE Machine Learning

Oleg Project [Oleg Workspace] USER1

Connected

Bank-Churn-Model

Took 3 sec. Last updated by USER1 at March 26 2020, 2:47:02 PM. (outdated)

Отображаем загруженные данные

FINISHED

%sql
select * from CHURN

settings

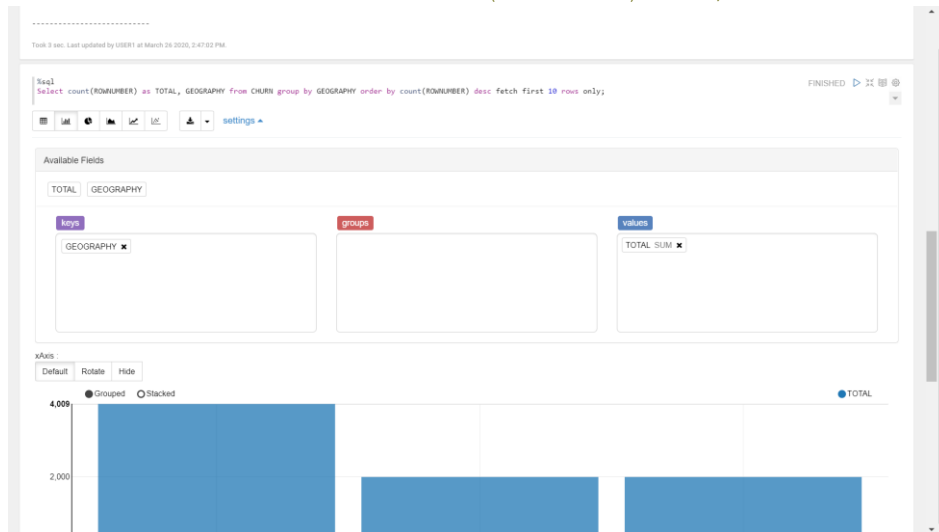
ROWNUMBER	CUSTOMERID	SURNAME	CREDITSCORE	GEOGRAPHY	GENDER	AGE	TENURE
1	15634602	Hargrave	619	France	Female	42	2
2	15647311	Hill	608	Spain	Female	41	1
3	15619304	Onio	502	France	Female	42	8
4	15701354	Boni	699	France	Female	39	1
5	15737888	Mitchell	850	Spain	Female	43	2
6	15574012	Chu	645	Spain	Male	44	8
7	15592531	Bartlett	822	France	Male	50	7
8	15656148	Obinna	376	Germany	Female	29	4

Took 6 sec. Last updated by USER1 at March 27 2020, 6:29:51 PM. (outdated)

FINISHED

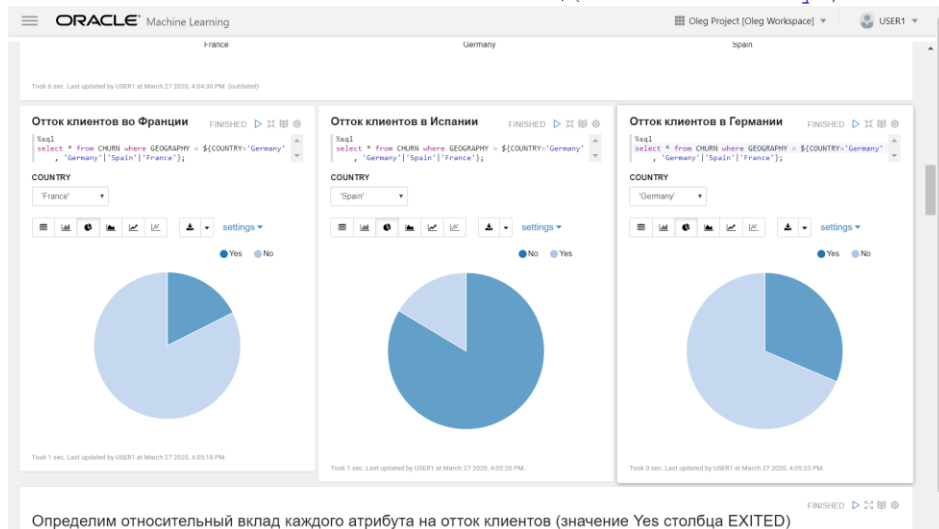
4. Посмотрим распределение записей по географиям

```
%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));
```

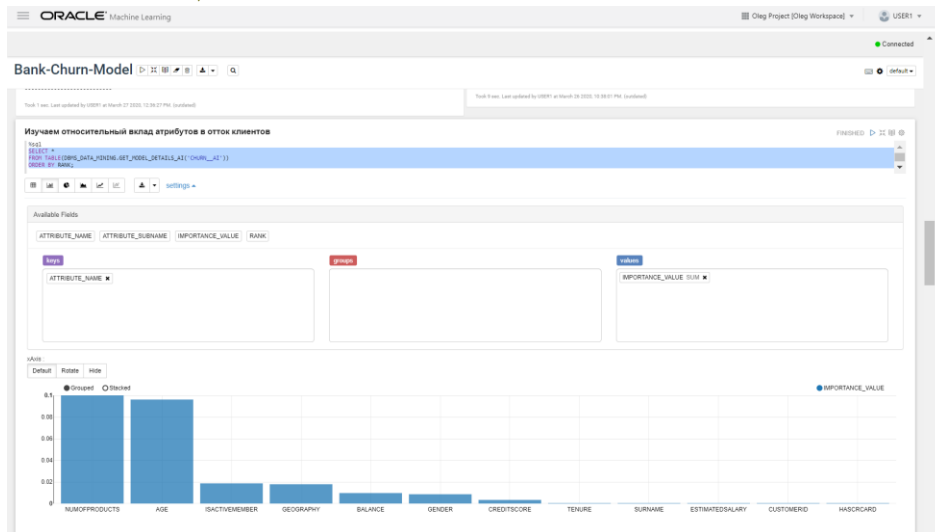
7. Настраиваем и рассчитываем модель (алгоритм Minimum Description Length)

```
%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;
```

```
-- создадим 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)

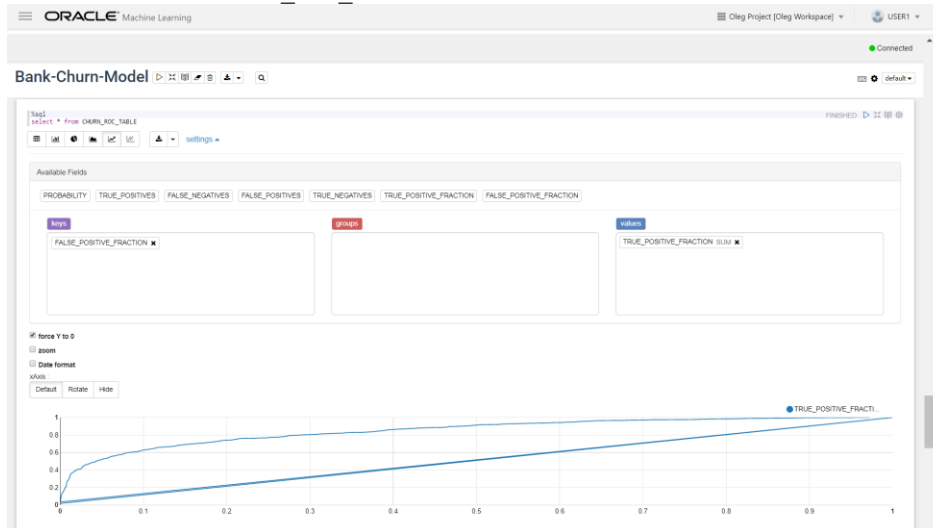
```
%sql
SELECT * FROM CHURN_CM_TABLE
```

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 рассчитала 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 ChurnPredictions
CREATE TABLE CHURNPREDICTION AS
SELECT ROWNUMBER, CUSTOMERID, SURNAME, CREDITSCORE, GEOGRAPHY, GENDER, BALANCE, NUMOFFPRODUCTS,
    HASCRCARD, prediction, probability
FROM (SELECT ROWNUMBER, CUSTOMERID, SURNAME, CREDITSCORE, GEOGRAPHY, GENDER, BALANCE,
    NUMOFFPRODUCTS, HASCRCARD, PREDICTION_SET(CHURN_CLASS_MODEL USING *) pset
    FROM CHURN) T, TABLE(pset) S
ORDER BY prediction, probability DESC;
```