

Snowfall Predictions: *Northern and Southern Hemispheres

Data Analytics
University of Oregon

Haley Monteblando
Caed Turner
Trevor Crane



Background

Questions

- Total snowfall per year
- Maximum powder day year
- Minimum month
- Maximum month
- Snowfall trends by region
- Snowfall predictions

Technologies Used

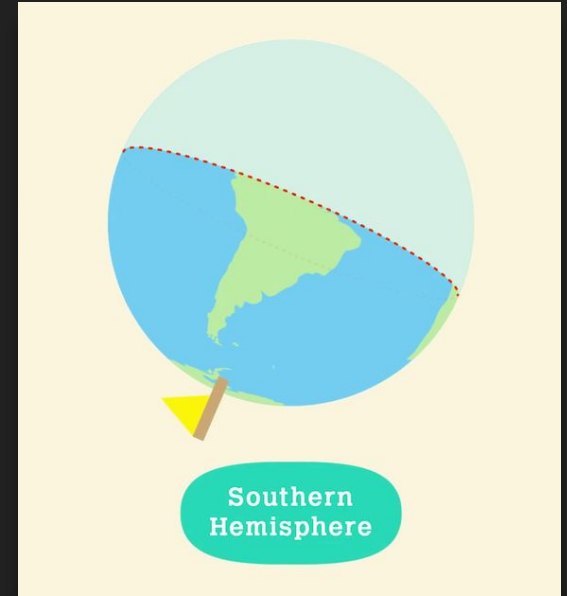
- Code:
 - Jupyter Notebook
 - Python
 - Pandas
 - SQLAlchemy
 - Scikit-learn
 - Path
 - Pickle
- Storage/Analysis:
 - PostgreSQL
 - PGAdmin
- Visualization:
 - Tableau
 - Python
 - HTML



Data ➡ **Python** ➡ **Database** ➡ **Model**

Southern Hemisphere

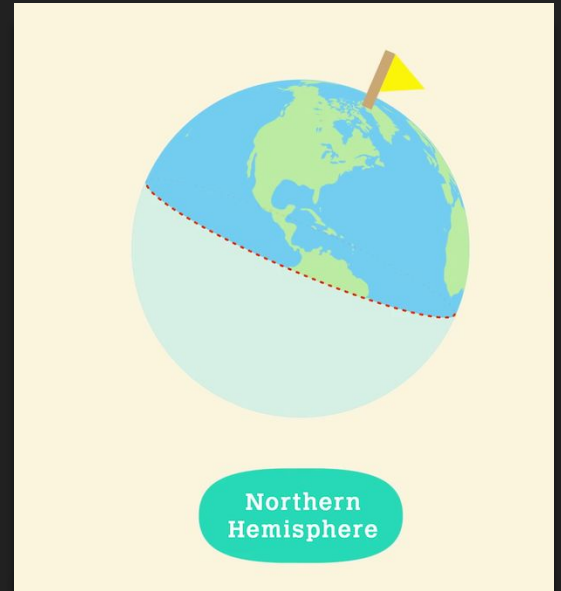
- Craigieburn Valley
- Tiffindell
- Valle Nevado



hemisphere
southern

Northern Hemisphere

- Furano
- Zermatt
- Timberline



Northern
Hemisphere

Cleaning

- Starting Columns
- Final Columns
- Resample Data
- Reindex

RangeIndex: 287592 entries, 0 to 287591

Data columns (total 10 columns):

#	Column	Non-Null Count	Dtype
0	dt	287592 non-null	int64
1	dt_iso	287592 non-null	object
2	city_name	287592 non-null	object
3	temp	287592 non-null	float64
4	temp_min	287592 non-null	float64
5	temp_max	287592 non-null	float64
6	humidity	287592 non-null	int64
7	snow_1h	287592 non-null	float64
8	weather_main	287592 non-null	object
9	weather_description	287592 non-null	object

dtypes: float64(4), int64(2), object(4)

memory usage: 21.9+ MB

```
[ ]:      temp  temp_min  temp_max  humidity  snowfall
      dt
1990-01-01  58.55      58.55      58.55      74.33      0.0
1990-01-02  60.58      60.58      60.58      76.46      0.0
1990-01-03  61.66      61.66      61.66      79.12      0.0
1990-01-04  62.28      62.28      62.28      69.67      0.0
1990-01-05  64.27      64.27      64.27      60.50      0.0
...         ...         ...         ...         ...         ...
2022-10-18  57.24      57.24      57.24      64.67      0.0
2022-10-19  59.89      59.89      59.89      59.12      0.0
2022-10-20  58.99      58.99      58.99      68.83      0.0
2022-10-21  58.43      58.43      58.43      71.21      0.0
2022-10-22  49.58      49.58      49.58      92.96      0.0
```

11983 rows × 5 columns

Export CSV to SQL

- Export to csv
- Db_string used to connect to postgres
- Create a database engine to connect to postgres
- Push CSV to postgres

```
# Export to CSV  
finished_df.to_csv('./cleaned_Furano_Ski_Resort_df.csv')
```

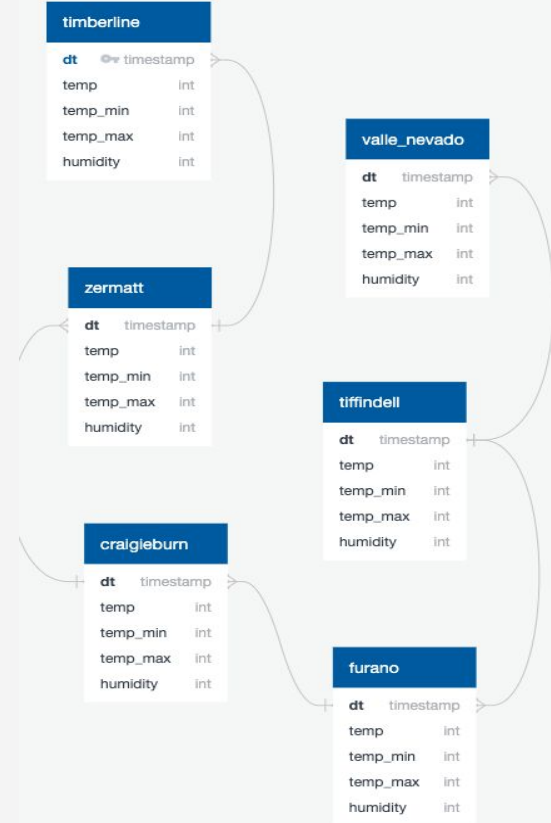
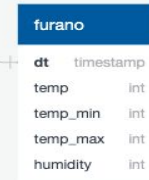
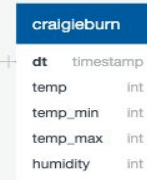
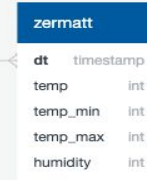
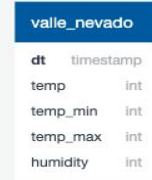
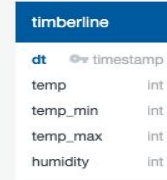
```
# db_string used to connect to postgres  
db_string = f"postgresql://postgres:{db_password}@127.0.0.1:5432/snow_data"
```

```
# create a Database engine to connect to postgres using the db_string  
engine = create_engine(db_string)
```

```
finished_df.to_sql(name='Furano_Ski_Resort', con=engine)
```

Entity Relationship Diagram (ERD)

www.quickdatabasediagrams.com



SQL Database

- Max snowfall per year
- Average snowfall per year

```
1  -- Create table for max snowfall in a single day per each year for timberline
2  CREATE TABLE timberline_max_per_year AS (
3      SELECT
4          EXTRACT(year FROM dt) AS year,
5          MAX(snowfall) AS max_snowfall
6      FROM timberline
7      GROUP BY year
8      ORDER BY year ASC);
9
10 -- Create table for timberline that only has days where snowfall is > 0
11 CREATE TABLE timberline_snow_days AS (
12     SELECT *
13     FROM timberline WHERE
14     snowfall > 0);
15
16 -- Create table showing average snowfall for all days that experience snow that year timberline
17 CREATE TABLE timberline_avg_per_year AS (
18     SELECT
19         EXTRACT(year FROM dt) AS year,
20         AVG(snowfall) AS avg_snowfall
21     FROM timberline_snow_days
22     GROUP BY year
23     ORDER BY year ASC);
```

Queries

Max and Min Pow days (6+ inches)

```
--Create table Timberline max pow days
CREATE TABLE timberline_pow_day_max AS
(SELECT
COUNT(EXTRACT(year FROM dt)),
  EXTRACT(year FROM dt) AS year
FROM timberline WHERE
snowfall >= 6
GROUP BY EXTRACT(year FROM dt));
```

```
--Create table Timberline min pow days
CREATE TABLE timberline_pow_day_min AS
(SELECT
COUNT(EXTRACT(year FROM dt)),
  EXTRACT(year FROM dt) AS year
FROM timberline WHERE
snowfall <= 6
GROUP BY EXTRACT(year FROM dt));
```


Queries

- Snowfall per month
- Max month snowfall

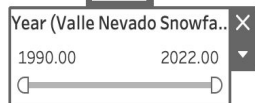
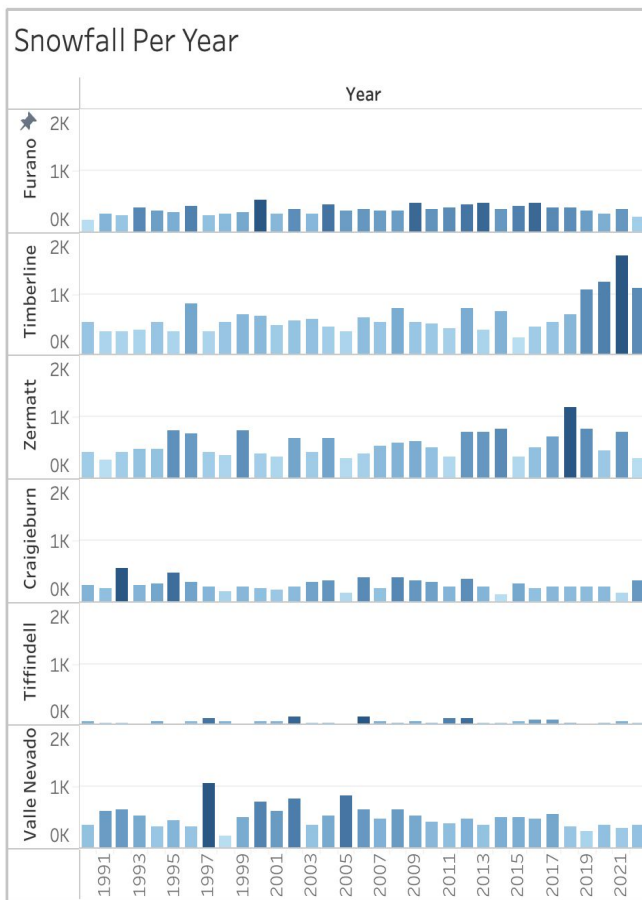
```
13 -- Finding and creating table for total snowfall in a month per each year in timberline
14 CREATE TABLE timberline_snowfall_per_month AS (
15 SELECT
16     EXTRACT(year FROM dt) AS year,
17     EXTRACT(month FROM dt) AS month,
18     SUM(snowfall) AS total_snowfall
19 FROM timberline
20 GROUP BY year,month
21 ORDER BY year ASC);
22
23 SELECT * FROM timberline_snowfall_per_month;
24
25 -- Finding and creating table for which months had the most snowfall in timberline
26 CREATE TABLE timberline_max_months_snowfall AS (
27 SELECT
28     EXTRACT(year FROM dt) AS year,
29     EXTRACT(month FROM dt) AS month,
30     SUM(snowfall) AS total_snowfall
31 FROM timberline
32 GROUP BY year,month
33 HAVING SUM(snowfall)>0
34 ORDER BY total_snowfall DESC);
35
36 SELECT * FROM timberline_max_months_snowfall;
```

Queries

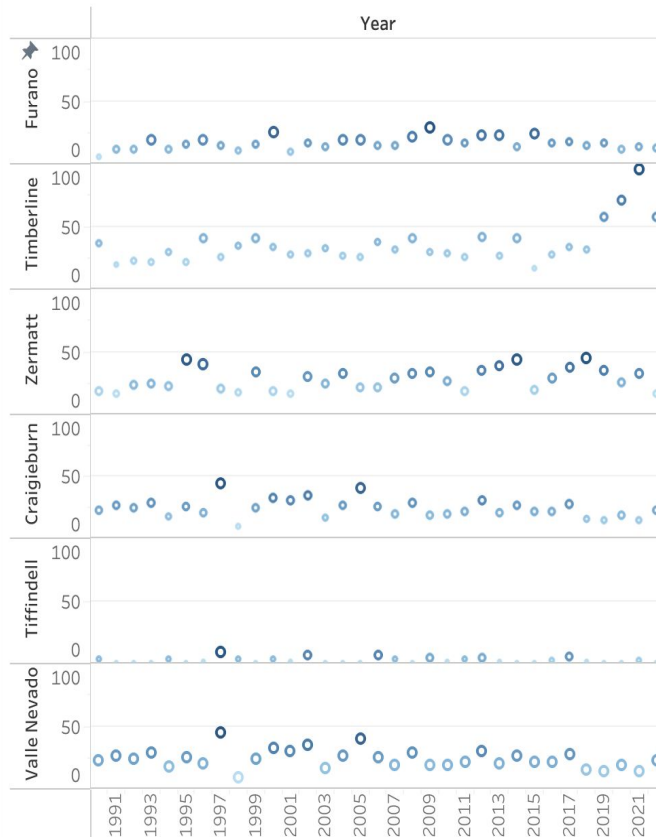
- Min month snowfall
- Snowfall month

```
38 -- Finding and creating which months had the the least snowfall but > 0
39 CREATE TABLE timberline_min_months_snowfall AS (
40 SELECT
41     EXTRACT(year FROM dt) AS year,
42     EXTRACT(month FROM dt) AS month,
43     SUM(snowfall) AS total_snowfall
44 FROM timberline
45 GROUP BY year,month
46 HAVING SUM(snowfall)>0
47 ORDER BY total_snowfall);
48
49 SELECT * FROM timberline_min_months_snowfall;
50
51 -- Finding Out and creating tables for which months in timberline had snowfall
52 CREATE TABLE snowfall_months_timberline AS(
53 SELECT
54     EXTRACT(year FROM dt) AS year,
55     EXTRACT(month FROM dt) AS month,
56     SUM(snowfall) AS total_snowfall
57 FROM timberline
58 GROUP BY year,month
59 HAVING SUM(snowfall)>0
60 ORDER BY year);
61
62 SELECT * FROM snowfall_months_timberline;
```

Dashboard 1

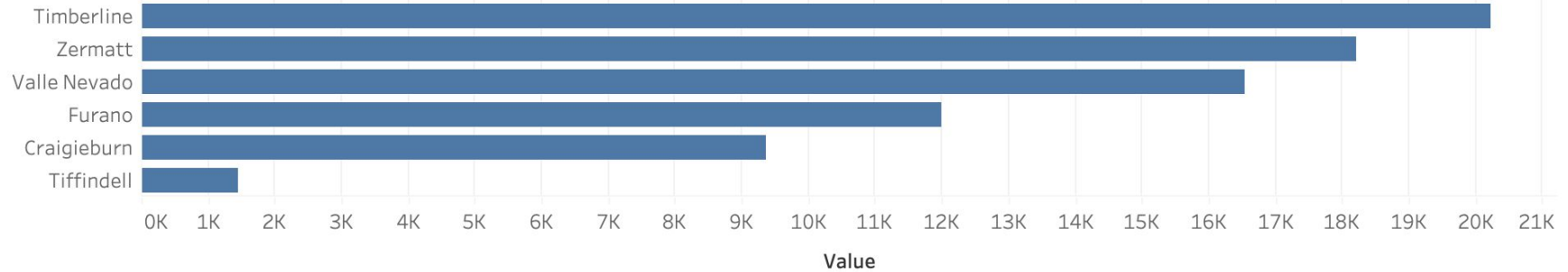


Pow Days (6+ inches) Per Year



Dashboard 2

Total Snowfall since 1990 (as inches)



Deepest Snowfall (in a single day) Per Each Year

Dashboard 3

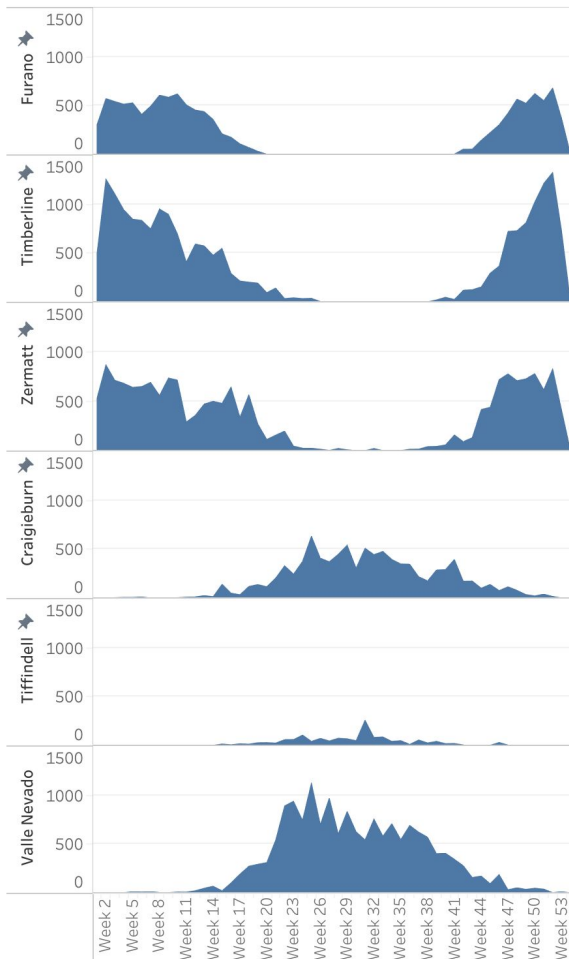


Deepest Day

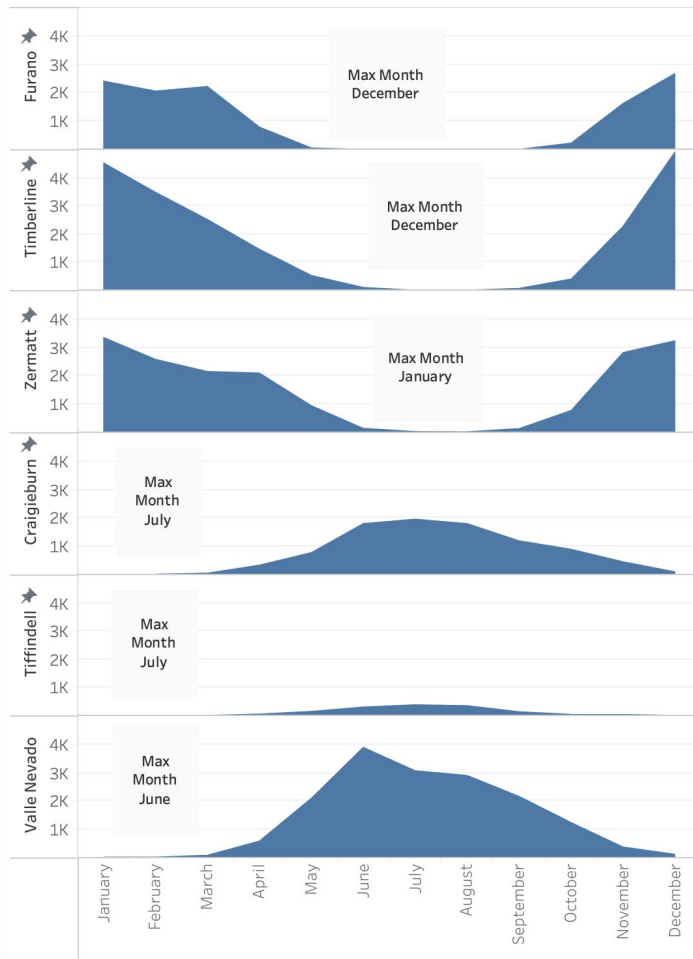
Furano - 47.60 (2004)
 Timberline - 71.43 (1996)
 Zermatt - 115.90 (2012)
 Craigieburn - 87.90 (2015)
 Tiffindell - 33.07 (2006)
 Valle Nevado - 96.05 (2002)

Dashboard 4

Snowfall by Week (encompassing since 1990)



Snowfall Per Month (encompassing since 1990)



Random Forest Classifier

Predict snowfall

Based on 2 features

- Humidity
- Temperature

```
# Calculate Accuracy Score
acc_score = accuracy_score(y_test, predictions)
acc_score
```

0.8626425591098749

```
# Displaying Results
print("Confusion Matrix")
display(cm_df)
print(f"Accuracy Score : {acc_score}")
print("Classification Report")
print(classification_report(y_test, predictions))
```

Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	12985	1248
Actual 1	1221	2521

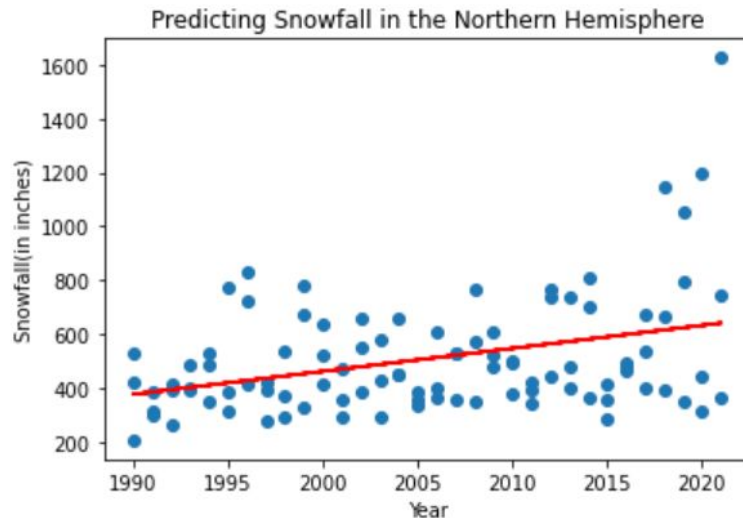
Accuracy Score : 0.8626425591098749

Classification Report

	precision	recall	f1-score	support
0	0.91	0.91	0.91	14233
1	0.67	0.67	0.67	3742
accuracy			0.86	17975
macro avg	0.79	0.79	0.79	17975
weighted avg	0.86	0.86	0.86	17975

Linear Regression

Northern Hemisphere

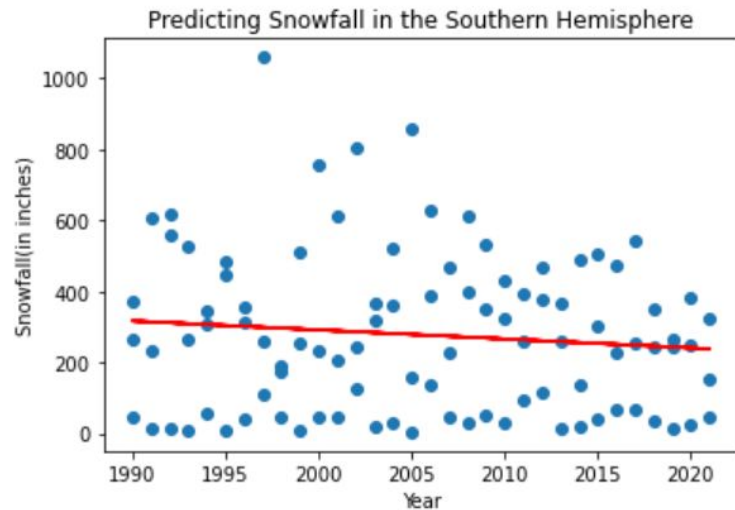


```
#Examine slope and y-intercept  
print(model.coef_)  
print(model.intercept_)
```

```
[8.53504643]  
-16608.737286168132
```


Linear Regression

Southern Hemisphere

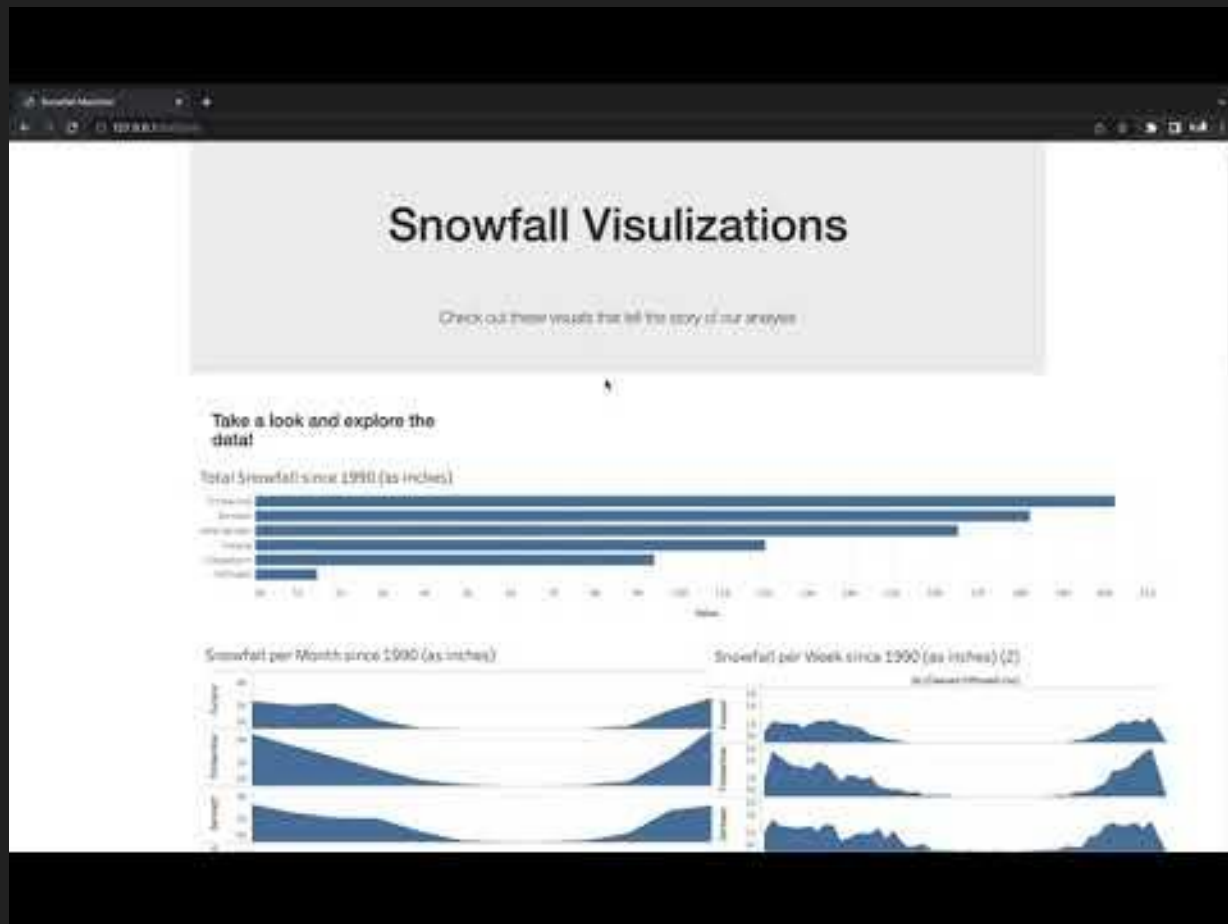


```
#Examine slope and y-intercept  
print(model.coef_)  
print(model.intercept_)
```

```
[-2.53033847]  
5352.146708822092
```

Flask App

- Interactive Snowfall Predictor
- Interactive Dashboard
- app.py + Pickle file + HTML



Conclusion

- Finds
- What could we have done differently



Citations

“Open Weather.” *Open Weather*, 2012.