

MACHINE LEARNING

Q1. The residual sum of squares (RSS) measures the level of variance in the error term, or residuals, of a regression model. The smaller the residual sum of squares, the better your model fits your data; the greater the residual sum of squares, the poorer your model fits your data. R-Squared (R^2 or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).

Q2. The TSS (TSS or SST) tells you how much variation there is in the dependent variable.

The ESS tells you how much of the variation in the dependent variable your model explained.

The residual sum of squares tells you how much of the dependent variable's variation your model did not explain. It is the sum of the squared differences between the actual Y and the predicted Y:

$$\text{Residual Sum of Squares} = \sum e^2$$

In ANOVA, Total SS is related to the total sum and explained sum with the following formula:
Total SS = Explained SS + Residual Sum of Squares.

Q3. Regularization is one of the most important concepts of machine learning. This technique prevents the model from overfitting by adding extra information to it.

Q4. Gini impurity, also known as Gini Index, calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly. If all the elements are linked with a single class then it can be called pure.

Q5. Decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions. This small sample could lead to unsound conclusions.

Q6. Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. 'Audience poll' is an example of Ensemble Technique.

Q7. Bagging aims to decrease variance, not bias while Boosting aims to decrease bias, not variance. In Bagging each model is built independently whereas in Boosting new models are influenced by performance of previously built models. Bagging tries to solve over-fitting problem while Boosting tries to reduce bias.

Q8. Out-of-bag (OOB) error, also called out-of-bag estimate, is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating.

Q9. K-fold Cross-Validation is when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data. K refers to the number of groups the data

sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation

Q10. A hyperparameter is a parameter whose value is set before the machine learning process begins. In contrast, the values of other parameters are derived via training. Algorithm hyperparameters affect the speed and quality of the learning process.

Q11. A learning rate that is too large can cause the model to converge too quickly to a suboptimal solution, whereas a learning rate that is too small can cause the process to get stuck.

Q12. No, logistic regression only forms linear decision surface, but the examples in the figure are not linearly separable

Q13. AdaBoost is the first designed boosting algorithm with a particular loss function. On the other hand, Gradient Boosting is a generic algorithm that assists in searching the approximate solutions to the additive modelling problem. This makes Gradient Boosting more flexible than AdaBoost.

Q14. If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

Q15. Gaussian radial basis function - used when there is no prior knowledge about the data. Polynomial kernel -It is popular in image processing. Linear is useful when dealing with large sparse data vectors. It is often used in text categorization.

STATISTICS WORKSHEET-5

Q1. D

Q2. C

Q3. C

Q4. B

Q5. C

Q6. B

Q7. A

Q8. A

Q9. B

Q10. A

WORKSHEET 5 SQL

Refer the following ERD and answer all the questions in this worksheet. You have to write the queries using MySQL for the required Operation.

1. Write SQL query to show all the data in the Movie table.

```
DROP TABLE IF EXISTS movie;
```

```
CREATE TABLE movie (
```

```
movie_id INT(30) NOT NULL,
```

```
title VARCHAR(1000) DEFAULT NULL,
```

```
budget INT(30) DEFAULT NULL,
```

```
homepage VARCHAR(5000) DEFAULT NULL,
```

```
overview VARCHAR(5000) DEFAULT NULL,
```

```
popularity DECIMAL(12,6) DEFAULT NULL,
```

```
release_date DATE DEFAULT NULL,
```

```
revenue INT(20) DEFAULT NULL,
```

```
runtime INT(10) DEFAULT NULL,
```

```
movie_status VARCHAR(50) DEFAULT NULL,
```

```
tagline VARCHAR(5000) DEFAULT NULL,
```

```
vote_average DECIMAL(4,2) DEFAULT NULL,
```

```
vote_count INT(10) DEFAULT NULL,
```

```
CONSTRAINT pk_movie PRIMARY KEY (movie_id)
```

```
);
```

2. Write SQL query to show the title of the longest runtime movie.

```
Select MySQL 8.0 Command Line Client

728 rows in set (0.01 sec)

mysql> select MAX(runtime) from movie;
ERROR 1630 (42000): FUNCTION movies.MAX does not exist. Check the 'Function Name Parsing and Resolution' section in the Reference Manual
mysql> SELECT MAX(runtime) AS "Longest Runtime Movie" FROM movie;
+-----+
| Longest Runtime Movie |
+-----+
|          238          |
+-----+
1 row in set (0.13 sec)

mysql> SELECT title, runtime FROM movie WHERE runtime = (SELECT MAX(runtime) FROM Movie);
+-----+-----+
| title          | runtime |
+-----+-----+
| Gone with the Wind |      238 |
+-----+-----+
1 row in set (0.03 sec)

mysql>
```

Q3. Write SQL query to show the highest revenue generating movie title.

```
Select MySQL 8.0 Command Line Client

S `person` ('person_id')
mysql> SELECT title, revenue FROM movie WHERE revenue = (SELECT MAX(revenue) FROM movie);
ERROR 1630 (42000): FUNCTION movies.MAX does not exist. Check the 'Function Name Parsing and Resolution' section in the Reference Manual
mysql> select MAX(revenue) FROM movie;
+-----+
| MAX(revenue) |
+-----+
| 1845034188   |
+-----+
1 row in set (0.01 sec)

mysql> select title, MAX(revenue) FROM movie;
+-----+-----+
| title          | MAX(revenue) |
+-----+-----+
| Four Rooms    | 1845034188   |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Q4. Write SQL query to show the movie title with maximum value of revenue/budget.

```
MySQL 8.0 Command Line Client

mysql> SELECT revenue,title,MAX(budget) FROM movie;
+-----+-----+-----+
| revenue | title      | MAX(budget) |
+-----+-----+-----+
| 4300000 | Four Rooms | 3800000000  |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

5. Write a SQL query to show the movie title and its cast details like name of the person, gender, character name, cast order.

```
Select MySQL 8.0 Command Line Client

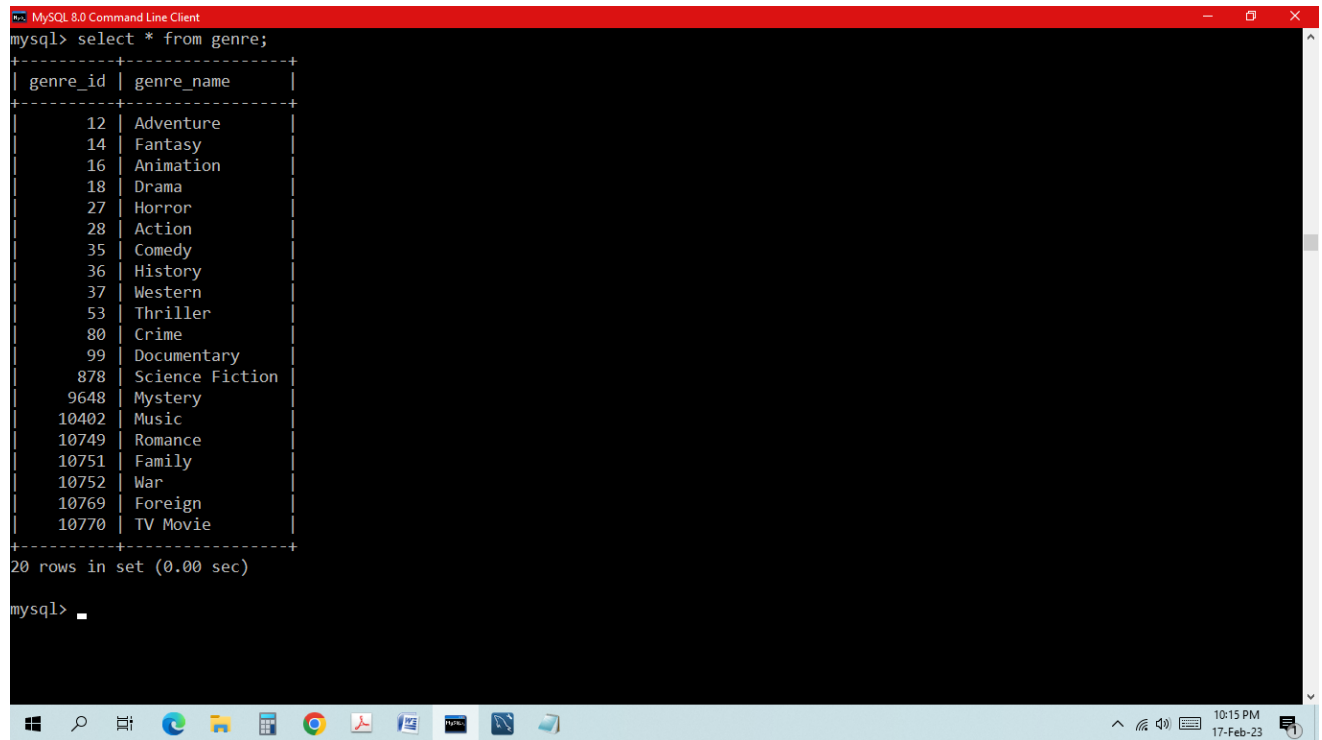
country          | BASE TABLE |
department       | BASE TABLE |
gender           | BASE TABLE |
genre            | BASE TABLE |
keyword          | BASE TABLE |
language         | BASE TABLE |
language_role    | BASE TABLE |
movie            | BASE TABLE |
movie_cast       | BASE TABLE |
movie_crew       | BASE TABLE |
movie_genre      | BASE TABLE |
movie_keywords   | BASE TABLE |
movie_languages  | BASE TABLE |
person           | BASE TABLE |
production_company | BASE TABLE |
+-----+-----+
15 rows in set (0.01 sec)

mysql> SELECT movie.title, person.person_name, gender.gender, movie_cast.character_name, movie_cast.cast_order
-> FROM movie
-> JOIN movie_cast
-> ON movie.movie_id = movie_cast.movie_id
-> JOIN gender
-> ON movie_cast.gender_id = gender.gender_id
-> JOIN person
-> ON movie_cast.person_id = person.person_id;
+-----+-----+-----+-----+-----+
| title                                     | person_name | gender | character_name | cast_order |
+-----+-----+-----+-----+-----+
| Pirates of the Caribbean: At World's End | Johnny Depp | Male   | Captain Jack Sparrow | 0          |
| Pirates of the Caribbean: At World's End | Orlando Bloom | Male   | Will Turner         | 1          |
| Pirates of the Caribbean: At World's End | Johnny Depp | Male   | Captain Jack Sparrow | 0          |
| Pirates of the Caribbean: At World's End | Orlando Bloom | Male   | Will Turner         | 1          |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

6. Write a SQL query to show the country name where maximum number of movies has been produced, along with the number of movies produced.

7. Write a SQL query to show all the genre_id in one column and genre_name in second column.



```
mysql> select * from genre;
```

genre_id	genre_name
12	Adventure
14	Fantasy
16	Animation
18	Drama
27	Horror
28	Action
35	Comedy
36	History
37	Western
53	Thriller
80	Crime
99	Documentary
878	Science Fiction
9648	Mystery
10402	Music
10749	Romance
10751	Family
10752	War
10769	Foreign
10770	TV Movie

```
20 rows in set (0.00 sec)
```

```
mysql>
```

8. Write a SQL query to show name of all the languages in one column and number of movies in that particular column in another column.

9. Write a SQL query to show movie name in first column, no. of crew members in second column and number of cast members in third column.

10. Write a SQL query to list top 10 movies title according to popularity column in decreasing order.

```
Select MySQL 8.0 Command Line Client
mysql> SELECT title, popularity FROM movie WHERE popularity > 128 ORDER BY popularity desc;
+-----+-----+
| title                                     | popularity |
+-----+-----+
| Pirates of the Caribbean: The Curse of the Black Pearl | 271.972889 |
| The Dark Knight                                     | 187.322927 |
| Fight Club                                           | 146.757391 |
| Pirates of the Caribbean: Dead Man's Chest           | 145.847379 |
| The Godfather                                        | 143.659698 |
| Pirates of the Caribbean: At World's End             | 139.082615 |
| The Lord of the Rings: The Fellowship of the Ring    | 138.049577 |
| The Shawshank Redemption                            | 136.747729 |
| Pirates of the Caribbean: On Stranger Tides           | 135.413856 |
| Harry Potter and the Chamber of Secrets              | 132.397737 |
+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

11. Write a SQL query to show the name of the 3rd most revenue generating movie and its revenue.

```
Select MySQL 8.0 Command Line Client

mysql> SELECT title, revenue FROM movie WHERE revenue >=1000000000;
+-----+-----+
| title                                     | revenue |
+-----+-----+
| Pirates of the Caribbean: Dead Man's Chest | 1065659812 |
| The Lord of the Rings: The Return of the King | 1118888979 |
| The Dark Knight                             | 1004558444 |
| Titanic                                     | 1845034188 |
| Pirates of the Caribbean: On Stranger Tides | 1045713802 |
+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

12. Write a SQL query to show the names of all the movies which have “rumoured” movie status.

```
Select MySQL 8.0 Command Line Client

mysql> SELECT * FROM movie WHERE movie_status='rumour';
Empty set (0.01 sec)

mysql>
```

13. Write a SQL query to show the name of the “United States of America” produced movie which generated maximum revenue.

```
MySQL 8.0 Command Line Client
mysql> SELECT title AS United_States_of_America,revenue AS revenue FROM movie WHERE revenue>1845000000
-> ORDER BY revenue;
+-----+-----+
| United_States_of_America | revenue |
+-----+-----+
| Titanic                  | 1845034188 |
+-----+-----+
1 row in set (0.01 sec)

mysql>
```

14. Write a SQL query to print the movie_id in one column and name of the production company in the second column for all the movies.

```
Select MySQL 8.0 Command Line Client
mysql> SELECT movie_id FROM movie_company
-> INNER JOIN production_company
-> ON movie_company.movie_id = production_company.company_name;
Empty set (0.03 sec)

mysql>
```

15. Write a SQL query to show the title of top 20 movies arranged in decreasing order of their budget.

```
Select MySQL 8.0 Command Line Client
mysql> SELECT title AS movie_name,budget FROM movie WHERE budget>160000000
-> ORDER BY budget DESC;
+-----+-----+
| movie_name | budget |
+-----+-----+
| Pirates of the Caribbean: On Stranger Tides | 380000000 |
| Pirates of the Caribbean: At World's End | 300000000 |
| Superman Returns | 270000000 |
| Spider-Man 3 | 258000000 |
| Harry Potter and the Half-Blood Prince | 250000000 |
| The Chronicles of Narnia: Prince Caspian | 225000000 |
| The Amazing Spider-Man | 215000000 |
| King Kong | 207000000 |
| Pirates of the Caribbean: Dead Man's Chest | 200000000 |
| Terminator 3: Rise of the Machines | 200000000 |
| Terminator Salvation | 200000000 |
| Spider-Man 2 | 200000000 |
| Titanic | 200000000 |
| The Dark Knight | 185000000 |
| Indiana Jones and the Kingdom of the Crystal Skull | 185000000 |
| The Chronicles of Narnia: The Lion, the Witch and the Wardrobe | 180000000 |
| The Golden Compass | 180000000 |
| Troy | 175000000 |
+-----+-----+
18 rows in set (0.01 sec)

mysql>
```