|  |  |
| --- | --- |
|  | |
| Working With Data–Assessment 2  TU060 : Data Warehouse Modelling / Data Analysis / Machine Learning using SQL | |
|  | |
| Ciaran Finnegan – Part Time – First Year  2021/2022  MSc in Computer Science (Data Science)  Student No : D21124026  4/1/2022 |  |
|  |  |

Table of Contents

1 Project Overview 3

1.1 High Level Description 3

1.2 Environment Assumptions 3

1.3 Project Execution Instructions 3

2 Section A: Business Drivers for Assignment 4

2.1 Background and Goals 4

2.2 Subject Area for Analysis 4

2.3 Key Stakeholders 4

2.4 KPIs 4

3 Section A: Data Modelling 5

3.1 Reasons for Design 5

3.2 Data Warehouse Schema 11

4 Section A: Implementing the Data Warehouse 12

4.1 Implementation using SQL Scripts 12

5 Section B: Data Analysis Using SQL 13

5.1 Data Analysis Objectives 13

5.2 Key SQL Reports + KPIs 14

5.3 Data Analysis – SQL Report Outputs from Data Warehouse 15

6 Section C: Machine Learning Using SQL 19

6.1 Overview of ML Process 19

6.2 Creating and Populating CASE Table for ML Modelling 20

6.3 Preparing Training and Test Data Sets 21

6.4 Create Models for ML Analysis on Customer Churn 22

6.5 Testing the Models 24

6.6 Comparing the Model Results 24

7 Appendices 26

7.1 Appendix 1 – SQL Scripts to build the Telecoms Database 26

7.2 Appendix 2 – SQL Scripts to Create the Data Warehouse Tables 26

7.3 Appendix 3 – SQL Scripts to Populate Data Warehouse Dimensions 28

7.4 Appendix 4 – SQL Scripts To Populate Data Warehouse Fact Table 31

7.5 Appendix 5 – SQL Scripts For All SQL Queries 35

7.6 Appendix 6 – SQL Scripts For ML Process 43

8 References 49

8.1 Data Warehouse Design 49

# Project Overview

## High Level Description

This document covers the design, implementation, and observations on all parts of the January 2022 CA(2) for the working With Data module in the TU060 Part Time/first Year MSc in Science (Data Science) course.

## Environment Assumptions

The project was developed in ORACLE SQL Developer.

All SQL scripts used in the assignment have been either embedded in the document and/or submitted separately as ***\*.SQL*** files.

These SQL files are grouped infolders in the ***\*.ZIP*** file that are numbered to indicate the sequence in which they should be executed in a test ORACLE database. The SQL Scripts are also numbered in order of execution.

## Project Execution Instructions

The project is submitted in a single ***ZIP*** file; ***WWD\_CA2\_D21124026.zip***.

The ***/DATABASE/1-CREATETABLE*** SQL Scripts folder contains the files creates the source data table with specific table names that are then referenced by later SQL Scripts. This is the first set of SQL scripts to be executed.

The *Voicemails* and *Calls* INSERT table scripts take longer to run than the others.

The ***/DATABASE/2-DATAWAREHOUSE*** SQL Scripts folder contains the files that create and populate the data warehouse tables from the telecoms database. This is the second set of SQL scripts to be executed.

**Note:-** The Time Dimension table takes a number of minutes to execute.

The ***/DATABASE/3-SQLQUERIES*** SQL Scripts folder contains the files that generate the data analysis output on the data warehouse. The is the third set of SQL scripts to be executed.

The ***/DATABASE/4-MACHINELEARNING*** SQL Scripts folder contains the final set of files that implement the predictive churn analysis on the customer records in the data warehouse.

The assignment report documents are in the /REPORTS folder.

# Section A: Business Drivers for Assignment

## Background and Goals

A new data warehouse model is being designed and built to improve customer profile data for this telecommunications company.

This report will explain to the key stakeholder group;

* The primary objectives driving the structure of the data warehouse.
* The range of reporting data that will be available from the warehouse.
* The predictive analysis that will be available to stakeholders in terms of possible customer churn.

## Subject Area for Analysis

In this assignment I have chosen to focus on an analysis of **Revenue performance.**

Therefore, the objective will be to ask which customers generate the most financial value for company based on their activity, profile, and call plans?

## Key Stakeholders

The KPI reports produced in this project are Revenue performance data based on monthly views of the data over a ‘Year-to-Date’ 2021 timeframe.

This information is thus not expected to be updated daily and is more strategic in value.

The key stakeholder for the type of reporting produced in this project would therefore be;

* **Senior Management.** Revenue performance data that could alter contract plans or rate types. These are the types of decisions that can only be actioned by those within our telecommunications company who control major resourcing and policy decisions.
* **Customer Services agents.** Those in the salesforce in our telecommunications company. The SQL queries in this assignment can be the basis for dashboards that provide a means for Customer Service to focus their attention on customers generating higher revenues.
* **Company owners/potential investors.** How much revenue is our telecommunications company generating?

## KPIs

Section 5.2 details the specific report objective for this assignment.

Section 5.3 displays the data analysis results from the reports built in SQL for this assignment.

# Section A: Data Modelling

## Reasons for Design

The Telecommunications Data Warehouse in this project is built following the design principles as described in Kimball’s four step process.

1. Identify the Business Process. Do not re-model the Business Department / Area.
2. Identify the Grain.
3. Choose the Dimensions.
4. Choose the Facts

These steps will be applied to the creation of a new Data Warehouse for this project, but this process could also be applied to the enhancement of an existing Data Warehouse to include a new business process reporting objective.

The objectives in creating the star schema model for the Data Warehouse are;

* Be simple.
* Be easy to use.
* Any process loading into these tables should be as simple as possible.
* Queries should perform well with SQL, or other Business Intelligence tools (which are not part of this assignment task).

### Identify the Business Process

This is the first step in designing the Data Warehouse.

A ‘Business Process’ can be defined as a natural operational activity performed in the organisation, in this case for our telecommunications company, that is supported by some form of data collection.

The following should be considered when identifying the process on which we wish to focus;

1. **Look at the business process not the business department.** This allows for data to be collated and reported on in a more consistent manner across the organisation. It helps in avoiding duplication of data, which might occur if we replicate the structure of business units in the Data Warehouse. In this project we are looking at the Revenue process by measuring Revenue performance, but we are not looking to build a Customer Service Department report. This assignment will look at Customer revenue generating activity across the company.
2. **Assess impact and risk in reporting on the chosen business process**. Impact is generating reports that the business actually want on a regular basis. The assignment will focus on identifying those types of customers, and their contract plans, who generate the most revenue, and being proactive to keep them in the business.
3. **In a real-world scenario, the business users would provide guidance on a data warehouse process.** Business users can also help decipher complex business processes. In this assignment we already have conducted an engagement with key stakeholders, which will feed into the structure of tables and the SQL queries/reports that will be executed.

For this assignment, the business process is to capture revenue generating activity by the customers of the company.

Building the data warehouse will allow the company to have a better analytical view of the revenue streams per customer, which will then help information subsequent decision-making processes.

### Identify the Grain

This is the second step in designing the Data Warehouse.

This is the most important phase of the design process. Redesigning a Data Warehouse at a later date to increase the level of granularity could be an expensive and time consuming process.

The resultant Fact table will be at the centre of our star schema. This table contains all of the measurable facts about the captured business process. We will use the Fact table to extract information on key Revenue data points for customers in this telecommunications company.

I have followed three particular guidelines in my project to identify the correct level of granularity when considering the design of this Data Warehouse.

1. **What is represented by one ‘Fact’ row?** What level of granularity is captured?
2. **Choose the most atomic level of information.** The data cannot be meaningfully subdivided any further. It also allows for easy and effective aggregations.
3. **Allow scope for future reporting requirements**. It is hard to predict future user requirements so the granularity is important to allow further, possibly ad-hoc, reporting requirements.

In my Fact table the focus is on measuring revenue from customer calls. Thus, in my dimensional model one Fact row represents *one call event to/from a specific customer at a specific time.*

This is a lower grain than total call events by a customer in each day, as the customer may make multiple calls or voicemails on one day in different time periods (peak or off-peak).

The term ‘call event’ is significant because the telecommunications company distinguishes, in separate database tables, between;

* A voice call made ***by a*** customer.
* A voicemail left ***by a*** customer.
* A call from a Customer Service agent ***to a*** customer.

We will not subdivide out the call ***to*** and ***from*** a customer. It will be assumed that calls made by a customer are a charge to them and revenue to the company. To maintain this level of granularity, calls from a Customer Services agent are not a charge to the telecommunications company itself and will be represented as a *zero*-revenue item. This is consistent with the *call\_rate* value of ‘0’ in the source ***Call\_Rates*** database table.

### Choose the Dimensions

This is the third step in designing the Data Warehouse.

Guidelines for this process can be summarised as follows;

1. Who, what, where, when?
2. Best attributes are descriptive.
3. De-normalizes design focuses on high performance reads.
4. Use smallest data types possible.

To capture the attributes of the Revenue performance process for our telecommunications company, I need to have the information on **who** (Customer) was involved in **what** call-event and **when**.

This question dictated the choice of the dimension tables I selected for my Data Warehouse schema, and the attributes in these tables;

* Customer (***dw\_dimtblCustomer***)
  + Phone Number – The Fact table stores the Connection ID; therefore, this text variable can be stored in the Customer DIM table.

* + Plan Name – text description of Customer Plan.
  + Plan Id – Numeric Identifier for the Customer Plan. Added in the dimension table to aid the update of values for call charge in the FACT table.
  + Social Class – text description of socio-economic demographic into which the customer has been classed.
  + Customer Age – the current age in years of the customer, which is extracted from the Date of Birth in the Customers table.
  + Out of Contract – a ‘Y’/’N’ flag, which is based on the existence of a Contract End Date in the Customer table. The flag forms a key input to the CASE table used in the Machine Learning customer churn predictive analysis. A ‘Y’ value indicates that the customer contract has ended.
* Call Event (***dw\_dimtblCallEvent***)
  + Connection Id – unique identifier for the call event.
  + Call event type – text description of call type – peak, roaming, voicemail, etc.
  + Call Event Type Id - – Numeric Identifier for the Call Type. It requires some data conversion to distinguish between ***Peak*** and ***Off-Peak*** for actual calls. Added in the dimension table to aid the update of values for call charge in the FACT table.
* Time (***dw\_dimtblDateTime***)
  + Calendar Date – text description from the Customer Support, Voicemail, and Calls tables.
  + Call Event Date – ORACLE DATE variable converted from ‘Calendar Date’ text.
  + Cal Timestamp - ORACLE Timestamp variable converted from ‘Calendar Date’ text.
  + Day of Week – Number representing day of week, Monday = 1, and so on.
  + Month of Year – Number representing month in year, January = 1, and so on.
  + Table Source – What type of call was made at this time.

The TimeDate dimension table (*dw\_dimtbltimeDate*) is built to provide additional date granularity and a conversion of part of the date into an integer format to improve reporting performance.

A new ‘surrogate key’ has been created for each of the dimension tables. It is a simple numeric value that I have set to increment in the SQL scripts used in the CREATE TABLE routines.

The surrogate key is necessary to uniquely identify each row in the dimension table and to avoid any confusion with the source Primary Keys from the ‘operational’ database tables of the telecommunications company. This is particularly useful if the key structure in the telecommunications company operational database changes in the future. Such changes will not then have a knock on impact on the Data Warehouse and reporting applications should still be valid.

The surrogate keys of each dimension table are simple integer values and are also added to the Fact table. This is done to minimise the number of joins needed to fetch data, which improves the response time of queries (as does the use of simple integer key values).

The SQL used to implement the DIMENSION tables can be seen in Section 7.2.

### Choose the Facts

This is the fourth step in designing the Data Warehouse.

The Fact table exists at the centre of the star schema, as can be seen in Section 3.2.

Defining the measures for the Fact table should follow guidelines such as these;

1. **How does the business measure success?** For Revenue performance we are looking at the charge totals for call events, and which customers are generating the most revenue?
2. **The best measures are fully additive**. It should be possible to roll up the measures and easily perform aggregations. For example, call charges in the Fact table can be easily summed at Customer or Contract Plan level.
3. **Data access tools, such as Tableau, PowerBI, (or even SQL Scripts) are suitable for non-additive measures.** *Year To Date* totals are calculated in the output of one of my SQL Scripts but would not be a meaningful unit of data in the Fact table.

The facts are numeric values that correspond to the grain of the table, as defined in Section 3.1.2.

The Fact table ***dw\_facttblCallRevenue*** will be created in out Telecommunications company data warehouse.

The columns for this Fact table can be identified as follows;

1. Date Time Foreign Key – link to DateTime Dimension table.
2. Customer Foreign Key - link to Customer Dimension table.
3. Call Event Foreign Key - link to Call Event Dimension table.
4. Cost Per Minute for Call Event – rate for this Call Event. A zero value indicates a call to the customer from Customer Service.
5. Duration of Call Event – recorded in seconds.
6. Charge Generated for Call Event

The Charge Generated (Revenue) amount per call event is a relatively simple metric to report on as it is stored in the Fact table, and allows for more straightforward, and performant, SQL queries.

The SQL used to implement the FACT table can be seen in Section 7.3.

## Data Warehouse Schema

The Data Warehouse for this project will be implemented with a Star Schema design.

This involves one central Fact table surrounded by a number of Dimension tables.

Diagram

Description automatically generated

The operation database, from which this Data Warehouse is built, has a normalised relational structure (or would do if fully operational).

To optimise queries on the Data Warehouse the tables are effectively partially ‘de-normalised’.

The Dimension tables primarily contain descriptive information. The Fact table contains keys to all the dimension table Primary keys, and all the measurable attributes required to meet the reporting purpose of this Data Warehouse.

# Section A: Implementing the Data Warehouse

## Implementation using SQL Scripts

Section 1.3 of this document explains the sequence in which the assignment SQL scripts must be unpacked and executed.

ZIP files containing the SQL script folders accompany this report file. These files and screenshots of the SQL code are also embedded in Section 7 of this report.

The SQL Developer GUI was used to lead the source assignment ***csv*** files into an ORACLE database. Those CREATE TABLE scripts were autogenerated by SQL Developer. This process is described in more detail in the supplementary report document ***Working With Data - CA2 - Data Imports - Student Ciaran Finnegan d21124026 v1-3 020122.docx***, which accompanies this main report.

The work to implement the data warehouse, build and execute SQL query analysis, and perform Machine Learning predictions, was all developed directly in SQL through ORACLE SQL developer.

# Section B: Data Analysis Using SQL

## Data Analysis Objectives

The assignment lists the following desired features for this data warehouse application;

* Identify how valuable a customer is to the company relative to other customers.
* Build up a picture of their customers’ profiles.
* Determine whether a customer’s behaviour patterns have changed recently.
* Identify the call plans which bring in the most revenue.

In this section of the report, we look at a revenue analysis from the data warehouse, and the underlying SQL queries that extracted the data.

The focus was on the following questions;

* **Who** are the customers generating most revenue?
* **What** types of customer activity is generating the most revenue?
* **How** is revenue performance changing over time?

The information in the output from theses SQL queries is very high level but it intended as an ‘at a glance’ overview for Senior Management.

The message is clear and focuses on the value of customers generated higher amounts of revenue.

## Key SQL Reports + KPIs

**SQL Report 1: Top 100 Customers – in last 30 days (by Revenue)**

* A snapshot view of the customer accounts, identified by phone number, which have generated the most revenue in the last 30 days. The timeframe is based on the last thirty days tracked in the data warehouse.

**SQL Report: Revenue Per Plan Per Month**

* A snapshot view of revenue trends broken down by customer call plans. Data is tracked over a year-to-date period (2021) on a monthly basis (this is the range of the data in the data warehouse).

**SQL Report 3: Top 100 Customers – in last 30 days (by Activity)**

* A snapshot view of the customer accounts, identified by phone number, which have been most active, as measured by call event duration.

**SQL Report 4: Top 20 Customers – Revenue Patterns (by Month)**

* Taking the Top 20 customers from the last month, display the revenue trends for the preceding three months.

**SQL Report 5: Moving Average (by Month) of Revenue from Contract Plans**

* Taking each of the three contract plans show the moving average in revenue that are recorded for the first four months of the year (2021).

**SQL Report 6: Top 100 Customer that Customer Services Contacted in 2021 (by Social Class)**

* A snapshot of the Top 100 customers who have been in contact with Customer Services the most in 2021 (year-to-date). The social grade is included to provide an indicator if this attribute seems to impact the level of contact.

## Data Analysis – SQL Report Outputs from Data Warehouse

The SQL that generates the output show here is represented in Section 7.4 of the Appendices.

In practice, these queries would be most likely to form the basis of inputs to tools like Tableau or PowerBI for more effective presentation and dissemination of data. For this assignment, ORACLE COLUMN functions are used to improve display presentation through SQL Developer.

**SQL Report 1: Top 100 Customers – in last 30 days (by Revenue)**

Partial view of (from the top) of the Top 100 Customer report.

Table

Description automatically generated

The purpose of this report is to provide a focus to Customer Services on who are the most valuable customers on the Telco network. This data could be extended over time into trends on Customers to see if previous high earners have dropped from the ‘Top 100’.

**SQL Report 2: Revenue Per Plan Per Month**

Display each of the Telco Contract Plans in terms of the revenue generated for each of the last four months.

Table

Description automatically generated

This report shows that all Contract Plans are dipping in revenue as the months progress through the early part of 2021.

No one plan is performing particularly worse than another so any corrective action should apply equally across all plan types.

**SQL Report 3: Top 100 Customers – in last 30 days (by Activity)**

Display the most active 100 customers, by duration of all Telco call events, over the last 30 days and include revenue for additional context (excerpt);

Table

Description automatically generated

This report compliments Report 1 in that it shows that the most active customers do not necessarily equate to the most profitable ones for the Telco.

**SQL Report 4: Top 20 Customers – Revenue Patterns (by Month)**

This report selects (excerpt below) the most profitable 20 customers in April and presents a horizontal view of how much revenue each has generated, month by month, since the start of 2021.

Table

Description automatically generated

The purpose of this report is to provide extra focus on the high revenue clients and see if there is any pattern to their Telco usage since the start of the year.

Almost all customers show a dip in revenue in February and March and no one customer stands out for additional Customer Service attention.

**SQL Report 5: Moving Average (by Month) of Revenue from Contract Plans**

This report supplements Report 2 by showing a moving average trend of how each particular contract is performing in terms of revenue generation.

Table

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

Table

Description automatically generated

This data further re-enforces the point that all the Telco Contract Plans are generating a reducing amount of revenue. Hence, a relatively broad plan of action by the Telcom may be required.

**SQL Report 6: Top 100 Customers that Customer Services Contacted in 2021 (by Social Class)**

This report looks at the 100 most contacted customers in 2021. It allows a view as to whether social class influences customer contact.

Text

Description automatically generated

The output shows that the Telco Customer Services department does not appear to be influenced by social class in terms of customer interaction, as the social groups appear to be reasonably distributed in this report (excerpt above).

# Section C: Machine Learning Using SQL

## Overview of ML Process

Section 3 of this report describes the process around designing the data warehouse Dimension and Fact Tables.

Section 7.2 and 7.3 provide details of the SQL scripts used to implement the actual data warehouse in our ORACLE environment.

The section of the report assignment explains how we will adapt the data warehouse information to execute machine learning predictive analysis on customer churn.

The data preparation process for this ORACLE machine learning process will begin with the design and implementation of a suitable CASE table for the customer churn problem.

The Case table will be a new database table that is extrapolated from our existing data warehouse but designed specifically for our customer churn data mining models. All the information required for the modelling process is contained in this Case table (there is no dependencies on external database tables through joins).

Table columns/granularity in the Case table is defined by the question; ‘can we predict if a customer is likely to churn?’

One column will be specifically identified as the label, for which we are trying to predict a value.

Section 6.2 describes the structure of our Case table, while the remaining sections describe how our data is split into separate train and tests sets, the ML predictive models that are built and tested, and the outcomes achieved.

## Creating and Populating CASE Table for ML Modelling

The Case table, named ***dw\_CaseMLChurn\_tbl***, for our customer churn ML problem is created with the following structure;

Text

Description automatically generated with medium confidence

* The ***CaseID*** is required as the column containing an ID Integer value that uniquely identifies each row in the Case table.
* ***Phone\_Number*** is actually another unique identifier for the customer. As such, it will not add to the eventual ML churn model accuracy, but it is a useful means of manual data validation, if required at some point.
* ***Customer\_Age*** is an integer value for the customer age, as extracted from the *dw\_dimtblCustomer* Customer Dimension table, through a join from the Fact table.
* ***Plan\_ID*** is a numeric identifier, again from the *dw\_dimtblCustomer* Customer Dimension table, that records the contract plan the customer to which the user is assigned.
* ***Social\_Class*** is a text description of the customer’s socio-economic group.
* ***Total\_Num\_Calls*** is a count of calls made by the customer in the data warehouse time period.
* ***Call\_Revenue\_Total*** is the total amount of revenue generate for this Telco by the customer in the data warehouse time period.
* ***Call\_Charge\_Avg*** is the average charge incurred by the customer taking all call events into account in the data warehouse time period.
* ***Call\_Duration\_Total*** is the combined time spent by a single customer on call events in the data warehouse time period.
* ***Call\_Duration\_Average*** is the mean time spent by a single customer on call events in the data warehouse time period.
* ***Days\_Since\_Last\_CallEvent*** is counted from the date of last active call record in the data warehouse (29th April 2021). The value tracks the last time the customer was active in the Telco system.
* The column ***Out\_of\_Contract*** contains the **label** for the Case table. This Machine Learning model will attempt to predict if this value could indicate that customer may churn.

All the information required for our ML churn process is contained in this Case table, and each row contains the information on one unique customer.

The SQL to implement the CASE table is given in Section 7.5.1.

## Preparing Training and Test Data Sets

Following common Machine Learning practice, the data in the Case Table is split according to following ratios;

* **80%** of the data is used as a Training set to build the Machine Learning models. A SQL Sample of 80% is taken from ***dw\_CaseMLChurn\_tbl*** to form the Training Data.
* The remaining **20%** is used as a Test Data set to assess the accuracy of the model. None of this is data used in the creation of the models themselves.

The SQL is set up to ensure the data is mutually exclusive across the Training and Test datasets.

Text

Description automatically generated

## Create Models for ML Analysis on Customer Churn

In this assignment two models will be generated based on the following algorithms;

* Naïve Bayes
* Decision Tree.

Both of these are appropriate for the Classification problem in this assignment, where is it required to predict the churn flag ‘Y’ or ‘N’ for a customer.

The SQL used to implement these models is given in Section 7.5.3.

Common settings for the model creation are:

* Set as a Classification model using the ***dbms\_data\_mining.classification*** value for the **mining\_function** setting.
* The ID column is the CASEID column. The label is the OUT\_OF\_CONTRACT column.

Key additional aspects of the model implementation for Naïve Bayes (NB);

* In the ORACLE DBMS\_DATA\_MINING.CREATE\_MODEL function the setting table entries are left as default (NULL), which invokes the Naïve Bayes algorithm.

The NB model setting entries created for this assignment, and stored in the all\_mining\_model\_settings table are;

Text

Description automatically generated

Running a SELECT from the ***all\_mining\_model\_attributes*** table for our NB model shows that these are the Case table variables that are considered significant for our NB modelling (this might help in future feature engineering);

Graphical user interface, application

Description automatically generated

Key aspects of the model implementation for the Decision Tree Model (DT);

* In the ORACLE DBMS\_DATA\_MINING.CREATE\_MODEL the settings are set to select a Decision Tree algorithm with Automation data preparation set to ‘ON’.

The DT model setting entries created for this assignment, and stored in the all\_mining\_model\_settings table are;

Graphical user interface, text, application, table

Description automatically generated

Running a SELECT from the ***all\_mining\_model\_attributes*** table for our DT model shows that these are the Case table variables that are considered significant for our DT modelling (this might help in future feature engineering);

Graphical user interface, application

Description automatically generated

***Note:-*** Neither algorithm model considers the customer ‘Age’ attribute as significant in the modelling process.

## Testing the Models

Section 7.5.4 shows the SQL used to generate predicted results into results tables for each model.

The Test dataset prepared earlier is run against the NB and DT models to produce ‘Predicted Values’ in a Results table with and associated probability attached to these predictions.

The following code and results snippers show a partial output from this process.

Graphical user interface, text, application, email

Description automatically generated

## Comparing the Model Results

Which model works best with the data for the Customer churn predictive analysis?

For this assignment, a separate Confusion Matrix and Accuracy score have been generated for each model and presented.

Section 7.5.5 shows the SQL used to implement the Confusion Matrices.

The SQL in Section 7.5.6 is used to display the comparative results.

The output from both Confusion Matrices is as follows;

**Naive Bayes**

The Confusion Matric for the NB model on the Test data produced the following results.;

Graphical user interface, text, application, email

Description automatically generated

**Decision Tree**

The Confusion Matric for the DT model on the Test data produced the following results.;

Graphical user interface, table

Description automatically generated

Accuracy is a count of all True and False positive results over all results.

Both models score well, with the DT results slightly better that NB (92.8% v 91.78% respectively).

# Appendices

## Appendix 1 – SQL Scripts to build the Telecoms Database

The supplementary report ***Working With Data - CA2 - Data Imports - Student Ciaran Finnegan d21124026 v1-3 020122.docx***, which accompanies this main report as part of the WWD CA2 submission, contains the files that populate the starting Telco database.

This document also describes the process by which the data was imported and set up

.

## Appendix 2 – SQL Scripts to Create the Data Warehouse Tables

Section 3.1 of this document explains the logic behind the design of the FACT and DIMENSION tables for this Telco data warehouse.

Below are the SQL Scripts use to create the Fact and Dimension tables, prior to data population.

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

## Appendix 3 – SQL Scripts to Populate Data Warehouse Dimensions

The Dimension tables in the Data Warehouse are populated in two stages.

SQL SCRIPT TWO populates the ***dw\_dimtblCustomer*** and ***dw\_dimtblCallEvent*** tables, which are the Customer and Call event Dimension tables respectively.

SQL SCRIPT THREE populates the ***dw\_dimtblDateTime*** table, which is the Time dimension table. This takes a VERY LONG time to run (10+ minutes).

**Customer and Call Event Dimension Tables**

The queries in these tables collect, largely categorical, information from the main tables and add them into the Dimensions for Customers and Call Events.

A ‘Call Event’ can be either a phone call, a voicemail, or a customer service call. All this data is combined into a single ‘Call Event’ Dimension table.

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Time Dimension Table**

This table is built with three successive INSERT statements from the Call Event tables (main database and Dimension tables).

Additional date formats are added through UPDATES.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

## Appendix 4 – SQL Scripts To Populate Data Warehouse Fact Table

The Fact table in the Data Warehouse is populated in the data warehouse by executing the SQL commands in the attached \*.SQL file (reproduced below);

The Fact table is built from a UNION of data in the Dimension tables.

Then the Fact table is updated to ensure that all FOREIGN KEY references from the Fact table to the Dimension tables are correct.

Additional UPDATES take place for calculated charge fields to fill out the revenue data for each call, in line with the desired grain of the Fact table.

Text

Description automatically generated

Text, timeline

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

A picture containing text

Description automatically generated

## Appendix 5 – SQL Scripts For All SQL Queries

### SQL Report 1: Top 100 Customers – in last 30 days (by Revenue)

The SQL report uses ORACLE COLUMN functions to improve the presentation of the output column headings.

The ‘last 30 days’ is based on reading the offset from the last date entry in the data warehouse.

The FACT and DIMENSION tables are joined to pull back the Top 100 customer by revenue.

Graphical user interface, text

Description automatically generated

### SQL Report 2: Revenue Per Plan Per Month

This report also uses ORACLE Column functions for output presentation.

To display the data in a more readable horizontal format an ORACLE PIVOT function is used at the end of the script to re-orient the output data.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated with medium confidence

### SQL Report 3: Top 100 Customers by Activity – last 30 Days.

Graphical user interface, text, application

Description automatically generated

### SQL Report 4: Top 20 Customers – Revenue Patterns (by Month)

Subquery used to first isolate the customers who were most profitable for April, and then use this sub-set of date to extract monthly revenue.

Further use of PIVOT function to re-orient data for display. YTD values added as a new calculated column.

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

### SQL Report 5: Monthly Moving Average of Revenue Per Plan

Another nested subquery is used to pull Contract Plan revenue data out of the data warehouse.

The table then calculates the current month revenue and the average of the months so far in the current rows.

Text

Description automatically generated

Text

Description automatically generated

### SQL Report 6: Customers Most Contacted by Customer Service

Text

Description automatically generated

## Appendix 6 – SQL Scripts For ML Process

### Case Table – SCRIPT FIVE

The **CASE Table**, which is the basis of the data for the ORACLE Machine Learning process in this assignment is created with the following SQL;

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

### Set Up Training and Test Views – SCRIPT SIX

Views created to contain Training and Test datasets for model creation and assessment.

Timeline

Description automatically generated

### Build ML Models – SCRIPT SEVEN and EIGHT

SQL Scripts to build NB and DT models for Customer Churn predictive analysis.

**Naive Bayes**

Text

Description automatically generated

**Decision Tree**

**Text

Description automatically generated**

**Text

Description automatically generated**

### Model Test Results – SCRIPT NINE

The SQL below runs the Test data against both models to produce predicted outputs into a results VIEW.

Text

Description automatically generated

### Create Confusion Matrices – SCRIPT TEN and ELEVEN

The following SQL procedure create a Confusion Matrix output for both the NB and DT test data output.

**Naive Bayes**

Graphical user interface, text, application

Description automatically generated

**Decision Tree**

Graphical user interface, text, application

Description automatically generated

### Display Confusion Matrices Results – SCRIPT TWELVE

The following SQL code will display the output of both Confusion Matrices. A number of ORACLE COLUMN functions are used to improve the results presentation.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# References

## Data Warehouse Design

In additional to the class Module notes I followed the data warehouse design principles that were discussed in these two YouTube training videos;

***Designing Your Data Warehouse from the Ground Up* -** <https://youtu.be/patBYUGwsHE>

***Implementing a Data Warehouse with SQL Server, 01, Design and Implement Dimensions and Fact Tables* -** <https://youtu.be/StoWu2A8Ufs>