Assessment Submission Cover Sheet

This Assessment Cover Sheet **must** be included on all Assessment submissions.

Assignment Title	Working with Data Assignment 2
Module	DTA9910: 2021-2022
Student Name	Robert O’Sullivan
Student Number	C08345457
Programme	TU060/DS
Part-Time/Full-Time	Part Time
Year of Study (First Year, Second Year, etc)	First Semester

Late Submissions: Assessment submitted after the deadline will have a late penalty applied.

Academic Integrity for assessment in TU Dublin Programmes

Each student is responsible for knowing and abiding by TU Dublin Academic Regulations and Policies. Any student in breach of these regulation/policies will be subject to action in accordance with the University’s procedures for breaches of assessment regulations. Please refer to the General Assessment Regulations at

- https://tudublin.libguides.com/c.php?g=674049&p=4794713
- https://www.tudublinsu.ie/advice/exams/breachesofregulations/

All students are expected to complete their courses/programmes in compliance with University regulations. No student shall engage in any activity that involves attempting to receive a grade by means other than honest effort, for example:

1. No student shall complete, in part or in total, any examination or assessment for another person.
2. No student shall knowingly allow any examination or assessment to be completed, in part or in total, for themselves by another person.
3. No student shall plagiarise or copy the work of another and submit it as their own work.
4. No student shall falsify any data. Falsification is the invention of data, its alteration, its copying from any other source, or otherwise obtaining it by unfair means, or inventing quotations and/or references.
5. No student shall use aids or devices excluded by the lecturer in undertaking course work or assessments/ examinations.
6. No student shall knowingly procure, provide, or accept any materials that contain questions or answers to any examination or assessment to be given at a subsequent time.
7. No student shall provide their assignments, in part or in total, to any other student in current or future classes of this module/ programme unless authorised to do so by the lecturer.
8. No student shall submit substantially the same material in more than one module/programme without prior authorization.
9. No student shall alter graded assignments or examinations and then resubmit them for regrading, unless specifically authorised to do so by the lecturer.
10. All programming code and documentation, unless correctly referenced, submitted for assessment or existing in the student’s computer accounts must be the students’ original work or material specifically authorized by the lecturer.
11. Collaborating with other students to develop, complete or correct course work is limited to activities explicitly authorized by the lecturer.
12. For all group assignments, each member of the group is responsible for the academic integrity of the entire submission. Consequently, all group members must satisfy themselves that all elements of their submission adhere to the academic integrity statement points above.

By submitting coursework, either physically or electronically, you are confirming that it is your own work (or, in the case of a group submission, that it is the result of joint work undertaken by members of the group that you represent) and that you have read and understand the University’s Regulations and Policies covering Academic Integrity (see General Assessment Regulations).

Coursework may be submitted to an electronic detection system in order to help ascertain if any plagiarised material is present. If you have queries about what constitutes plagiarism, please speak to your lecturer.

Student Signature	Robert O’Sullivan.
Date	04/01/2022

Table of Contents

Introduction 3
Section A: Data Warehouse Modelling 4
Oracle Data Importing Process 4
Business Topic 5
Grain Declared 6
Dimensions Discovered 6
Facts Declared 7
Interesting Queries 7
STAR Schema Design 8
Calls Dimension 8
Customer Dimension 8
Rates Dimension 8
Dates Dimension 8
Call Facts Table 8
Section B: Data Analysis and Queries Using Sql 9
1. What was the Total Monthly Calls made? 9
2. How much Revenue was generated from different calls? 9
3. What were Total Calls made per Month vs Customer Service Calls Made? 10
4. Who are the Top 10 Revenue Generating Customers who used Customer Service? 10
5. What is the Total Monthly Revenue Generated per Month? 10
6. What is the Most Popular Contract? 11
7. Who spends the Longest Time with Customer Service? 11
8. What is the monthly moving average of Revenue? 12
Section C: Machine Learning Using SQL 12
Case Table 13
Generating the Training & Testing Data Sets 13
Decision Tree 13
Building the decision tree model 13
Training the decision tree model 14
Testing the decision tree model 15
Neural Network 16
Building the neural network model 16
Training the neural network model 17
Testing the neural network model 17
Results Discussion & Conclusion 19
References 19

Introduction

A telecommunication company wanted customer service agents to have a better picture of the customers they were speaking to on the phone. Customer value, customer profile, customer behaviour patterns and revenue generating potential of different call plans were all important to stakeholders. This report comprised of three sections with the aim of creating a data warehouse, obtaining information through querying the data warehouse and developing a customer churn predictor using machine learning.

Firstly, Section A outlines the design a data warehouse for this business case. A fact table was created along with dimension tables that followed the STAR schema model. To create the STAR schema model, CSV files were imported and reviewed. The origin of these CSV files was assumed to be taken from the company's database or ERP system. The business process related to this data warehouse was specified, grain was declared which then helped dimensions to be discovered and facts to be described. Some interesting queries were also discussed. When both the fact table and dimension tables were created, data was transformed and loaded into the STAR schema model using several SQL statements in an Oracle Database Management System.

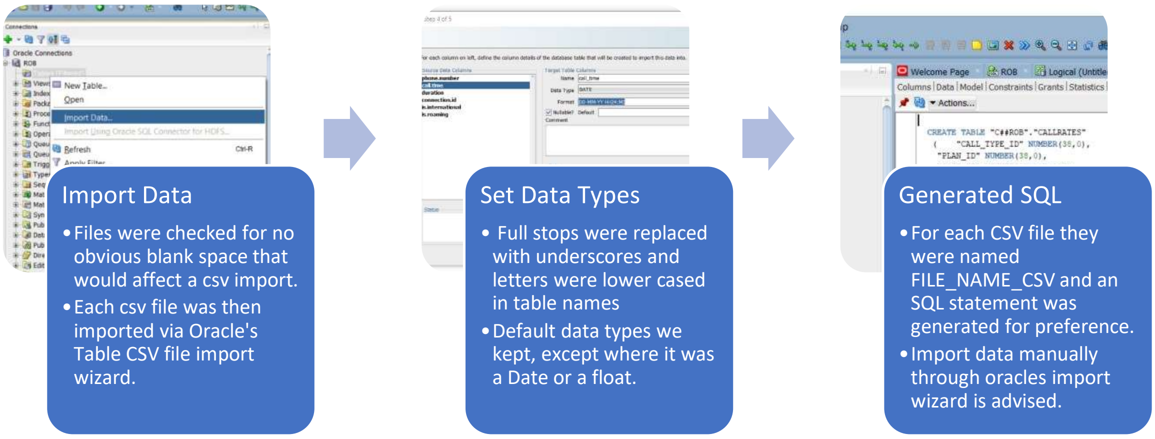
Secondly, in Section B interesting queries previously discussed in Section A were implemented using SQL statements. A screenshot was provided of each query's result along with a short description of how the SQL was created.

Finally, in Section C both a Decision Tree machine learning model and a Neural Network Model were created using the Fact Table from the data warehouse, data was divided into a training and testing set, models were trained and their predictability was tested. Performance of each model was then reviewed and predictability of both models was discussed.

Section A: Data Warehouse Modelling

Oracle Data Importing Process

The following screenshots and points below outline how csv data was imported into oracle via the oracle import wizard.



```
CREATE TABLE "C##ROB"."CALLRATES"
   (    "CALL_TYPE_ID" NUMBER(38,0),
        "PLAN_ID" NUMBER(38,0),
```

Import Data

- Files were checked for no obvious blank space that would affect a csv import.
- Each csv file was then imported via Oracle's Table CSV file import wizard.

Set Data Types

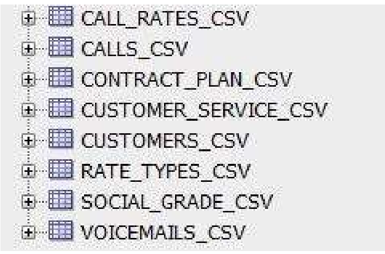
- Full stops were replaced with underscores and letters were lower cased in table names
- Default data types we kept, except where it was a Date or a float.

Generated SQL

- For each CSV file they were named FILE_NAME_CSV and an SQL statement was generated for preference.
- Import data manually through oracles import wizard is advised.

Figure 1 - importing csv files into Oracle.

When the import was successful the following CSV eight tables were created:



CALL_RATES_CSV
CALLS_CSV
CONTRACT_PLAN_CSV
CUSTOMER_SERVICE_CSV
CUSTOMERS_CSV
RATE_TYPES_CSV
SOCIAL_GRADE_CSV
VOICEMAILS_CSV

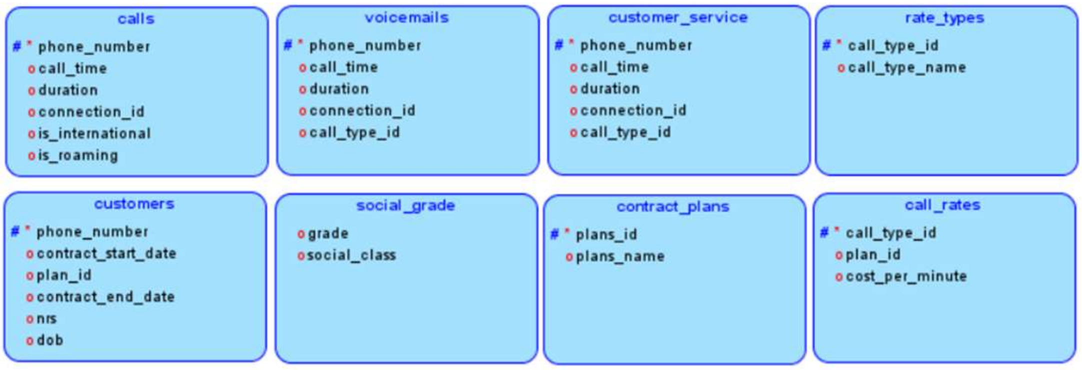
Figure 2 - Oracle generated tables from csv file import.

Data was later transformed (i.e data cleaned), physical structured and then loaded into a star schema model using an SQL file called 'etl.sql'.

The following is an overview of each table containing data:

- **Calls** table showed calls either international/roaming by phone numbers with a timestamp and duration. Connection id seemed to be unique for each call. Unlike voicemails and customer service calls do not show call types.
  - Determine peak and off-peak hours
  - Convert roaming and international into call types
  - Use connection id as primary key
- **Voicemails** table showed a call type, phone numbers with a timestamp and duration
  - Use connection id as primary key
- **Customer Service** table like voicemails table showed a call type, phone numbers, timestamp and duration.
- **Rate Types** table showed six different call types from customer service calls to peak/off-peak calls. Each one is identified by an id number.
  - Change table name to call_type
  - Change id column to call_type_id

  - Change name column to call_type_name
- **Customers** table showed the customer's phone number, date of birth, social status, their contract period, what plan they have. Phone number seems to be unique for each customer.
- **Social Grade** table showed six classes from non-working to upper middle class. Each class is uniquely identified by a letter or letter wit a number (e.g C1 = lower middle class).
  - Change grade to social_grade_id
  - Change social class to social_grade_name
- **Contract Plans** table showed three types of plans for customers. Each are uniquely identified.
  - Change id to plan_id and name to plan_name
- **Call Rates** table showed eighteen different prices for calls depending on what the call type was and what plan the customer had.



calls
* phone_number
o call_time
o duration
o connection_id
o is_international
o is_roaming

voicemails
* phone_number
o call_time
o duration
o connection_id
o call_type_id

customer_service
* phone_number
o call_time
o duration
o connection_id
o call_type_id

rate_types
* call_type_id
o call_type_name

customers
* phone_number
o contract_start_date
o plan_id
o contract_end_date
o nrs
o dob

social_grade
o grade
o social_class

contract_plans
* plans_id
o plans_name

call_rates
* call_type_id
o plan_id
o cost_per_minute

Figure 3 - Overview of attributes from data collected.

Business Topic

To determine the business process of our data warehouse, a diagram was constructed (see Figure 4). Data collected from the business formed eight tables describing customers making calls, checking voicemail or contacting customer service. The type of contract the customer had and the type of call determine the cost the customer had to pay. Date of birth and social status were also included as additional attributes to distinguish customers.

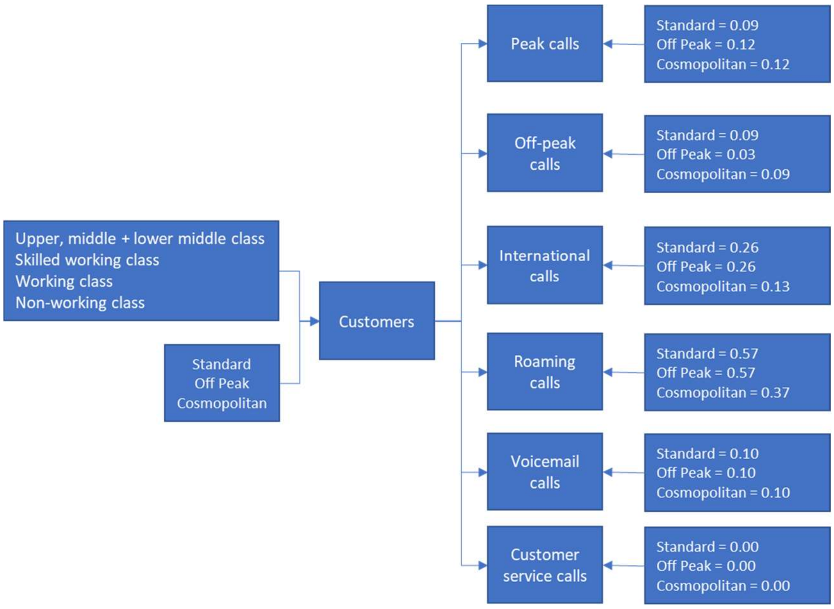


Figure 4 - Overview of business process

It was determined from a review of the data that the business process to be analysed was 'Customer making Calls' as this activity triggered the most of supporting data generated. Key performance metrics from calls could also be able to capture.

Feature requirements were able to be created as follows:

- We were able to define valuable customers as those who generate the most revenue for the company making a high number of calls and/or make expensive calls while making fewer customer service calls (which we assumed reduces operation costs).
- We were able to define build initial customer profiles from date of birth, contract plan and social status.
- We were able to define customer behaviours as call type by frequencies, call type by time of day, call type by duration.

Grain Declared

For granularity in the data warehouse, we defined what rows describe in our fact table. We wanted to use atomic data where possible such as call type and call rates. This allowed for data to be aggregated for a big picture overview of the process. It would also all analytical queries to be performed (See Section B: Data Analysis and Queries Using Sql).

The following were possible grains considered:

1. Quantity of customer service calls (customer service calls by customers); We wanted to better understand customer service calls but we would have lost revenue information if we just focused on customer service calls.
2. Revenue generated from contracts (Customer contract start/end date by total cost of calls made); This would have been too rough of a grain so was not chosen.
3. Monthly revenue generated from calls (customer calls by call rates). This would have been interesting to the business but too narrow of scope for our purposes.
4. Calls (calls per customer); This captured information about calls, contracts and customers by time.

After reviewing possible grains, our grain was defined as **Calls** per **Customer** per **Contract** per **Time Interval**.

Dimensions Discovered

Using our defined grain, the following questions were asked about the grain to derive dimensions for our fact table. The following dimensions were discovered "**Customers** make **Calls** at different **Date**/**Time** intervals costing different **Rates** on different **Contracts"**.

1. **DateTime** Dimension (When)
2. **Customer** Profile Dimension (Who)
3. **Call** Behaviour Dimension (What)
4. **Rates** Dimension (How)

An integer called 'cf_id' was autogenerated and used as the primary key to identify each row of the fact table.

Facts Declared

The following columns declared corresponded to facts from the grains of data which was imported from eight csv tables (see Oracle Data Importing Process):

1. **CF_id** (Primary Key)
2. **DateTime_id** (FK DateTime Dimension)
3. **Customer_id** (FK Customer Profile Dimension)
4. **Call_id** (FK Customer Behaviour Dimension)
5. **Call_rate_id** (FK Call Plan Revenue Dimension)
6. **Customer_Age** (taken from customer behaviour dimension); Non-additive snapshot
7. **Customer_Social_Status** (taken from customer profile dimension); Non-additive snapshot
8. **Customer_Contract_Plan** (taken from customer profile dimension); Semi-Additive snapshot
9. **Days_on_Plan** (contract end date minus dateTime); Additive Cumulative
10. **Revenue_Generated** (calculated using Call Plan Revenue Dimension); Additive Cumulative
11. **Time_Spent** with Customer Semi-Additive snapshot
12. **Customer_Active**; Customer with the company (end date null=1); semi-additive snapshot

Revenue_Generated is an additive fact because it can be summed up for all dimensions.

Interesting Queries

The following queries helped better understand customer value, their profile, behaviour and potential revenue from contract plans, in Section B: Data Analysis and Queries Using Sql the purpose, method and result of these queries are discussed further:

1. What was the Total Monthly Calls made?
2. How much Revenue was generated from different calls?
3. What were Total Calls made per Month vs Customer Service Calls Made?
4. Who are the Top 10 Revenue Generating Customers who used Customer Service?
5. What is the Total Monthly Revenue Generated per Month?
6. What is the Most Popular Contract?
7. Who spends the Longest Time with Customer Service?
8. What is the monthly moving average of Revenue?

STAR Schema Design

The following STAR schema was designed based on the business process and what was deemed important by stakeholders; Customer value, their profile, their behaviour patterns and contract plan revenue generating potential. Originally a snowflake design was generated in the first draft, however according to (Kimball & Ross, 1996) snowflaking may reduce disk space they discourage the practice as savings are usually insignificant when compared to an entire data warehouse.

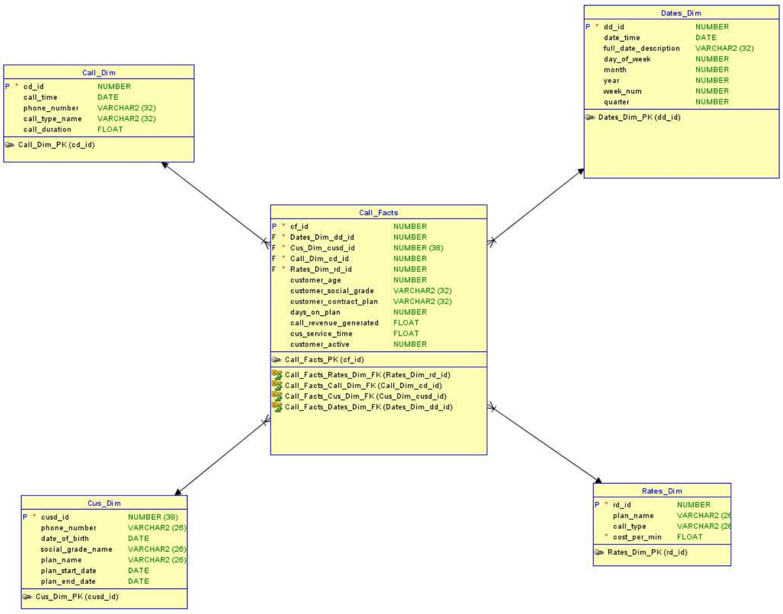


Figure 5 - STAR schema design for calls

Calls Dimension

Comprised of time, phone number, call type and duration. This dimension is used to help relate to customers behaviour and how much revenue customers generated for the company. This answers questions around the 'what' of our grain.

Customer Dimension

Comprised of the customer's date of birth (to calculate age), their social grade along with details about the contract plan they were on (or currently on). This answered questions we would have around the 'Who' of our grain.

Rates Dimension

Cost per minute of each call and each plan. The company can change this parameter to optimise revenue generation of each plan. This answered questions around the 'How' of our grain.

Dates Dimension

Allows information about monthly and quarterly calls or revenue generated. This answers questions around the 'when' of our grain,

Call Facts Table

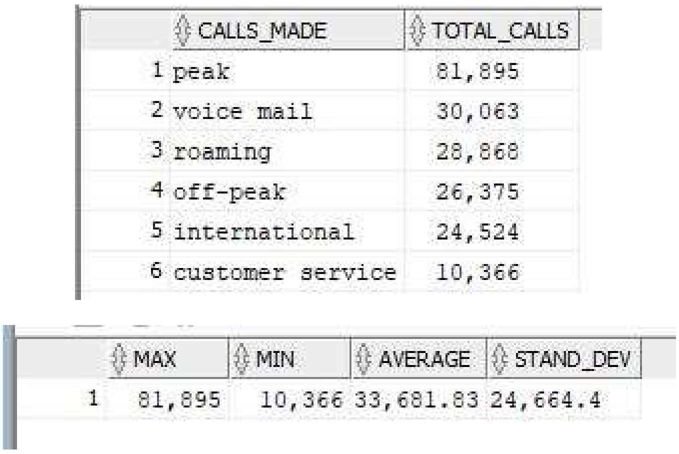
Several facts from this dimension were derived which focused on revenue generated and time spent with customer service. The assumption made here was that increasing time spent with customer service decreases potential revenue generated as customers call time is tied up on free call time activities.

Section B: Data Analysis and Queries Using Sql

Using both fact table and dimension tables created in the previous section an analysis was carried out on the data using SQL. The following queries were generated based on their suitability to what stakeholders deemed important to the business and what we identified in the previous section as interesting queries. A screenshot of each query's result will be provided along with a short description of how the SQL was created to implement each query.

1. What was the Total Monthly Calls made?

Purpose of the query below was to determine the scale of calls made, to identify the above average calls and to determine the number of calls made to customer service. A CALLS_VIEW table was created which fed into a CALLS_SUMMARY_VIEW where the customer id was counted and revenue summed. The query was grouped by CALL_TYPE_NAME and ordered by (descending) TOTAL_CALLS.



	CALLS_MADE	TOTAL_CALLS
1	peak	81,895
2	voice mail	30,063
3	roaming	28,868
4	off-peak	26,375
5	international	24,524
6	customer service	10,366

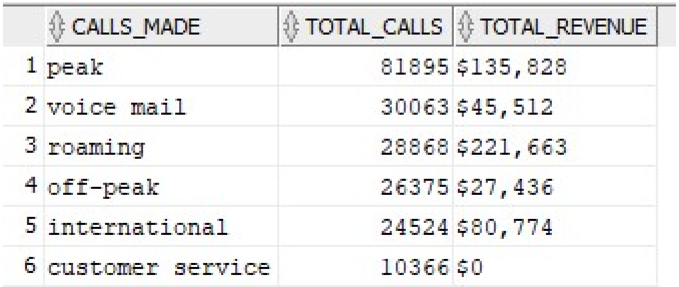
	MAX	MIN	AVERAGE	STAND_DEV
1	81,895	10,366	33,681.83	24,664.4

Figure 6 - total calls by type with stat descriptions

Results show 10,366 calls were made to customer service which were deemed lost opportunity. Peak calls were the most popular with 81,895 made (above average of 33,681.83).

2. How much Revenue was generated from different calls?

Purpose of this query was to quantify the value of each call made. Simple select statement was executed with revenue formatted to show dollar currency values.



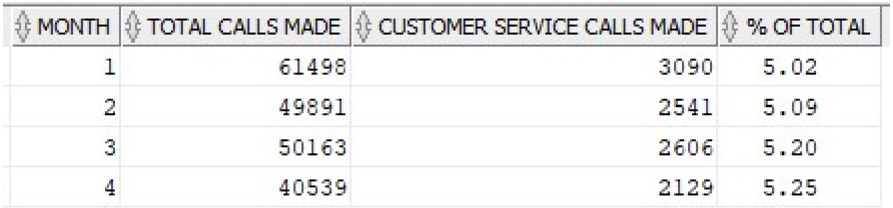
	CALLS_MADE	TOTAL_CALLS	TOTAL_REVENUE
1	peak	81895	$135,828
2	voice mail	30063	$45,512
3	roaming	28868	$221,663
4	off-peak	26375	$27,436
5	international	24524	$80,774
6	customer service	10366	$0

Figure 7 - revenue generated by calls

Results show although peak calls were the most popular, roaming calls generated the most revenue for the company at $221,663. There is an interesting use (30,063) and revenue generated ($45,512) from voicemails which could be an opportunity to the company.

3. What were Total Calls made per Month vs Customer Service Calls Made?

Purpose of query was to see if more calls to customer service were made in the last few months, an increase in calls could hinder potential revenue generation. Month was extracted from DATE_TIME all call types were summed with a percentage customer time calculated



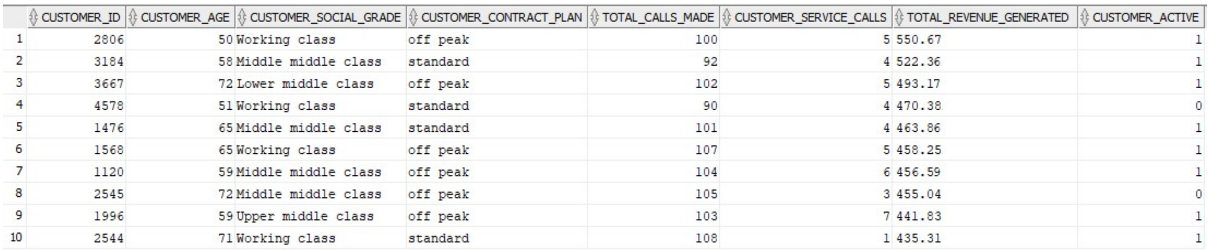
MONTH	TOTAL CALLS MADE	CUSTOMER SERVICE CALLS MADE	% OF TOTAL
1	61498	3090	5.02
2	49891	2541	5.09
3	50163	2606	5.20
4	40539	2129	5.25

Figure 8 - customer service calls as a percentage of total calls.

Results showed that customer service calls make up 5% of total calls per month with a slight increase between .05 and .1 of a percent increase.

4. Who are the Top 10 Revenue Generating Customers who used Customer Service?

Purpose of this query was to identify top customers; these should spend the least amount of time with customer service and instead be incentivised to make more calls. This time a CUSTOMER_VIEW was created with each customer and sum of each call type. A TOP_CUSTOMER_VIEW was created where the top 10 customers were fetched.



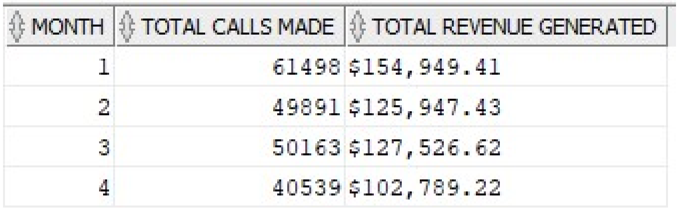
	CUSTOMER_ID	CUSTOMER_AGE	CUSTOMER_SOCIAL_GRADE	CUSTOMER_CONTRACT_PLAN	TOTAL_CALLS_MADE	CUSTOMER_SERVICE_CALLS	TOTAL_REVENUE_GENERATED	CUSTOMER_ACTIVE
1	2806	50	Working class	off peak	100	5	550.67	1
2	3184	58	Middle middle class	standard	92	4	522.36	1
3	3667	72	Lower middle class	off peak	102	5	493.17	1
4	4578	51	Working class	standard	90	4	470.38	0
5	1476	65	Middle middle class	standard	101	4	463.86	1
6	1568	65	Working class	off peak	107	5	458.25	1
7	1120	59	Middle middle class	off peak	104	6	456.59	1
8	2545	72	Middle middle class	off peak	105	3	455.04	0
9	1996	59	Upper middle class	off peak	103	7	441.83	1
10	2544	71	Working class	standard	108	1	435.31	1

Figure 9 - customers, revenue and number of calls to customer service.

Results show older customers between the ages of 50 to 72 of a working middle class background generated the most calls. Two are no longer active with the company and both standard and off-peak contracts seem to be popular with them.

5. What is the Total Monthly Revenue Generated per Month?

The purpose of this query was to see how well the company performed over the last few months.



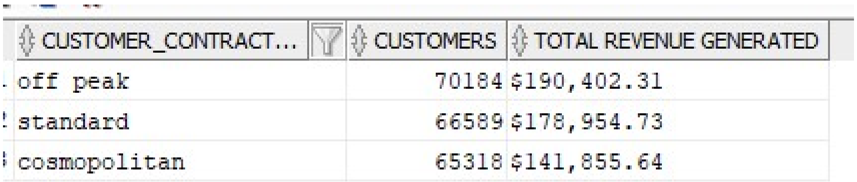
MONTH	TOTAL CALLS MADE	TOTAL REVENUE GENERATED
1	61498	$154,949.41
2	49891	$125,947.43
3	50163	$127,526.62
4	40539	$102,789.22

Figure 10 - revenue generated by calls made

Results showed that calls decreased over the last four months with revenue also decreasing. This should be a concern to the company with further investigation needed as to the why.

6. What is the Most Popular Contract?

Purpose of this query was to understand how revenue is generated by contracts.



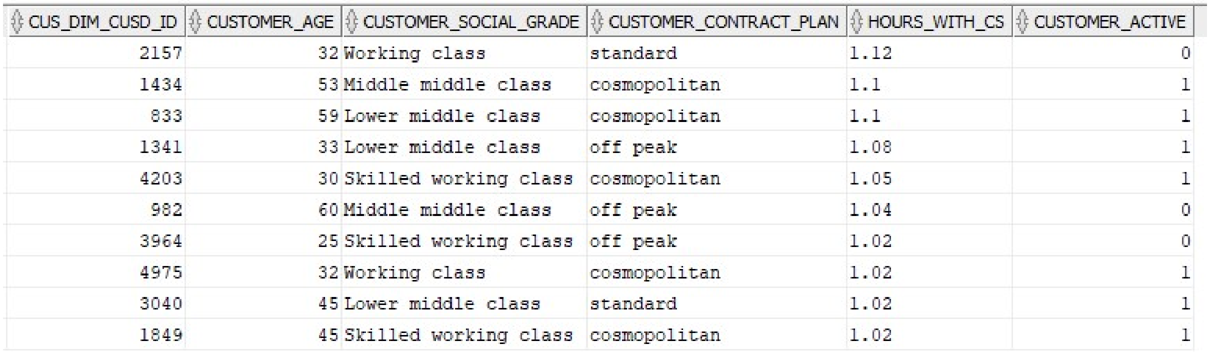
CUSTOMER_CONTRACT...	CUSTOMERS	TOTAL REVENUE GENERATED
off peak	70184	$190,402.31
standard	66589	$178,954.73
cosmopolitan	65318	$141,855.64

Figure 11 - revenue by contract type

Result show that most customers are on 'off peak' contracts with are generating $2.71 per customer.

7. Who spends the Longest Time with Customer Service?

Purpose of query was to determine which customers were spending the most time with customer service. The idea is to understand which contracts are not working and which age groups should be focused on. A POTENTIAL_REVENUE_VIEW was created where the OVER function is used with the average total revenue. This created a window from which each average was calculated using the proceeding values.



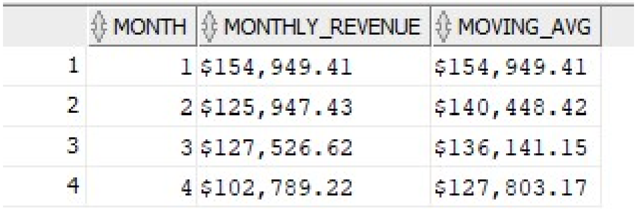
CUS_DIM_CUSD_ID	CUSTOMER_AGE	CUSTOMER_SOCIAL_GRADE	CUSTOMER_CONTRACT_PLAN	HOURS_WITH_CS	CUSTOMER_ACTIVE
2157	32	Working class	standard	1.12	0
1434	53	Middle middle class	cosmopolitan	1.1	1
833	59	Lower middle class	cosmopolitan	1.1	1
1341	33	Lower middle class	off peak	1.08	1
4203	30	Skilled working class	cosmopolitan	1.05	1
982	60	Middle middle class	off peak	1.04	0
3964	25	Skilled working class	off peak	1.02	0
4975	32	Working class	cosmopolitan	1.02	1
3040	45	Lower middle class	standard	1.02	1
1849	45	Skilled working class	cosmopolitan	1.02	1

Figure 12 - customers with customer service the longest.

Results show that customers who are on cosmopolitan contracts seem to spend a long time with customer service. These contracts as we found Figure 11 generate the lease $/customer. There is an opportunity to move customers from cosmopolitan contracts to 'off peak'.

8. What is the monthly moving average of Revenue?

Purpose of this query was to look at the moving average to filter out potential noise seen in short term revenue fluctuations as we previously saw in Figure 10.



	MONTH	MONTHLY_REVENUE	MOVING_AVG
1	1	$154,949.41	$154,949.41
2	2	$125,947.43	$140,448.42
3	3	$127,526.62	$136,141.15
4	4	$102,789.22	$127,803.17

Figure 13 - Monthly moving average of revenue.

Results show, in four months $511,212.68 were generated. From results of query 6 we know this is made up of customers who are on peak (37%), standard (35%) and cosmopolitan (28%) contracts. The moving average is indicating a downward trend.

If we assume a 6.12% decrease as we see from March to April, we could see $119,981.60 ($127.8k less 6.12%) in next month's revenue made up of:

- $44,393.19 potentially from customers on peak contracts,
- $41,993.56 potentially from customers on standard contracts,
- $33,594.85 potentially from customers on cosmopolitan contracts,

Section C: Machine Learning Using SQL

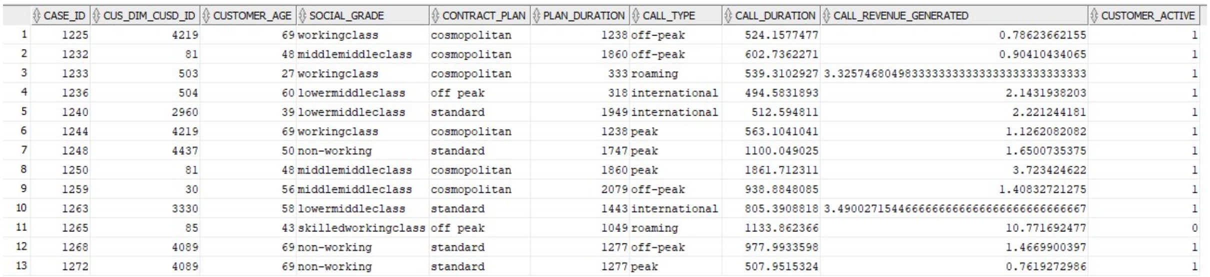
In this section two machine learning models were created using Oracle's in-database machine learning features. A churn model was produced which predicted customers who were likely to leave. The following sections explain the parameters values selected and the accuracy of both models.

During testing, predicting of any given customer based on any given month proved difficult. This report was unable to predict customer churn by month. The following will outline a Decision Tree Model and a Neural Network Model that predicts customer churn without including the date dimension. Duration of contract (contract start date vs end date) was instead used. Where end date was null, the date of this report was used as a replacement. Fact table generated in Section A: Data Warehouse Modelling was used with these models. A training and test set was prepared with 80% set aside for training and 20% for testing.

The accuracy of the two separate models was then evaluated. PL/SQL was attempted to combine the accuracy measures from the two models and to present them to the user. In the end a union function was used and the result can be found in Figure 28.

Case Table

Several categorical and numeric variables were selected from the fact table and prepared in case table below. Date was not included. No encoding or dummy variables were used as Oracle will adapt to the labelled data provided.



	CASE_ID	CUS_DIM_CUSD_ID	CUSTOMER_AGE	SOCIAL_GRADE	CONTRACT_PLAN	PLAN_DURATION	CALL_TYPE	CALL_DURATION	CALL_REVENUE_GENERATED	CUSTOMER_ACTIVE
1	1225	4219	69	workingclass	cosmopolitan	1238	off-peak	524.1577477	0.78623662155	1
2	1232	81	48	middlemiddleclass	cosmopolitan	1860	off-peak	602.7362271	0.90410434065	1
3	1233	503	27	workingclass	cosmopolitan	333	roaming	539.3102927	3.3257468049833333333333333333333333333	1
4	1236	504	60	lowermiddleclass	off peak	318	international	494.5831893	2.1431938203	1
5	1240	2960	39	lowermiddleclass	standard	1949	international	512.594811	2.221244181	1
6	1244	4219	69	workingclass	cosmopolitan	1238	peak	563.1041041	1.1262082082	1
7	1248	4437	50	non-working	standard	1747	peak	1100.049025	1.6500735375	1
8	1250	81	48	middlemiddleclass	cosmopolitan	1860	peak	1861.712311	3.723424622	1
9	1259	30	56	middlemiddleclass	cosmopolitan	2079	off-peak	938.8848085	1.40832721275	1
10	1263	3330	58	lowermiddleclass	standard	1443	international	805.3908818	3.4900271544666666666666666666666666667	1
11	1265	85	43	skilledworkingclass	off peak	1049	roaming	1133.862366	10.771692477	0
12	1268	4089	69	non-working	standard	1277	off-peak	977.9933598	1.4669900397	1
13	1272	4089	69	non-working	standard	1277	peak	507.9515324	0.7619272986	1

Figure 14 – case table prepared for oracle machine learning.

Generating the Training & Testing Data Sets

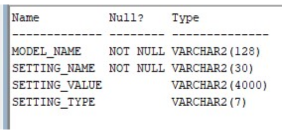
202,091 rows of data existed in the case table. These were divided up into 161,673 (80%) for training data and 40,418 (20%) testing data.

Decision Tree

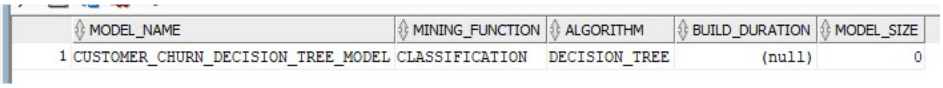
A decision tree was used as the first model. Default parameters were used. Where stems of the decision tree went to a depth of 7.

Building the decision tree model

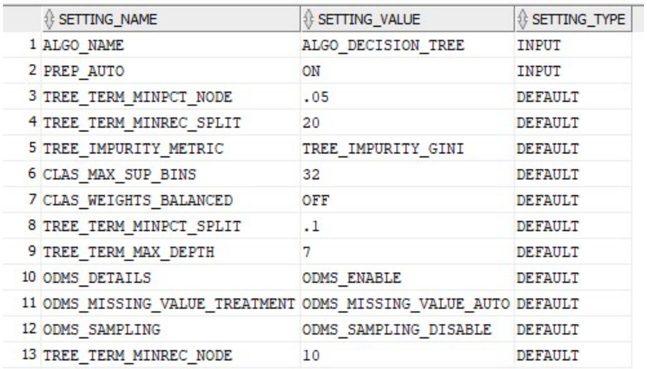
First the model was created with default hyperparameters used



```
Name          Null?    Type
------------- -------- --------------
MODEL_NAME    NOT NULL VARCHAR2(128)
SETTING_NAME  NOT NULL VARCHAR2(30)
SETTING_VALUE          VARCHAR2(4000)
SETTING_TYPE           VARCHAR2(7)
```



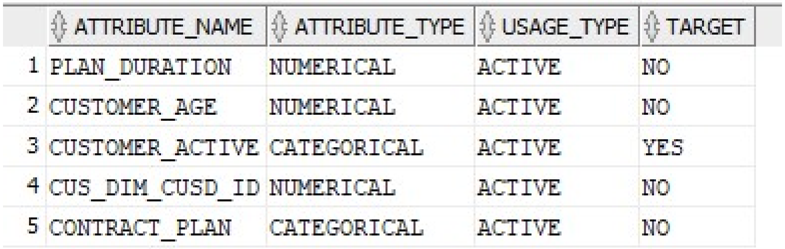
	MODEL_NAME	MINING_FUNCTION	ALGORITHM	BUILD_DURATION	MODEL_SIZE
1	CUSTOMER_CHURN_DECISION_TREE_MODEL	CLASSIFICATION	DECISION_TREE	(null)	0



	SETTING_NAME	SETTING_VALUE	SETTING_TYPE
1	ALGO_NAME	ALGO_DECISION_TREE	INPUT
2	PREP_AUTO	ON	INPUT
3	TREE_TERM_MINPCT_NODE	.05	DEFAULT
4	TREE_TERM_MINREC_SPLIT	20	DEFAULT
5	TREE_IMPURITY_METRIC	TREE_IMPURITY_GINI	DEFAULT
6	CLAS_MAX_SUP_BINS	32	DEFAULT
7	CLAS_WEIGHTS_BALANCED	OFF	DEFAULT
8	TREE_TERM_MINPCT_SPLIT	.1	DEFAULT
9	TREE_TERM_MAX_DEPTH	7	DEFAULT
10	ODMS_DETAILS	ODMS_ENABLE	DEFAULT
11	ODMS_MISSING_VALUE_TREATMENT	ODMS_MISSING_VALUE_AUTO	DEFAULT
12	ODMS_SAMPLING	ODMS_SAMPLING_DISABLE	DEFAULT
13	TREE_TERM_MINREC_NODE	10	DEFAULT

Figure 15 - Decision Tree hyper parameters used.

Plan duration, customer age, contract plan and customer ID was used as input parameters with customer active as the target attribute. Customer ID should have been manually removed but Oracle was left to manage it.

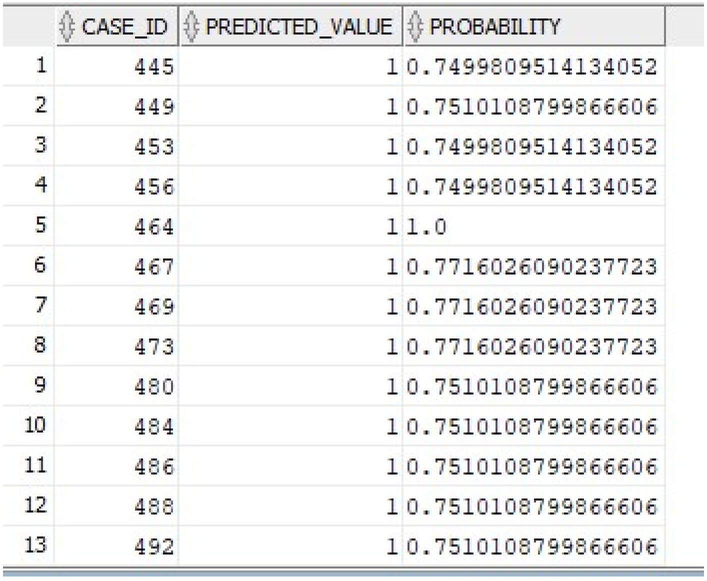


	ATTRIBUTE_NAME	ATTRIBUTE_TYPE	USAGE_TYPE	TARGET
1	PLAN_DURATION	NUMERICAL	ACTIVE	NO
2	CUSTOMER_AGE	NUMERICAL	ACTIVE	NO
3	CUSTOMER_ACTIVE	CATEGORICAL	ACTIVE	YES
4	CUS_DIM_CUSD_ID	NUMERICAL	ACTIVE	NO
5	CONTRACT_PLAN	CATEGORICAL	ACTIVE	NO

Figure 16 - attributes used in DT

Training the decision tree model

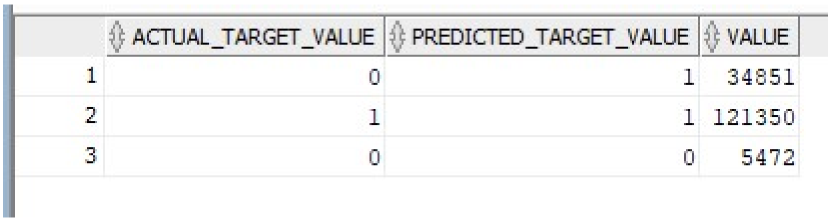
The decision tree model was trained making predictions on each case in the training data set. 75% to 100% probabilities were computed for predicted values of 1.



	CASE_ID	PREDICTED_VALUE	PROBABILITY
1	445	1	0.7499809514134052
2	449	1	0.7510108799866606
3	453	1	0.7499809514134052
4	456	1	0.7499809514134052
5	464	1	1.0
6	467	1	0.7716026090237723
7	469	1	0.7716026090237723
8	473	1	0.7716026090237723
9	480	1	0.7510108799866606
10	484	1	0.7510108799866606
11	486	1	0.7510108799866606
12	488	1	0.7510108799866606
13	492	1	0.7510108799866606

Figure 17 DT model training

A Confusion matrix was generated to determine the accuracy of the model on the training data. It returned a 78% predictability.

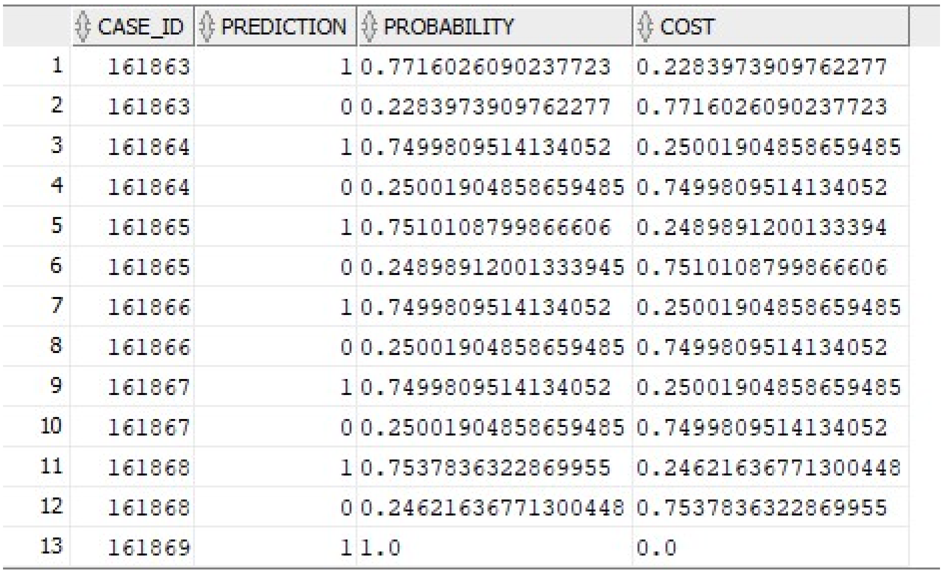


	ACTUAL_TARGET_VALUE	PREDICTED_TARGET_VALUE	VALUE
1	0	1	34851
2	1	1	121350
3	0	0	5472

Figure 18 - Confusion matrix showing the DT accuracy

Testing the decision tree model

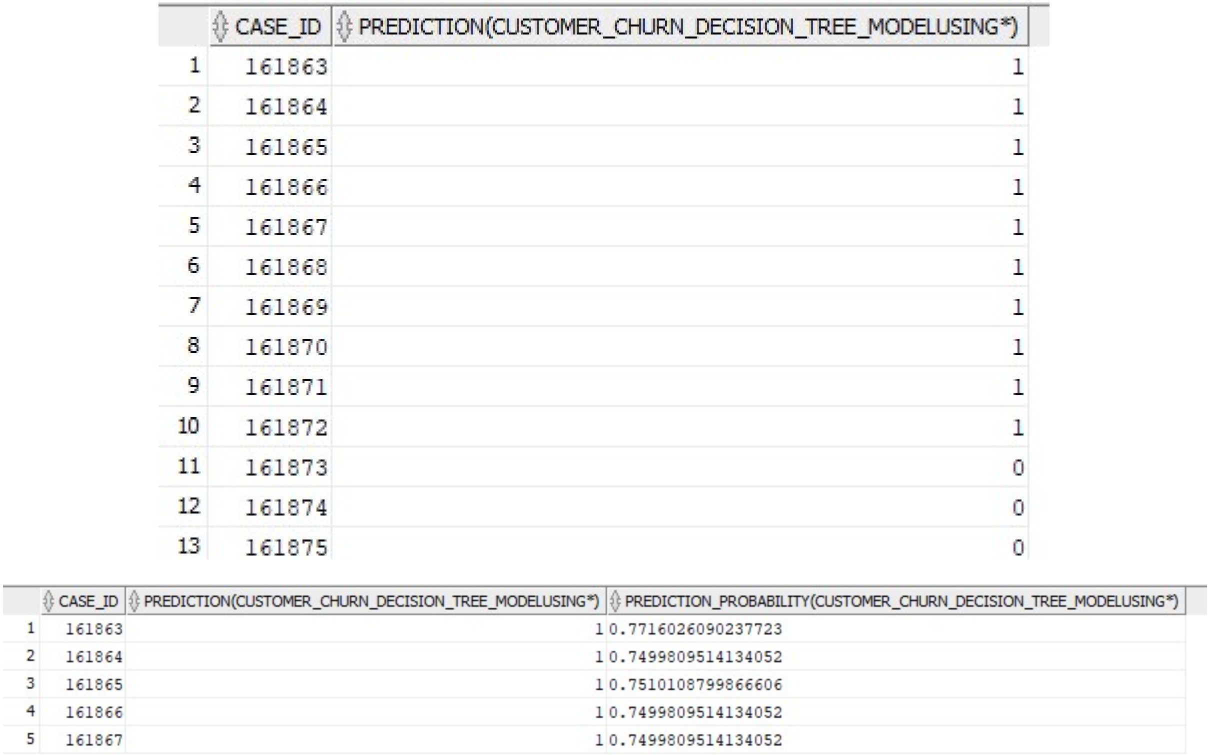
The model was then tested on the testing data set. The cost function was also computed to determine how good the model performed in relation to the training data.



	CASE_ID	PREDICTION	PROBABILITY	COST
1	161863	1	0.7716026090237723	0.2283973909762277
2	161863	0	0.2283973909762277	0.7716026090237723
3	161864	1	0.7499809514134052	0.25001904858659485
4	161864	0	0.25001904858659485	0.7499809514134052
5	161865	1	0.7510108799866606	0.2489891200133394
6	161865	0	0.24898912001333945	0.7510108799866606
7	161866	1	0.7499809514134052	0.25001904858659485
8	161866	0	0.25001904858659485	0.7499809514134052
9	161867	1	0.7499809514134052	0.25001904858659485
10	161867	0	0.25001904858659485	0.7499809514134052
11	161868	1	0.7537836322869955	0.24621636771300448
12	161868	0	0.24621636771300448	0.7537836322869955
13	161869	1	1.0	0.0

Figure 19 - probability and cost function on testing data

The model was then applied to real-time data where it generated 13 cases where 3 customers would stay and 13 leave. Prediction accuracy was also determined for 5 cases (75% to 77%).

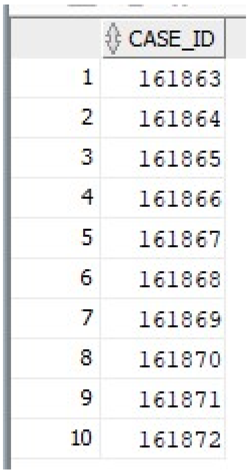


	CASE_ID	PREDICTION(CUSTOMER_CHURN_DECISION_TREE_MODELUSING*)
1	161863	1
2	161864	1
3	161865	1
4	161866	1
5	161867	1
6	161868	1
7	161869	1
8	161870	1
9	161871	1
10	161872	1
11	161873	0
12	161874	0
13	161875	0

	CASE_ID	PREDICTION(CUSTOMER_CHURN_DECISION_TREE_MODELUSING*)	PREDICTION_PROBABILITY(CUSTOMER_CHURN_DECISION_TREE_MODELUSING*)
1	161863	1	0.7716026090237723
2	161864	1	0.7499809514134052
3	161865	1	0.7510108799866606
4	161866	1	0.7499809514134052
5	161867	1	0.7499809514134052

Figure 20 - prediction on real-time data

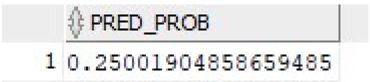
As part of real time prediction, a query to predict the 10 customers most likely to leave was conducted. This is important as it has real world applications. Customer Service could contract these customers and try entice them to say.



	CASE_ID
1	161863
2	161864
3	161865
4	161866
5	161867
6	161868
7	161869
8	161870
9	161871
10	161872

Figure 21 - 10 customers who are predicted to leave

A what-if analysis on dummy data was also made which predicted that a 46 year old non-working customer who spent 432 days on a cosmopolitan contract who spent over 9 minutes making an off-peak for 84c was going to leave.



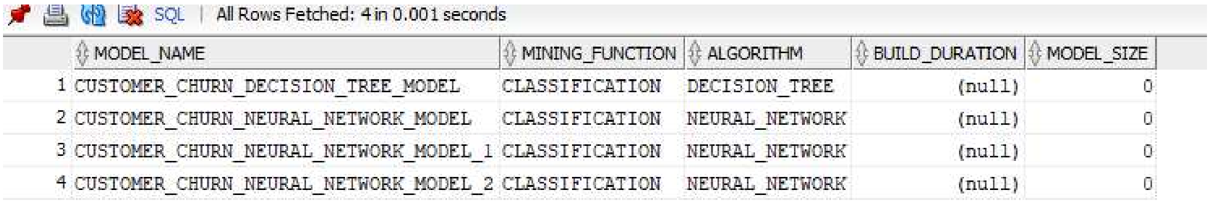
	PRED_PROB
1	0.25001904858659485

Figure 22 - what-if analysis on dummy data

Neural Network

A neural network was then used as the second model. Default parameters were also used. This was after two different attempts (see Figure 23).

Building the neural network model

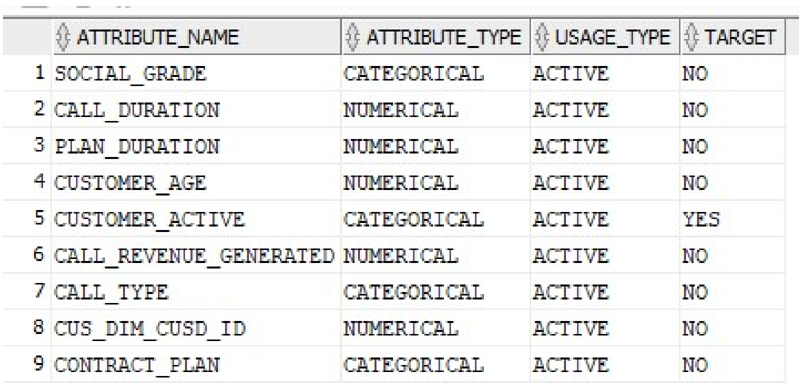


SQL | All Rows Fetched: 4 in 0.001 seconds

	MODEL_NAME	MINING_FUNCTION	ALGORITHM	BUILD_DURATION	MODEL_SIZE
1	CUSTOMER_CHURN_DECISION_TREE_MODEL	CLASSIFICATION	DECISION_TREE	(null)	0
2	CUSTOMER_CHURN_NEURAL_NETWORK_MODEL	CLASSIFICATION	NEURAL_NETWORK	(null)	0
3	CUSTOMER_CHURN_NEURAL_NETWORK_MODEL_1	CLASSIFICATION	NEURAL_NETWORK	(null)	0
4	CUSTOMER_CHURN_NEURAL_NETWORK_MODEL_2	CLASSIFICATION	NEURAL_NETWORK	(null)	0

Figure 23 - table showing machine learning model iterations.

More attributes were included in the neural network (e.g customer social grade)

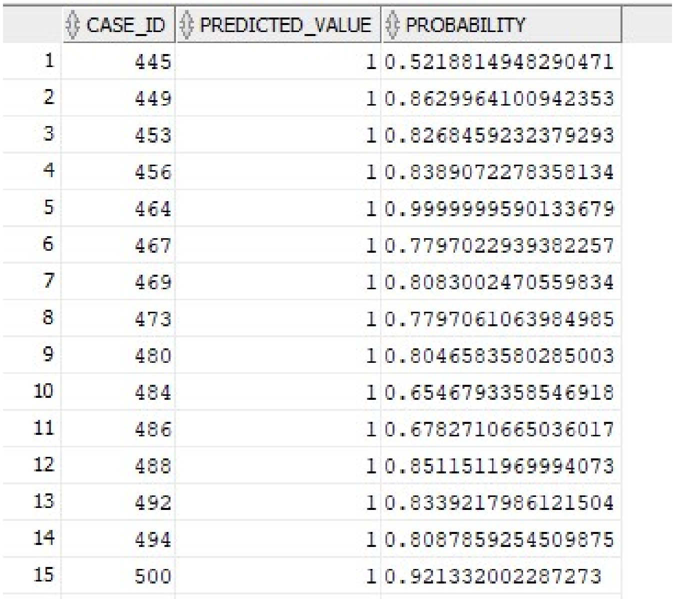


	ATTRIBUTE_NAME	ATTRIBUTE_TYPE	USAGE_TYPE	TARGET
1	SOCIAL_GRADE	CATEGORICAL	ACTIVE	NO
2	CALL_DURATION	NUMERICAL	ACTIVE	NO
3	PLAN_DURATION	NUMERICAL	ACTIVE	NO
4	CUSTOMER_AGE	NUMERICAL	ACTIVE	NO
5	CUSTOMER_ACTIVE	CATEGORICAL	ACTIVE	YES
6	CALL_REVENUE_GENERATED	NUMERICAL	ACTIVE	NO
7	CALL_TYPE	CATEGORICAL	ACTIVE	NO
8	CUS_DIM_CUSD_ID	NUMERICAL	ACTIVE	NO
9	CONTRACT_PLAN	CATEGORICAL	ACTIVE	NO

Figure 24 - attributes used by the neural network model

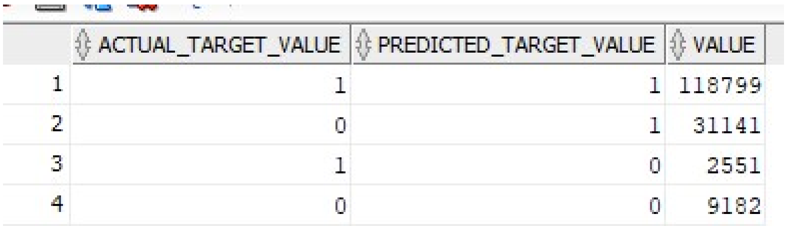
Training the neural network model

The neural network model was trained on the same dataset as the decision tree model. The model makes various predictions from 52% to 99% using a sigmoid activation function.



	CASE_ID	PREDICTED_VALUE	PROBABILITY
1	445	1	0.5218814948290471
2	449	1	0.8629964100942353
3	453	1	0.8268459232379293
4	456	1	0.8389072278358134
5	464	1	0.9999999590133679
6	467	1	0.7797022939382257
7	469	1	0.8083002470559834
8	473	1	0.7797061063984985
9	480	1	0.8046583580285003
10	484	1	0.6546793358546918
11	486	1	0.6782710665036017
12	488	1	0.8511511969994073
13	492	1	0.8339217986121504
14	494	1	0.8087859254509875
15	500	1	0.921332002287273

A Confusion matrix was again generated to determine the accuracy of the model on the training data. It returned a 79% predictability.

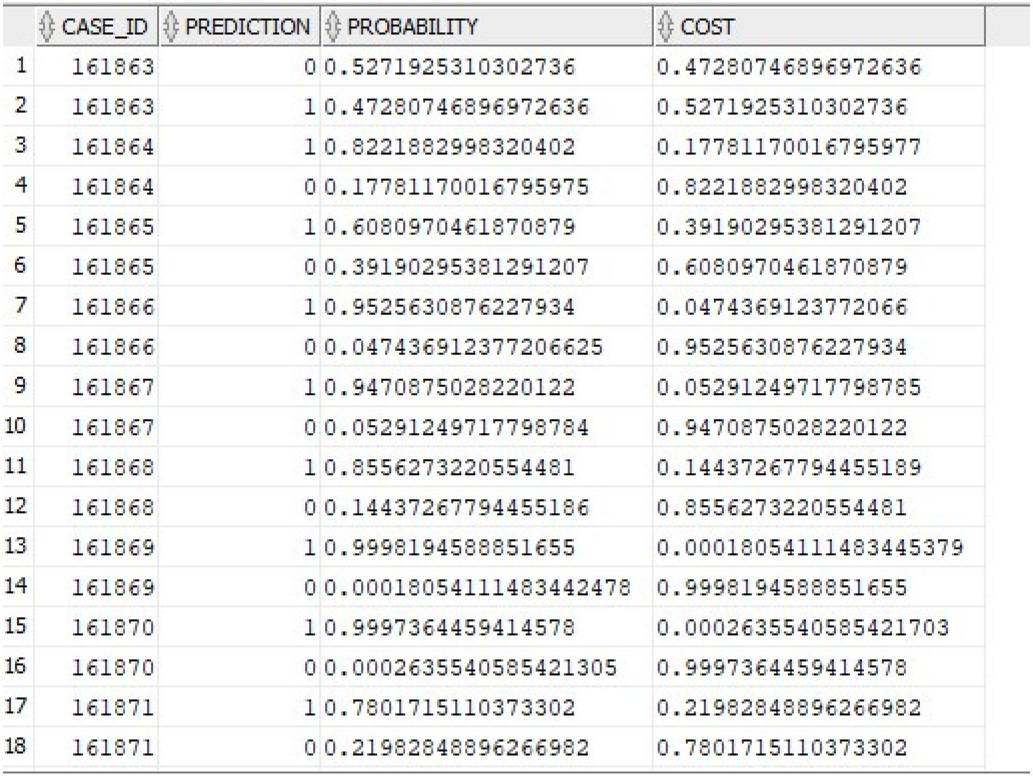


	ACTUAL_TARGET_VALUE	PREDICTED_TARGET_VALUE	VALUE
1	1	1	118799
2	0	1	31141
3	1	0	2551
4	0	0	9182

Figure 25 - confusion matrix showing the predictability of the NN model.

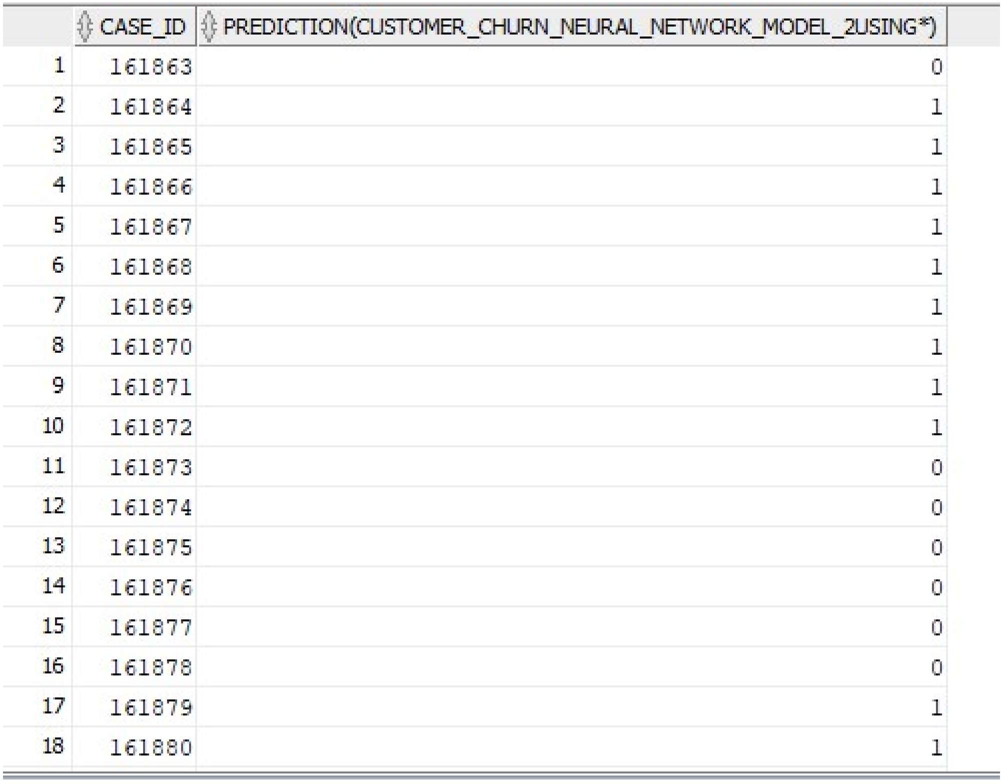
Testing the neural network model

The model was tested on the same data as the decision tree model. Cost function, displayed below, is used in backpropagation to allow the model to compute the gradient for the training set. It is the average loss over the entire training set. The smaller the cost function the better the model's ability to make predictions.

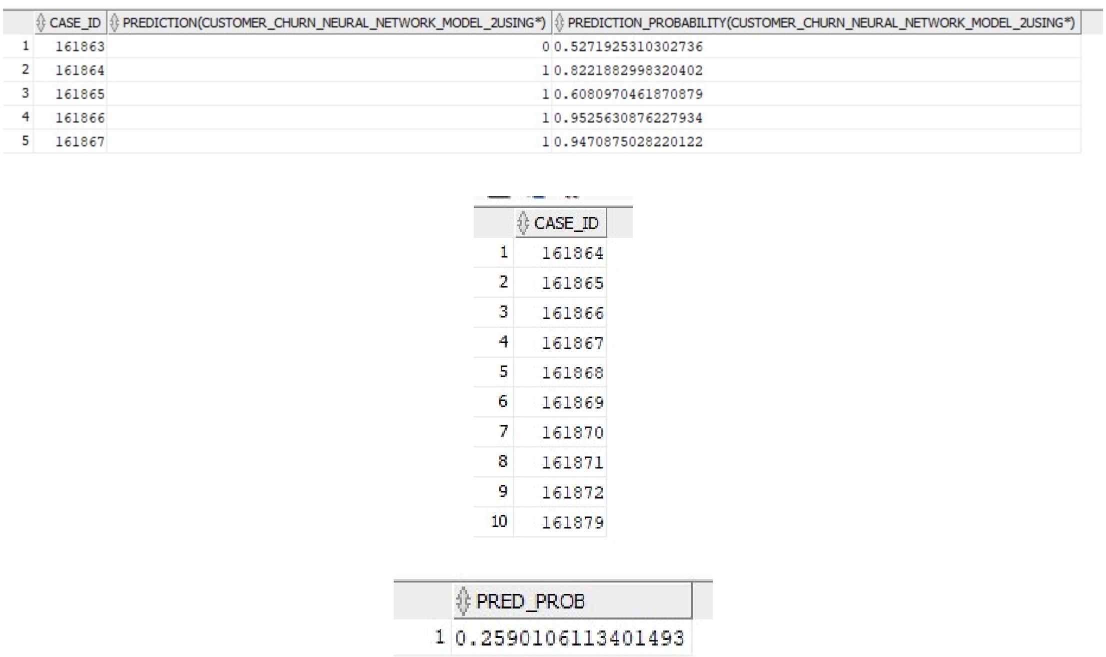


	CASE_ID	PREDICTION	PROBABILITY	COST
1	161863	0	0.5271925310302736	0.47280746896972636
2	161863	1	0.47280746896972636	0.5271925310302736
3	161864	1	0.8221882998320402	0.17781170016795977
4	161864	0	0.17781170016795975	0.8221882998320402
5	161865	1	0.6080970461870879	0.39190295381291207
6	161865	0	0.39190295381291207	0.6080970461870879
7	161866	1	0.9525630876227934	0.0474369123772066
8	161866	0	0.047436912377206625	0.9525630876227934
9	161867	1	0.9470875028220122	0.05291249717798785
10	161867	0	0.05291249717798784	0.9470875028220122
11	161868	1	0.8556273220554481	0.14437267794455189
12	161868	0	0.14437267794455186	0.8556273220554481
13	161869	1	0.9998194588851655	0.00018054111483445379
14	161869	0	0.00018054111483442478	0.9998194588851655
15	161870	1	0.9997364459414578	0.0002635540585421703
16	161870	0	0.0002635540585421305	0.9997364459414578
17	161871	1	0.7801715110373302	0.21982848896266982
18	161871	0	0.21982848896266982	0.7801715110373302

Figure 26 - NN testing and cost function results



	CASE_ID	PREDICTION(CUSTOMER_CHURN_NEURAL_NETWORK_MODEL_2USING*)
1	161863	0
2	161864	1
3	161865	1
4	161866	1
5	161867	1
6	161868	1
7	161869	1
8	161870	1
9	161871	1
10	161872	1
11	161873	0
12	161874	0
13	161875	0
14	161876	0
15	161877	0
16	161878	0
17	161879	1
18	161880	1



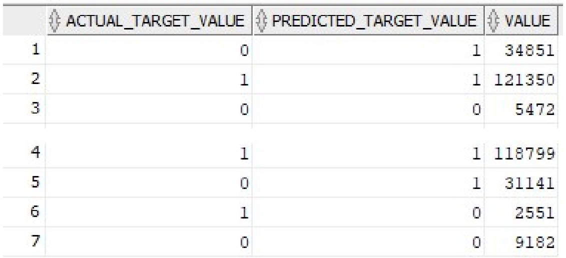
	CASE_ID	PREDICTION(CUSTOMER_CHURN_NEURAL_NETWORK_MODEL_2USING*)	PREDICTION_PROBABILITY(CUSTOMER_CHURN_NEURAL_NETWORK_MODEL_2USING*)
1	161863	0	0.5271925310302736
2	161864	1	0.8221882998320402
3	161865	1	0.6080970461870879
4	161866	1	0.9525630876227934
5	161867	1	0.9470875028220122

	CASE_ID
1	161864
2	161865
3	161866
4	161867
5	161868
6	161869
7	161870
8	161871
9	161872
10	161879

	PRED_PROB
1	0.2590106113401493

Figure 27 - several real type tests along with a what-if test on dummy data.

Results Discussion & Conclusion



	ACTUAL_TARGET_VALUE	PREDICTED_TARGET_VALUE	VALUE
1	0	1	34851
2	1	1	121350
3	0	0	5472
4	1	1	118799
5	0	1	31141
6	1	0	2551
7	0	0	9182

Figure 28 - row 1-3 DT results; row 4-7 NN results

A Decision Tree model and a Neural Network model were trained on the case table data to predict customer churn. Both Models were given 80% as training data and 20% as test data. Default Oracle hyperparameters were used. A review of both models' confusion matrix showed 78% predictability for the decision tree model and 79% for the neural network model.

References

1keydata. (2022). *Fact And Fact Table Types.* Retrieved from 1keydata.com: https://www.1keydata.com/datawarehousing/fact-table-types.html

Ault, M. (2003). *Oracle Data Warehouse Management.* Kittrell, North Carolina, USA: Rampant TechPress.

Hornick, M. (2019, 12 05). *Machine Learning in Oracle Database*. Retrieved from oracle.com: https://blogs.oracle.com/machinelearning/post/machine-learning-in-oracle-database

Kimball, R., & Ross, M. (1996). Telecommunications. In K. e. al, *The Data Warehouse Toolkit* (p. 236). USA: Wiley.