SQL QUERIES ON FAKE NEWS PREDICTION:-

1.Get the number of fake and real news:

```
sql
SELECT SUM(CASE WHEN l.label = 'FAKE' THEN 1 ELSE 0 END) AS fake_count,
       SUM(CASE WHEN l.label = 'REAL' THEN 1 ELSE 0 END) AS real_count
FROM dataset d
JOIN labels l ON d.id = l.id;
```

2.Get the top 10 fake news titles:

```
sql
SELECT title, COUNT(*) AS num_fake
FROM dataset d
JOIN labels l ON d.id = l.id
WHERE l.label = 'FAKE'
GROUP BY title
ORDER BY num_fake DESC
LIMIT 10;
```

3. Get the average length of fake and real news:

```
sql
SELECT CASE WHEN l.label = 'FAKE' THEN 'Fake' ELSE 'Real' END AS label,
        AVG(LENGTH(text)) AS avg_length
FROM dataset d
JOIN labels l ON d.id = l.id
GROUP BY l.label;

ORDER BY num_fake DESC

LIMIT 10;

4.Top 10 most popular real news titles:

sql
SELECT title, COUNT(*) AS num_real
FROM dataset d
JOIN labels l ON d.id = l.id
WHERE l.label = 'REAL'
GROUP BY title
ORDER BY num_real DESC
LIMIT 10;

5.Top 10 most popular fake news authors:

sql
SELECT author, COUNT(*) AS num_fake
FROM (
  SELECT SUBSTR(title, 1, INSTR(title, ' ')-1) AS author, id
```

```
  FROM dataset d
  JOIN labels l ON d.id = l.id
  WHERE l.label = 'FAKE'
  GROUP BY author
) t
GROUP BY author
ORDER BY num_fake DESC
LIMIT 10;
```

6.Top 10 most popular real news authors:

```
sql
SELECT author, COUNT(*) AS num_real
FROM (
  SELECT SUBSTR(title, 1, INSTR(title, ' ')-1) AS author, id
  FROM dataset d
  JOIN labels l ON d.id = l.id
  WHERE l.label = 'REAL'
  GROUP BY author
) t
GROUP BY author
ORDER BY num_real DESC
LIMIT 10;
```

7.Average length of fake and real news:

```
sql
SELECT CASE WHEN l.label = 'FAKE' THEN 'Fake' ELSE 'Real' END AS label,
        AVG(LENGTH(text)) AS avg_length
FROM dataset d
JOIN labels l ON d.id = l.id
GROUP BY l.label;
```

8.Average number of words in fake and real news:

```
sql
SELECT CASE WHEN l.label = 'FAKE' THEN 'Fake' ELSE 'Real' END AS label,
        AVG(LENGTH(text) - LENGTH(REPLACE(text, ' ', ''))) + 1 AS avg_words
FROM dataset d
JOIN labels l ON d.id = l.id
GROUP BY l.label;
```

9.Most common words in fake and real news:

```
sql
WITH word_counts AS (
  SELECT SUBSTRING(text, 1, INSTR(text, ' ', 1, 1) - 1) AS word,
         COUNT(*) AS word_count,
         CASE WHEN l.label = 'FAKE' THEN 'Fake' ELSE 'Real' END AS label
  FROM dataset d
```

```
  JOIN labels l ON d.id = l.id
  CROSS JOIN UNNEST(SPLIT_PART(text, ' ', 1)) AS word
  GROUP BY word, label
  ORDER BY word_count DESC
  LIMIT 10
)
SELECT word, SUM(word_count) AS total_count, label
FROM word_counts
GROUP BY word, label
ORDER BY total_count DESC;
```

10.Most common phrases in fake and real news:

```
sql
WITH phrase_counts AS (
  SELECT SUBSTRING(text, 1, INSTR(text, ' ', 1, 1) - 1) || ' ' ||
          SUBSTRING(text, INSTR(text, ' ', 1, 2), INSTR(text, ' ', 1, 2, -1) -
INSTR(text, ' ', 1, 1)) AS phrase,
         COUNT(*) AS phrase_count,
         CASE WHEN l.label = 'FAKE' THEN 'Fake' ELSE 'Real' END AS label
  FROM dataset d
  JOIN labels l ON d.id = l.id
  CROSS JOIN UNNEST(SPLIT_PART(text, ' ', 1)) AS word1,
               UNNEST(SPLIT_PART(text, ' ', 2, INFINITY)) AS word2
  WHERE word1 <> word2
  GROUP BY phrase, label
  ORDER BY phrase_count DESC
  LIMIT 10
)
SELECT phrase, SUM(phrase_count) AS total_count, label
FROM phrase_counts
GROUP BY phrase, label
ORDER BY total_count DESC;
```

11.Most common sources in fake and real news:

```
sql
WITH source_counts AS (
  SELECT SUBSTR(title, INSTR(title, ' ', -1) + 1) AS source,
         COUNT(*) AS source_count,
         CASE WHEN l.label = 'FAKE' THEN 'Fake' ELSE 'Real' END AS label
  FROM dataset d
  JOIN labels l ON d.id = l.id
  WHERE INSTR(title, ' ', -1) > 0
  GROUP BY source, label
  ORDER BY source_count DESC
  LIMIT 10
)
SELECT source, SUM(source_count) AS total_count, label
FROM source_counts
```

```
GROUP BY source, label
ORDER BY total_count DESC;

12.Most common domains in fake and real news:

sql
WITH domain_counts AS (
  SELECT SUBSTR(source, INSTR(source, '.', 1, 1) + 1) AS domain,
         COUNT(*) AS domain_count,
         CASE WHEN l.label = 'FAKE' THEN 'Fake' ELSE 'Real' END AS label
  FROM (
    SELECT SUBSTR(title, INSTR(title, ' ', -1) + 1) AS source
    FROM dataset d
    JOIN labels l ON d.id = l.id
    WHERE INSTR(title, ' ', -1) > 0
  ) t
  CROSS JOIN UNNEST(SPLIT_PART(source,
```