Can you predict the future? No? Well, Snowflake kinda can! How? Well it's those Cortex functions that everyone keeps going on about.

This week we'll be using one of their ML functions.

Your job is to get the training data:

```
CREATE STAGE IF NOT EXISTS frosty_aws_stage
  URL = 's3://frostyfridaychallenges/';

CREATE OR REPLACE TABLE WEEK_90 AS
SELECT
    $1::TIMESTAMP_NTZ AS SALE_DATE,
    $2::INT AS PRODUCT_ID,
    $3::INT AS QUANTITY_SOLD,
    $4::INT AS UNIT_PRICE,
    $5/100::FLOAT AS TAX_PCT,
    $6/100::FLOAT AS DCT_PCT
FROM @FROSTY_AWS_STAGE/challenge_90
WHERE $1 != 'Date';
```

And see what Snowflake has to say forecasting-wise about the below:

```
CREATE OR REPLACE TABLE WEEK_90_F LIKE WEEK_90;
ALTER TABLE WEEK_90_F DROP COLUMN QUANTITY_SOLD;

INSERT INTO WEEK_90_F VALUES
(TO_TIMESTAMP_NTZ('2023-10-29'), 1000, 450, 0.1, 0.02),
(TO_TIMESTAMP_NTZ('2023-10-29'), 1001, 150, 0.15, 0.02),
(TO_TIMESTAMP_NTZ('2023-10-29'), 1002, 100, 0.13, 0.18),
(TO_TIMESTAMP_NTZ('2023-10-29'), 1003, 170, 0.11, 0.03),
(TO_TIMESTAMP_NTZ('2023-10-29'), 1004, 300, 0.04, 0.03);
```

Your output should look like the below:

	SERIES	TS	FORECAST	LOWER_BOUND	UPPER_BOUND
1	1000	2023-10-28 00:00:00.000	56.38741879	-3.843978585	116.618816164
2	1000	2023-10-29 00:00:00.000	42.005507932	-19.397366722	103.408382586
3	1001	2023-10-28 00:00:00.000	40.854282704	-13.799616407	95.508181814
4	1001	2023-10-29 00:00:00.000	37.218238605	-19.396168136	93.832645347
5	1002	2023-10-28 00:00:00.000	72.843189694	25.738131611	119.948247776
6	1002	2023-10-29 00:00:00.000	77.422581353	30.31752327	124.527639435
7	1003	2023-10-28 00:00:00.000	67.458301086	13.124158423	121.792443749
8	1003	2023-10-29 00:00:00.000	50.74008334	-3.83035573	105.31052241
9	1004	2023-10-28 00:00:00.000	57.648129693	12.88672955	102.409529835
10	1004	2023-10-29 00:00:00.000	54.832124331	7.579302471	102.084946191

Happy forecasting!