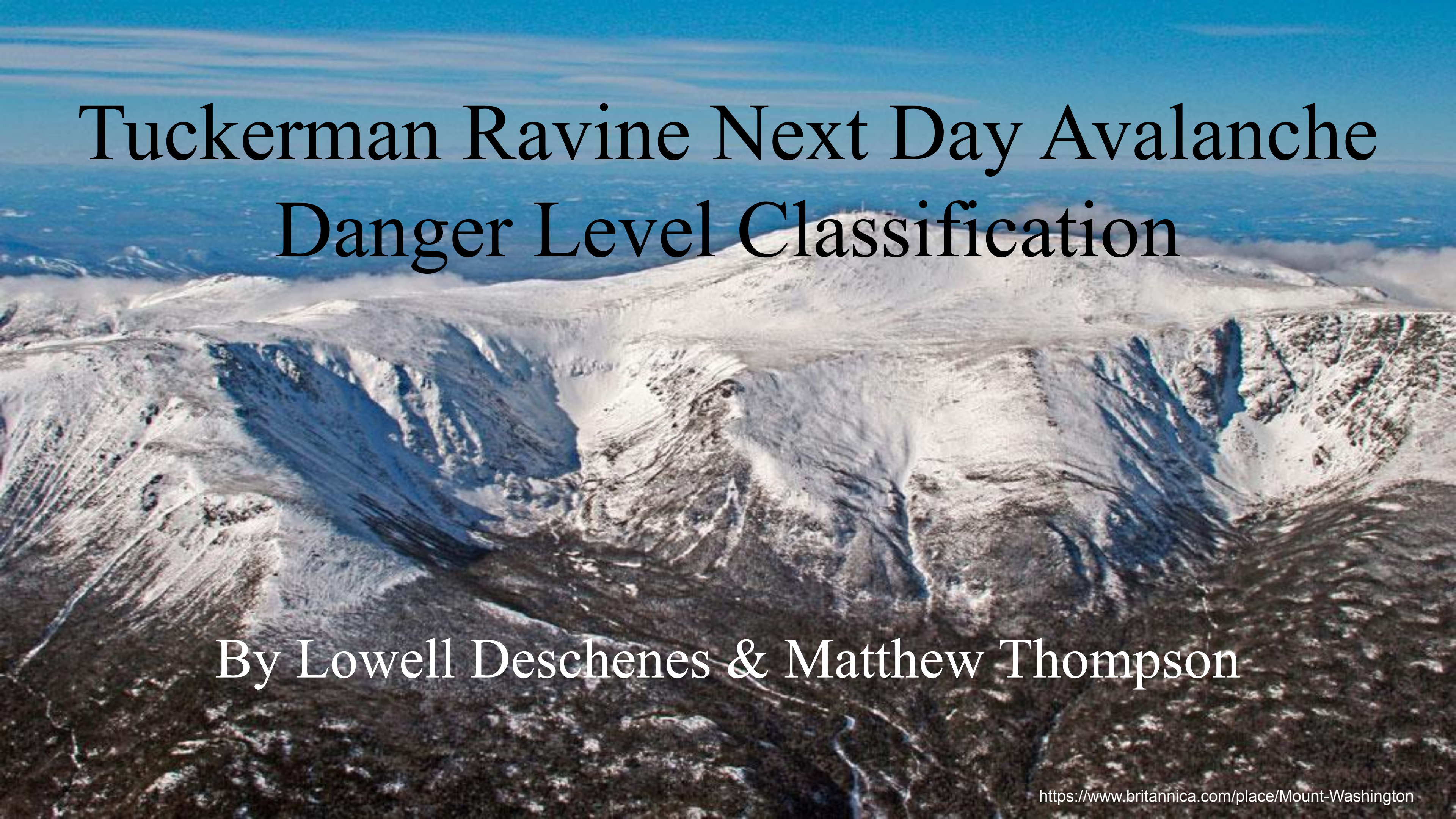


Tuckerman Ravine Next Day Avalanche Danger Level Classification

By Lowell Deschenes & Matthew Thompson

https://www.britannica.com/place/Mount-Washington


Tuckerman Ravine

Our Motivation

- Models have been created to predict avalanche occurrence in other mountainous regions of the world, but as far as we know never been created for Tuckerman Ravine
- Difficult problem due to lack of consistent avalanche and snow data
- Access to phenomenal weather data from Mount Washington Observatory
- Common place of recreation for backcountry skiing, hobby that Lowell and I have in common

https://www.britannica.com/place/Mount-Washington

Related Work

- Studies with high-dimensionality terrain, snow, weather, and avalanche data boast impressive results
- Researchers at the University of Iran published a paper in 2019 attempting to predict snow avalanche hazard paths using SVM with spatial data, achieved ROC AUC value of 90
- Studies without spatio-temporal data shown to predict the occurrence of an avalanche with a decent effectiveness of ~70% accuracy
- SVM and logistic regression were most successful models without spatial data
- All the studies examined tried to predict avalanche occurrence rather than the avalanche danger level

What Causes an Avalanche?

Requires Four Ingredients

1. A Slab
2. Weak Layer
3. Trigger:
    - Unnatural pressure (snow sports)
    - Higher temperatures
    - New snowfall
    - Wind-drifted snow
4. Slopes Steep Enough to Slide:
    - Between 25-45 degrees
    - Most frequently between 36-38 degrees

Photo Credit: iStock.com/NaniP

The Data

- Weather data from the Mount Washington Observatory, recorded at summit of Mt. Washington
    - 1935-Present
    - Consistent, well-formatted
- Snow and avalanche data from Mount Washington Avalanche Center, recorded at Hermit Lake (near base of Tuckerman Ravine)
    - 2010-2011 season to 2019-2020 season
    - Inconsistent, many NULL values
- Combined these datasets into dataframe by joining on date column
- Total of 1,258 samples in 72 dimensions
- Also scraped avalanche occurrence data, turned out to be very small dataset, shifted focus from predicting avalanche occurrence to next day danger level

Data Cleaning

- Dropped Feature If:
  - Abundance of NULL values in a feature
  - Domain knowledge, knew would not contribute significantly
  - Very low significance via random forest feature importance
  - Feature did not include consistent/useful data
  - Feature displayed extreme linear correlation with another feature
- Created boxplots for all numerical features to see if there were outliers or misentered values, found none
- Looked at the unique values of categorical features for odd entries, found in wind direction feature, fixed by creating one-hot encoded wind features, dropped bad value samples
- Total of 1,253 samples in 25 dimensions post-cleaning

Features We Dropped

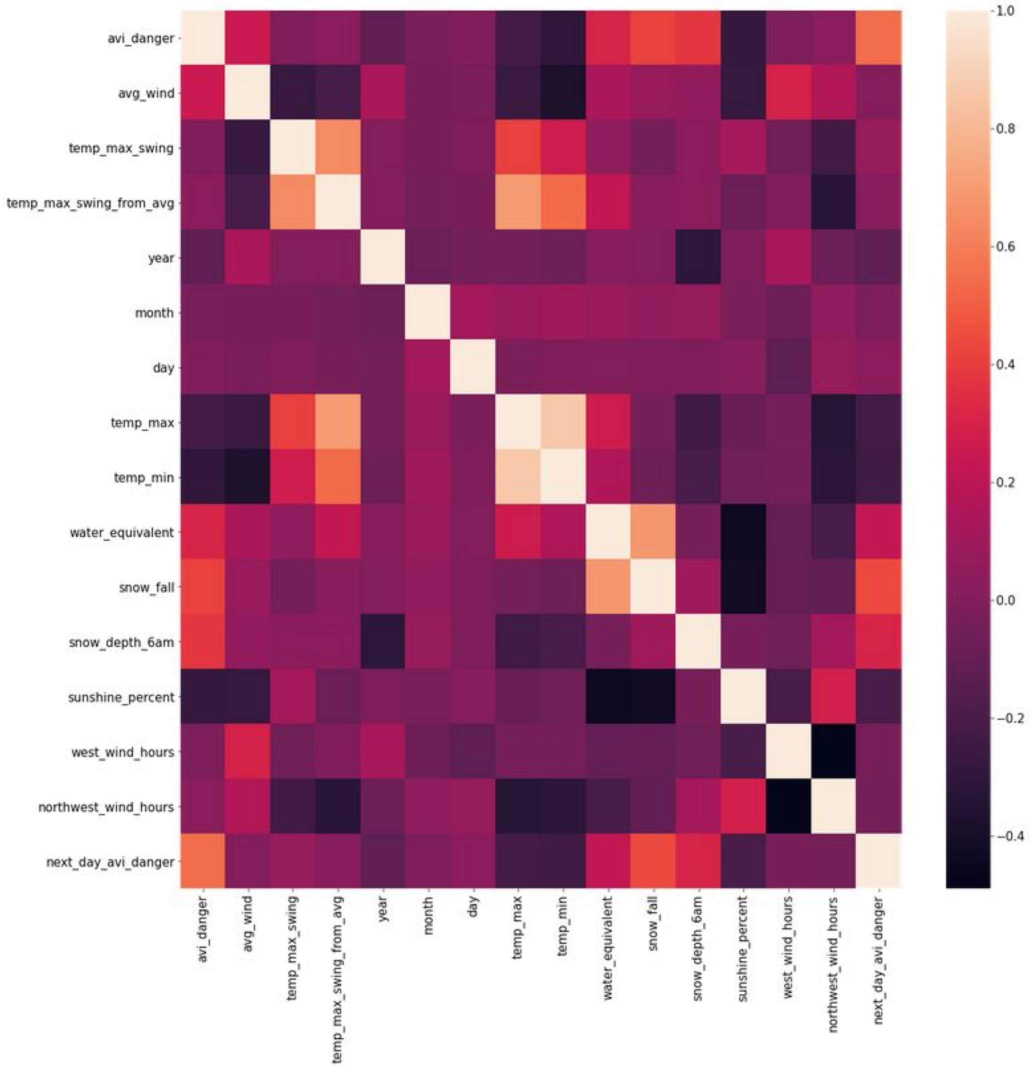
- PGTM
- AVY_CHARACTER
- WET_DANGER
- DRY_DANGER
- WET_LOOSE
- WET_SLAB
- WIND_SLAB
- STORM_SLAB
- CORNICE_FALL
- PERSISTENT_SLAB
- DEEP_SLAB
- DRY_LOOSE
- GLIDE_AVALANCHE
- LONG_SLIDING_FALL
- CURRENT TEMP
- WSF5_ATTRIBUTES
- SKY CONDITION
- PRECIP TYPE/RATE

- PSUN
- PRCP
- PRCP_ATTRIBUTES
- SNOW_ATTRIBUTES
- SNWD
- SNWD_ATTRIBUTES
- TMAX
- TMAX_ATTRIBUTES
- TMIN
- TMIN_ATTRIBUTES
- TSUN
- TSUN_ATTRIBUTES
- WDF5_ATTRIBUTES
- WSF5

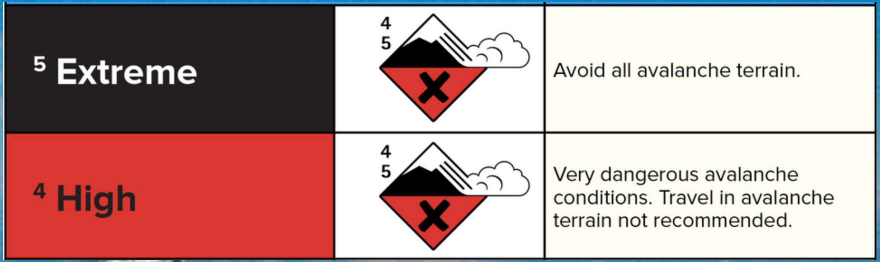
- FORM/SIZE
- HN24_CM
- HW in Tube (MM)
- H24W in Can (MM)
- DENSITY
- HST
- water_equivalent_trace
- snow_fall_trace
- snow_depth_6am_trace
- Sunshine_sum
- Skycover_sum
- skycover_avg_sunrisetosunset
- year_y
- month_y
- day_y

Completed Data Cleaning

```
RangeIndex: 1253 entries, 0 to 1252
Data columns (total 25 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   avi_danger               1253 non-null   float64
 1   avg_wind                 1253 non-null   float64
 2   temp_max_swing           1253 non-null   float64
 3   temp_max_swing_from_avg  1253 non-null   float64
 4   year                     1253 non-null   float64
 5   month                    1253 non-null   float64
 6   day                      1253 non-null   float64
 7   temp_max                 1253 non-null   int64
 8   temp_min                 1253 non-null   int64
 9   water_equivalent         1253 non-null   float64
 10  snow_fall                1253 non-null   float64
 11  snow_depth_6am           1253 non-null   float64
 12  wind_speed_sum           1253 non-null   int64
 13  sunshine_percent         1253 non-null   int64
 14  west_wind_hours          1253 non-null   int64
 15  northwest_wind_hours     1253 non-null   int64
 16  prevailing_wind_E        1253 non-null   int64
 17  prevailing_wind_N        1253 non-null   int64
 18  prevailing_wind_NE       1253 non-null   int64
 19  prevailing_wind_NW       1253 non-null   int64
 20  prevailing_wind_S        1253 non-null   int64
 21  prevailing_wind_SE       1253 non-null   int64
 22  prevailing_wind_SW       1253 non-null   int64
 23  prevailing_wind_W        1253 non-null   int64
 24  next_day_avi_danger      1253 non-null   float64
```


avi_danger
avg_wind
temp_max_swing
temp_max_swing_from_avg
year
month
day
temp_max
temp_min
water_equivalent
snow_fall
snow_depth_6am
sunshine_percent
west_wind_hours
northwest_wind_hours
next_day_avi_danger
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

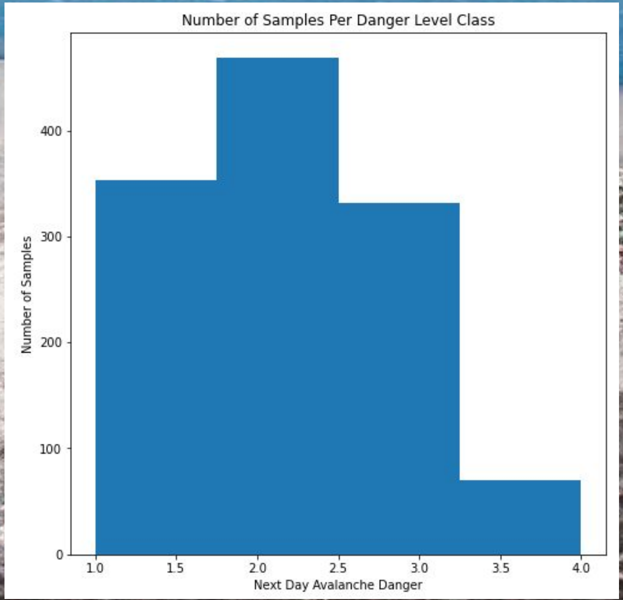
Feature Engineering



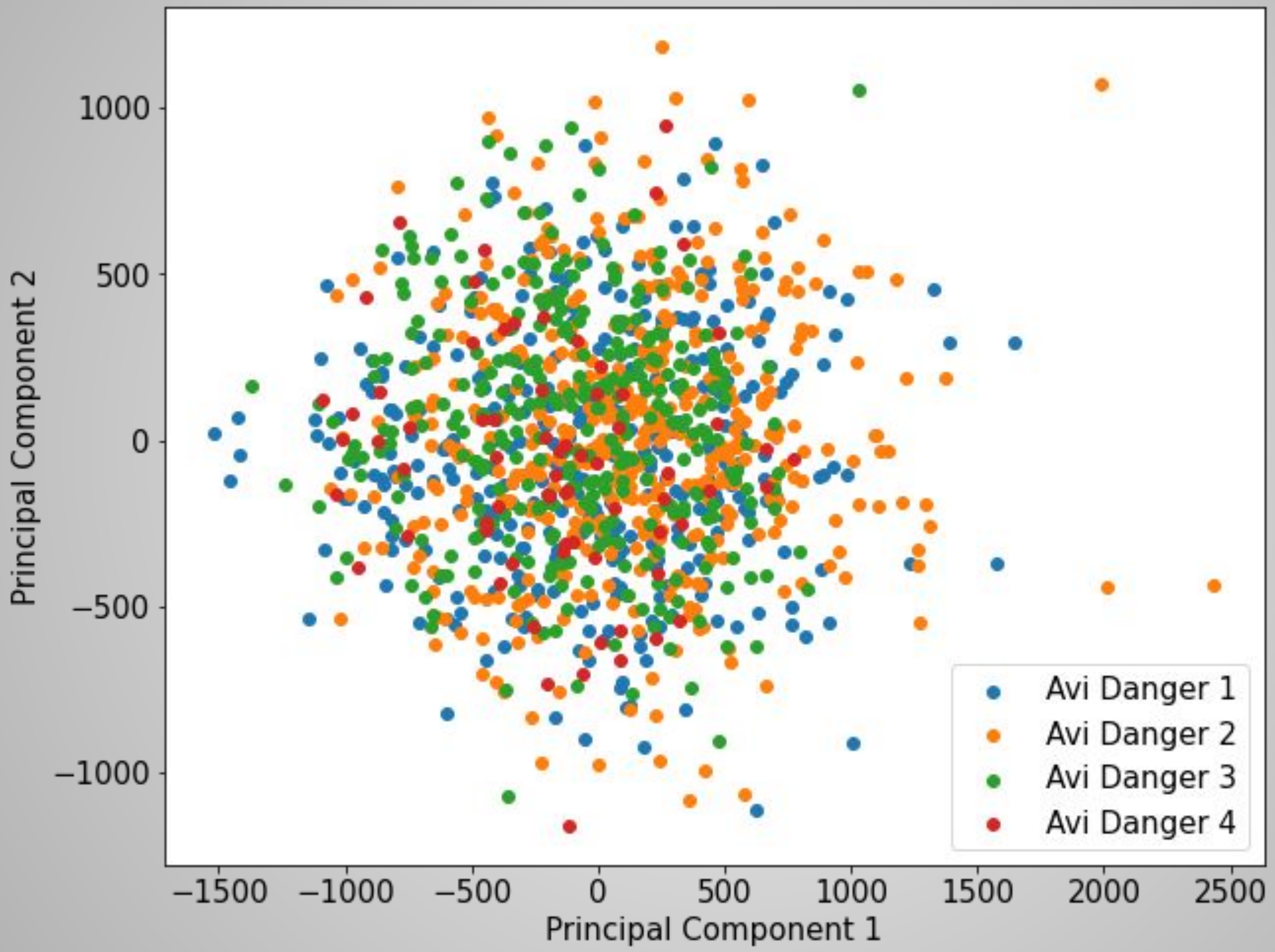
- Combined danger levels 4 and 5 due to lack of data
  - Only 2 samples of danger level 5
- Created new features:
  - three_day_snowfall
  - five_day_snowfall
  - next_day_avi_danger
    - Dropped last day of each season so next day remains within same season
  - month
  - day
  - year
  - Prevailing wind features created via one-hot encoding
  - Previous days features for every feature (1-day ago, 2-days ago)
    - Dropped first 2 days of each season so previous days remain within same season

Data Exploration - Unbalanced Data

- Avalanche danger levels skewed towards lower danger level
- Created balanced class_weight dictionary using sklearn.utils.class_weight function, applied to models to balance classes
  - Danger Level 1 = 353 samples, 0.89 weighted
  - Danger Level 2 = 469 samples, 0.64 weighted
  - Danger Level 3 = 332 samples, 0.94 weighted
  - Danger Levels 4 & 5 = 70 samples, 3.96 weighted
- Attempted use of oversampling (Random Over Sampling, SMOTE), showed no improvement over class_weight



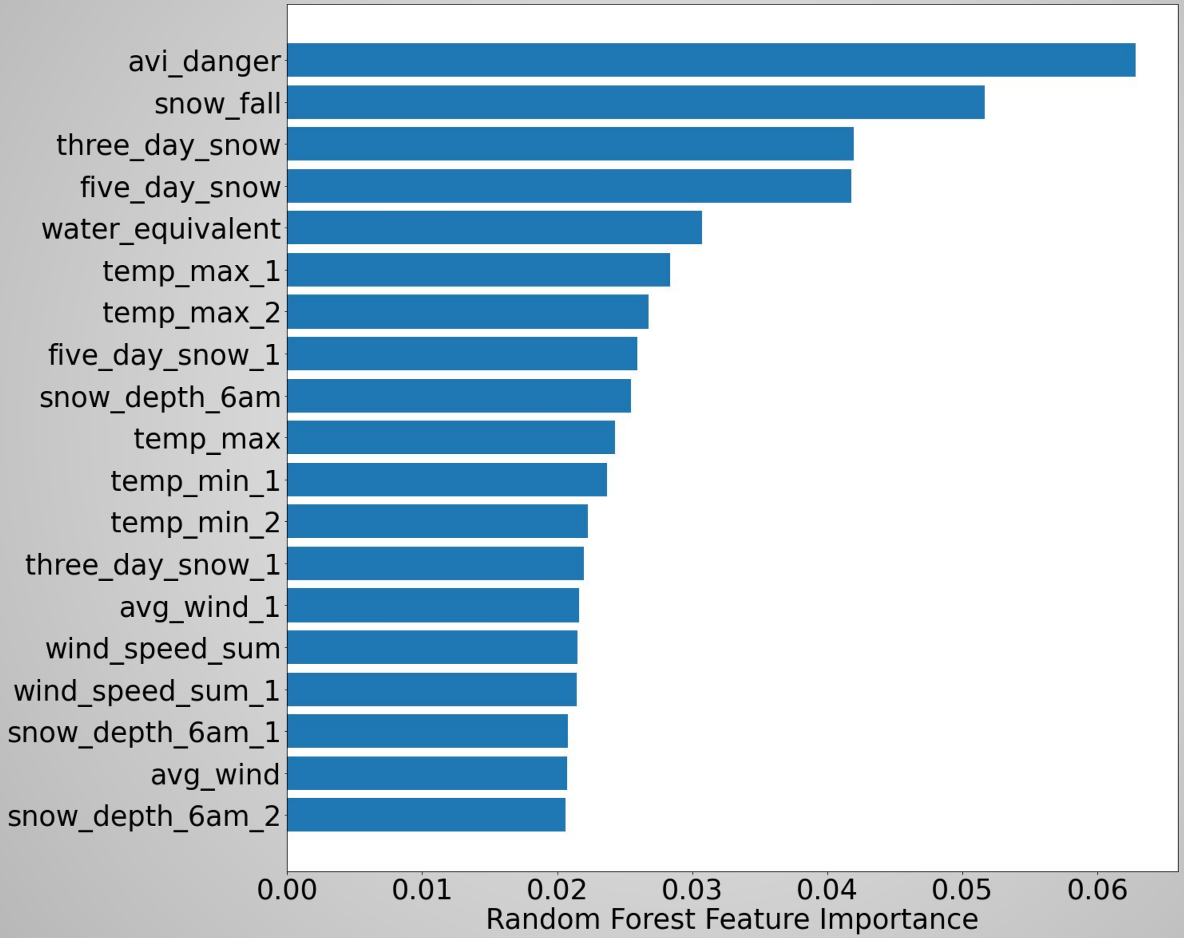
PCA of Our Chosen Features



Data Modeling

- Prior knowledge motivated us to focus on logistic regression and SVM models
- Instead of a binary classification we are dealing with a multiple classification problem where random forest or decision trees may perform best
- This motivated us to try a plethora of models including
    - Random Forest
    - Decision Trees
    - Logistic Regression
    - SVM
    - Extra Trees
    - Gradient Boosting
    - KNN

The Top 20 Features



Final Results

Accuracies reported with cross validation

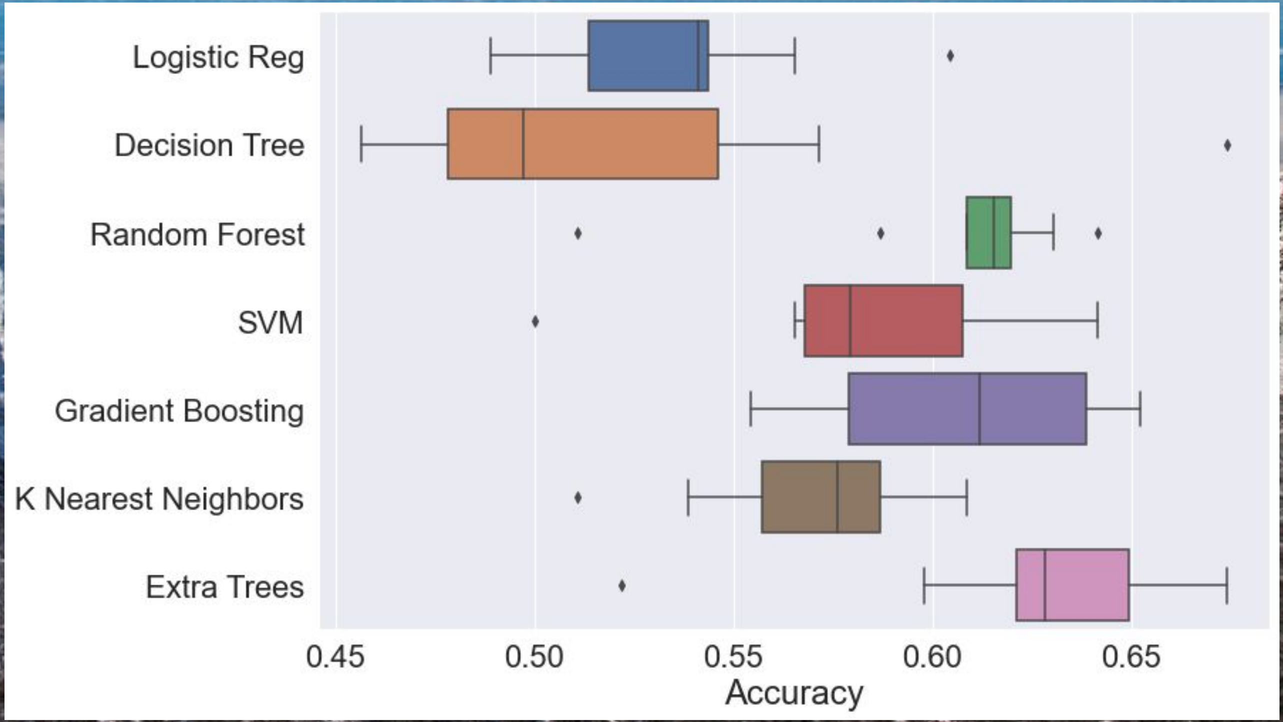
- Logistic Regression: Accuracy Mean = 0.534938 Standard Deviation = 0.033049
- Decision Tree: Accuracy Mean = 0.519625 Standard Deviation = 0.062619
- Random Forest: Accuracy Mean = 0.605686 Standard Deviation = 0.034371
- SVM: Accuracy Mean = 0.584986 Standard Deviation = 0.037875
- Gradient Boosting: Accuracy Mean = 0.608887 Standard Deviation = 0.033662
- K Nearest Neighbors: Accuracy Mean = 0.570807 Standard Deviation = 0.028429
- Extra Trees: Accuracy Mean = 0.624188 Standard Deviation = 0.039488

All models fit with Randomized Search CV for optimized hyperparameters

Analysis of Results

Best Classifiers:

- Random Forest
- Gradient Boosting
- Extra Trees



Statistical Model Comparison

Ran the Wilcoxin Rank Sum test to get a non parametric comparison of each of the top models

- Random Forest vs. Gradient Boosting: p-value = 0.138860 no statistically significant difference
- Random Forest vs. Extra Trees: p-value = 0.448723 no statistically significant difference
- Gradient Boosting vs. Extra Trees: p-value = 0.043693 statistically significant difference

Due to multiple comparisons these p-values were corrected using the Benjamini/Hochberg procedure

Discussion and Conclusions

- More data necessary to create effective predictor
- Many approaches tried to fix unbalanced data, no improvements found, points to small dataset being limiting factor
- No features on snowpack layers, slope steepness, or avalanche occurrence
- This project could be redone in the future and will likely achieve better results with larger dataset and spatial data

Photo Credit: Mike Pelchat

Thank You For Listening!