Build\_Data:

Text

Description automatically generated

Test Data:

Text

Description automatically generated

Apply\_Data:

Text

Description automatically generated

SVM Model:

Settings Table:

REM Create the settings table

CREATE TABLE CC\_svm\_model\_settings (

setting\_name VARCHAR2(30),

setting\_value VARCHAR2(30));

Text

Description automatically generated

Populate Settings Table:

REM Populate the settings table

BEGIN

INSERT INTO CC\_svm\_model\_settings VALUES

(dbms\_data\_mining.algo\_name,

dbms\_data\_mining.algo\_support\_vector\_machines);

INSERT INTO svm\_model\_settings VALUES

(dbms\_data\_mining.prep\_auto,

dbms\_data\_mining.prep\_auto\_on);

COMMIT;

END;

/

Text

Description automatically generated

Model Creation :

REM Create the model using the specified settings

BEGIN

DBMS\_DATA\_MINING.CREATE\_MODEL(

model\_name => 'Card\_svm\_model',

mining\_function => dbms\_data\_mining.classification,

data\_table\_name => 'Build\_data',

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

target\_column\_name => 'DEFAULTPAYNXTMNT',

settings\_table\_name => 'CC\_svm\_model\_settings');

END;

/

Text

Description automatically generated

Model Testing:

REM testing the model

SELECT DEFAULTPAYNXTMNT AS actual\_target\_value,

PREDICTION(Card\_svm\_model USING \*) AS predicted\_target\_value,

COUNT(\*) AS total\_value

FROM test\_Data

GROUP BY DEFAULTPAYNXTMNT, PREDICTION(Card\_svm\_model USING \*)

ORDER BY 1, 2;

Text

Description automatically generated

Accuracy:

REM calculating the model's accuracy

COLUMN ACCURACY FORMAT 99.99

SELECT (SUM(correct)/COUNT(\*))\*100 AS accuracy

FROM (SELECT DECODE(DEFAULTPAYNXTMNT,

PREDICTION(Card\_svm\_model USING \*), 1, 0) AS correct

FROM test\_Data);

Text

Description automatically generated

Prediction Data As given by Akthar:

REM first we create the view to identify the customers

CREATE OR REPLACE VIEW TOP10\_CC\_Cust AS

SELECT CARD

FROM (SELECT CARD, rank() over (order by

PREDICTION\_PROBABILITY(Card\_svm\_model, 1 USING \*)

DESC, CARD) RNK

FROM apply\_data

WHERE CREDIT\_LIMIT < 50000)

WHERE RNK <= 10

ORDER BY RNK;

Text

Description automatically generated

REM now we shall display the customers’ names

SELECT CARD

FROM CreditCardsV2

WHERE CARD IN (SELECT \* FROM TOP10\_CC\_Cust);

Text

Description automatically generated

NAÏVE BAYES:

Model Settings:

REM Create the settings table

CREATE TABLE Card\_naive\_model\_settings (

setting\_name VARCHAR2(30),

setting\_value VARCHAR2(30));

Text

Description automatically generated

Populate Settings Table:

REM Populate the settings table

BEGIN

INSERT INTO Card\_naive\_model\_settings VALUES

(dbms\_data\_mining.algo\_name,

dbms\_data\_mining.algo\_naive\_bayes);

INSERT INTO Card\_naive\_model\_settings VALUES

(dbms\_data\_mining.prep\_auto,

dbms\_data\_mining.prep\_auto\_on);

COMMIT;

END;

/

Text

Description automatically generated

Model Creation:

REM Create the model using the specified settings

BEGIN

DBMS\_DATA\_MINING.CREATE\_MODEL(

model\_name => 'Card\_nb\_model',

mining\_function => dbms\_data\_mining.classification,

data\_table\_name => 'Build\_data',

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

target\_column\_name => 'DEFAULTPAYNXTMNT',

settings\_table\_name => 'Card\_naive\_model\_settings');

END;

/

Text

Description automatically generated

Testing Model:

REM testing the model

SELECT DEFAULTPAYNXTMNT AS actual\_target\_value,

PREDICTION(Card\_nb\_model USING \*) AS predicted\_target\_value,

COUNT(\*) AS total\_value

FROM test\_Data

GROUP BY DEFAULTPAYNXTMNT, PREDICTION(Card\_nb\_model USING \*)

ORDER BY 1, 2;

Text

Description automatically generated

Accuracy:

COLUMN ACCURACY FORMAT 99.99

SELECT (SUM(correct)/COUNT(\*))\*100 AS accuracy

FROM (SELECT DECODE(DEFAULTPAYNXTMNT,

PREDICTION(Card\_nb\_model USING \*), 1, 0) AS correct

FROM test\_Data);

Text

Description automatically generated

Prediction top 10:

CREATE OR REPLACE VIEW TOP10\_CC\_Cust\_NB AS

SELECT CARD

FROM (SELECT CARD, rank() over (order by

PREDICTION\_PROBABILITY(Card\_nb\_model, 1 USING \*)

DESC, CARD) RNK

FROM apply\_data

WHERE CREDIT\_LIMIT < 50000)

WHERE RNK <= 10

ORDER BY RNK;

Text

Description automatically generated

REM now we shall display the customers’ names

SELECT CARD

FROM CreditCardsV2

WHERE CARD IN (SELECT \* FROM TOP10\_CC\_Cust\_NB);

Text

Description automatically generated

Decision Tree:

Settings:

REM Create the settings table

CREATE TABLE CC\_decision\_tree\_settings (

setting\_name VARCHAR2(30),

setting\_value VARCHAR2(30));

COMMIT;

Text

Description automatically generated

Populate setting table:

REM Populate the settings table

BEGIN

INSERT INTO CC\_decision\_tree\_settings VALUES

(dbms\_data\_mining.algo\_name,

dbms\_data\_mining.algo\_decision\_tree);

INSERT INTO CC\_decision\_tree\_settings VALUES

(dbms\_data\_mining.prep\_auto,

dbms\_data\_mining.prep\_auto\_on);

COMMIT;

END;

/

Text

Description automatically generated

Model Creation:

REM Create the model using the specified settings

BEGIN

DBMS\_DATA\_MINING.CREATE\_MODEL(

model\_name => 'CC\_decison\_tree',

mining\_function => dbms\_data\_mining.classification,

data\_table\_name => 'Build\_data',

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

target\_column\_name => 'DEFAULTPAYNXTMNT',

settings\_table\_name => 'CC\_decision\_tree\_settings');

END;

/

Text

Description automatically generated

Testing model:

REM testing the model

SELECT DEFAULTPAYNXTMNT AS actual\_target\_value,

PREDICTION(CC\_decison\_tree USING \*) AS predicted\_target\_value,

COUNT(\*) AS total\_value

FROM test\_Data

GROUP BY DEFAULTPAYNXTMNT, PREDICTION(CC\_decison\_tree USING \*)

ORDER BY 1, 2;

Text

Description automatically generated

Accuracy:

REM calculating the models accuracy

COLUMN ACCURACY FORMAT 99.99

SELECT (SUM(correct)/COUNT(\*))\*100 AS accuracy

FROM (SELECT DECODE(DEFAULTPAYNXTMNT,

PREDICTION(CC\_decison\_tree USING \*), 1, 0) AS correct

FROM test\_Data);

Text

Description automatically generated

Predict top 10:

REM first we create the view to identify the customers

CREATE VIEW TOP10\_CC\_Cust\_DT AS

SELECT CARD

FROM (SELECT CARD, rank() over (order by

PREDICTION\_PROBABILITY(CC\_decison\_tree, 1 USING \*)

DESC, CARD) RNK

FROM apply\_data

WHERE CREDIT\_LIMIT < 50000)

WHERE RNK <= 10

ORDER BY RNK;

Graphical user interface, text

Description automatically generated

SELECT CARD

FROM CreditCardsV2

WHERE CARD IN (SELECT \* FROM TOP10\_CC\_Cust\_DT);

Graphical user interface, text

Description automatically generated

Materialized Views:

1D in assessment which require 3 MVs:

Creation :

CREATE MATERIALIZED VIEW promo\_products\_mv

PCTFREE 5

BUILD IMMEDIATE

REFRESH FORCE

ENABLE QUERY REWRITE

AS

SELECT

P.promo\_name,

P.promo\_cost,

P.promo\_begin\_date,

P.promo\_end\_date,

S.prod\_id,

sum(S.quantity\_sold) AS Total\_Quantity,

sum(S.amount\_sold) AS Amount,

T.FISCAL\_YEAR

FROM Sales S,Promotions P,Times T

WHERE S.promo\_id = P.promo\_id

AND S.time\_id=T.time\_id

GROUP BY P.promo\_name,P.promo\_cost,P.promo\_begin\_date,

P.promo\_end\_date,T.FISCAL\_YEAR,S.prod\_id;

Text

Description automatically generated

SELECT DW:

Alter session set query\_rewrite\_integrity = trusted;

Alter session set query\_rewrite\_enabled = TRUE;

EXPLAIN PLAN FOR

SELECT

P.promo\_name,

P.promo\_cost,

P.promo\_begin\_date,

P.promo\_end\_date,

S.prod\_id,

sum(S.quantity\_sold) AS Total\_Quantity,

sum(S.amount\_sold) AS Amount,

T.FISCAL\_YEAR

FROM Sales S,Promotions P,Times T

WHERE S.promo\_id = P.promo\_id

AND S.time\_id=T.time\_id

AND T.FISCAL\_YEAR in ('2001')

GROUP BY P.promo\_name,P.promo\_cost,P.promo\_begin\_date,

P.promo\_end\_date,T.FISCAL\_YEAR,S.prod\_id;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated

Select Query ON SH2 :

Alter session set query\_rewrite\_integrity = trusted;

Alter session set query\_rewrite\_enabled = TRUE;

EXPLAIN PLAN FOR

SELECT

P.promo\_name,

P.promo\_cost,

P.promo\_begin\_date,

P.promo\_end\_date,

S.prod\_id,

sum(S.quantity\_sold) AS Total\_Quantity,

sum(S.amount\_sold) AS Amount,

T.FISCAL\_YEAR

FROM SH2.Sales S,SH2.Promotions P,SH2.Times T

WHERE S.promo\_id = P.promo\_id

AND S.time\_id=T.time\_id

AND T.FISCAL\_YEAR in ('2001')

GROUP BY P.promo\_name,P.promo\_cost,P.promo\_begin\_date,

P.promo\_end\_date,T.FISCAL\_YEAR,S.prod\_id;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated

Creation MV 2 on DWU:

CREATE MATERIALIZED VIEW cust\_regions\_mv

PCTFREE 5

BUILD IMMEDIATE

REFRESH FORCE

ENABLE QUERY REWRITE

AS

SELECT

C.cust\_First\_Name,

C.cust\_Gender,

C.cust\_city,

C.cust\_postal\_code,

sum(S.quantity\_sold) AS Total\_Quantity,

sum(S.amount\_sold) AS Amount,

Co.country\_subregion,

Co.country\_region,

Co.country\_name

FROM Sales S,Customers C,Countries Co

WHERE S.cust\_id =C.cust\_id

AND C.country\_id=Co.country\_id

GROUP BY C.cust\_First\_Name,

C.cust\_Gender,

C.cust\_city,

C.cust\_postal\_code,

Co.country\_subregion,

Co.country\_region,

Co.country\_name;

Text

Description automatically generated

SELECT QUERY:

Alter session set query\_rewrite\_integrity = trusted;

Alter session set query\_rewrite\_enabled = TRUE;

EXPLAIN PLAN FOR

SELECT

C.cust\_First\_Name,

C.cust\_Gender,

C.cust\_city,

C.cust\_postal\_code,

sum(S.quantity\_sold) AS Total\_Quantity,

sum(S.amount\_sold) AS Amount,

Co.country\_subregion,

Co.country\_region,

Co.country\_name

FROM Sales S,Customers C,Countries Co

WHERE S.cust\_id =C.cust\_id

AND C.country\_id=Co.country\_id

AND Co.country\_name='Asia'

GROUP BY C.cust\_First\_Name,

C.cust\_Gender,

C.cust\_city,

C.cust\_postal\_code,

Co.country\_subregion,

Co.country\_region,

Co.country\_name;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

EXPLAN:

Text

Description automatically generated

Text

Description automatically generated

SELECT ON SH2:

Alter session set query\_rewrite\_integrity = trusted;

Alter session set query\_rewrite\_enabled = TRUE;

EXPLAIN PLAN FOR

SELECT

C.cust\_First\_Name,

C.cust\_Gender,

C.cust\_city,

C.cust\_postal\_code,

sum(S.quantity\_sold) AS Total\_Quantity,

sum(S.amount\_sold) AS Amount,

Co.country\_subregion,

Co.country\_region,

Co.country\_name

FROM SH2.Sales S,SH2.Customers C,SH2.Countries Co

WHERE S.cust\_id =C.cust\_id

AND C.country\_id=Co.country\_id

AND Co.country\_name='Asia'

GROUP BY C.cust\_First\_Name,

C.cust\_Gender,

C.cust\_city,

C.cust\_postal\_code,

Co.country\_subregion,

Co.country\_region,

Co.country\_name;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated

Cretation of MV3:

CREATE MATERIALIZED VIEW prod\_categ\_mv

PCTFREE 5

BUILD IMMEDIATE

REFRESH FORCE

ENABLE QUERY REWRITE

AS

SELECT

P.prod\_name,

P.prod\_category,

P.prod\_unit\_of\_measure,

P.prod\_pack\_size,

sum(C.unit\_cost) AS unit\_Cost,

sum(C.unit\_price) AS Unit\_Price,

T.calendar\_year,

T.end\_of\_fis\_year

FROM Costs C,Times T,Products P

WHERE C.prod\_id =P.prod\_id

AND C.time\_id=T.time\_id

GROUP BY P.prod\_name,

P.prod\_category,

P.prod\_unit\_of\_measure,

P.prod\_pack\_size,T.calendar\_year,

T.end\_of\_fis\_year;

Text

Description automatically generated

SEELCT QUERY ON DW:

Alter session set query\_rewrite\_integrity = trusted;

Alter session set query\_rewrite\_enabled = TRUE;

EXPLAIN PLAN FOR

SELECT

P.prod\_name,

P.prod\_category,

P.prod\_unit\_of\_measure,

P.prod\_pack\_size,

sum(C.unit\_cost) AS unit\_Cost,

sum(C.unit\_price) AS Unit\_Price,

T.calendar\_year,

T.end\_of\_fis\_year

FROM Costs C,Times T,Products P

WHERE C.prod\_id =P.prod\_id

AND C.time\_id=T.time\_id

AND T.end\_of\_fis\_year='30-DEC-01'

GROUP BY P.prod\_name,

P.prod\_category,

P.prod\_unit\_of\_measure,

P.prod\_pack\_size,T.calendar\_year,

T.end\_of\_fis\_year;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated

Select on SH2:

Alter session set query\_rewrite\_integrity = trusted;

Alter session set query\_rewrite\_enabled = TRUE;

EXPLAIN PLAN FOR

SELECT

P.prod\_name,

P.prod\_category,

P.prod\_unit\_of\_measure,

P.prod\_pack\_size,

sum(C.unit\_cost) AS unit\_Cost,

sum(C.unit\_price) AS Unit\_Price,

T.calendar\_year,

T.end\_of\_fis\_year

FROM Sh2.Costs C,SH2.Times T,SH2.Products P

WHERE C.prod\_id =P.prod\_id

AND C.time\_id=T.time\_id

AND T.end\_of\_fis\_year='30-DEC-01'

GROUP BY P.prod\_name,

P.prod\_category,

P.prod\_unit\_of\_measure,

P.prod\_pack\_size,T.calendar\_year,

T.end\_of\_fis\_year;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated

Materialized Views 1C:

1. Select Query SH2:

alter session set query\_rewrite\_integrity = TRUSTED;

alter session set query\_rewrite\_enabled = TRUE;

set echo on

set timing on

EXPLAIN PLAN FOR

SELECT t.calendar\_month\_desc

, sum(s.amount\_sold) AS Money

FROM sh2.sales s

, sh2.times t

WHERE s.time\_id = t.time\_id

AND t.calendar\_month\_desc in ('2003-12','2002-08','2004-01')

GROUP BY t.calendar\_month\_desc;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated

DWU:

alter session set query\_rewrite\_integrity = TRUSTED;

alter session set query\_rewrite\_enabled = TRUE;

set echo on

set timing on

EXPLAIN PLAN FOR

SELECT t.calendar\_month\_desc

, sum(s.amount\_sold) AS Money

FROM sales s

, times t

WHERE s.time\_id = t.time\_id

AND t.calendar\_month\_desc in ('2003-12','2002-08','2004-01')

GROUP BY t.calendar\_month\_desc;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated

1. SELECT Query on MV on SH2:

alter session set query\_rewrite\_integrity = TRUSTED;

alter session set query\_rewrite\_enabled = TRUE;

set echo on

set timing on

EXPLAIN PLAN FOR

SELECT t.week\_ending\_day

, p.prod\_subcategory

, sum(s.amount\_sold) AS Money

, s.channel\_id

, s.promo\_id

FROM SH2.sales s

, SH2.times t

, SH2.products p

WHERE s.time\_id = t.time\_id

AND s.prod\_id = p.prod\_id

AND p.prod\_subcategory in ('Shoes - Boys','Sweaters - Men')

GROUP BY t.week\_ending\_day

, p.prod\_subcategory

, s.channel\_id

, s.promo\_id;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

DWU :

alter session set query\_rewrite\_integrity = TRUSTED;

alter session set query\_rewrite\_enabled = TRUE;

set echo on

set timing on

EXPLAIN PLAN FOR

SELECT t.week\_ending\_day

, p.prod\_subcategory

, sum(s.amount\_sold) AS Money

, s.channel\_id

, s.promo\_id

FROM sales s

, times t

, products p

WHERE s.time\_id = t.time\_id

AND s.prod\_id = p.prod\_id

AND p.prod\_subcategory in ('Shoes - Boys','Sweaters - Men')

GROUP BY t.week\_ending\_day

, p.prod\_subcategory

, s.channel\_id

, s.promo\_id;

set linesize 200

set pagesize 50

set markup html preformat on

select \* from table(dbms\_xplan.display());

set linesize 80

Text

Description automatically generated

Text

Description automatically generated with medium confidence