ORIE4741 Final Project Report: Mount Washington Weather

Benjamin Moose, Yama Bazger, Adrianna Ahn

May 10, 2024

Project Motivation

Mount Washington, New Hampshire, is the site of some of the most extreme weather in the United States, with frequent wind gusts above 100mph and frigid temperatures. Meteorologists at the Mount Washington Observatory are responsible for issuing forecasts for the summit and surrounding peaks ("Higher Summits Forecast", 2024). Current numerical weather prediction (NWP) models used in operational forecasting provide forecasts with resolutions up to 3km x 3km, but small-scale topography could impact the weather that the summit of Mt. Washington sees relative to other points in a 3km x 3km spatial grid cell surrounding it (National Oceanic and Atmospheric Administration, 2024). Since models cannot explicitly resolve processes that occur inside each grid cell, we hypothesize that we may be able to improve (or correct any potential bias in) predictions of wind speed at the summit of Mt. Washington via machine learning techniques applied to this 3km x 3km model output, as well as earlier observations from the summit. To formalize the problem, we intend to use station observation data and NWP model forecasts from early morning (10 UTC) to predict wind speeds at the Mt. Washington summit during the afternoon (20UTC) on the same day.

Data Acquisition

We use two sources of data for this project. The first is a dataset from the Iowa Environmental Mesonet (IEM) containing weather observations from an automated station on the summit of Mt. Washington ("IEM", n.d.).

These observations are taken semi-regularly at a near-hourly interval (though missing data exists for some times). This dataset includes features such as a 'present weather' text field, temperatures, dewpoint temperatures, relative humidity values, and wind speeds and directions, among others ("IEM", n.d.). As we intend to predict wind speeds, we use wind speed data from this source as our verification dataset. Further, however, we believe that observational data may enhance our model's accuracy, so we also use morning (10UTC) observations from the summit from this dataset within our set of features.

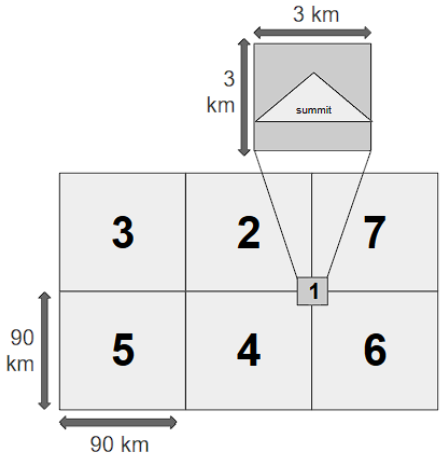

Figure 1: Schematic of regions where HRRR output was averaged and input into feature set. Number corresponds to number in feature label within code.

The second dataset we consider, entirely for use as features, is a database of NWP model forecasts from the High-Resolution Rapid Refresh weather model (HRRR), a 3km-resolution model that provides forecasts up to 18 hours from its initialization time (National Oceanic and Atmospheric Administration, 2024). We consider HRRR forecasts from runs initialized at 10UTC (5/6AM EST/EDT) for 20UTC (2/3PM EST/EDT), and collect predicted values of multiple potentially-relevant atmospheric variables at the nearest model grid point to Mt. Washington, as well as spatial means of forecast fields for regions around Mt. Washington (visualized in Figure 1). These data were obtained via Amazon Web Services through the Python package *Herbie* (Blaylock, 2024; National Oceanic and Atmospheric Administration, 2024).

Data Cleaning and Feature Transformation

The IEM dataset includes multiple fields that require adjustment into a suitable quantitative format for our regression problem. We first select a subset of the IEM data including seemingly useful features (temperature, wind speed, precipitation) and remove repetitive, constant, and/or likely irrelevant features (ice accretion, snow depth, station name). We then filter by time to select only observations between (1) 19:30-20:30 UTC (for verification / labels), (2) 9:30-10:30 UTC (for morning observation features), and (3) 3:30-4:30 UTC (to obtain 6-hour changes in fields leading up to the morning observations. The below table outlines some further adjustments made to the station observation dataset.

Column/Feature	Adjustment
'tmpf' [temperature]	Remove 6 (out of 10485) data points with missing temperature data
'gust' [wind gust]	Set equal to wind speed when missing
'wxcodes' [current weather text field]	Use one-hot text encoding for some conditions ('RA', 'SN', 'BL', 'FG', 'UP', 'M').
'drct' [wind direction]	Remove 25 (of 10479 total) data points with missing wind direction
'drct' [wind direction]	Convert degree value to direction using 8 categories (N, NW, W, SW, etc.). Then use one-hot encoding for each category.
'rh' [relative humidity]	Drop 9 (of 10454 total) data points with missing data
'valid' [datetime]	Select only data points for which the 4UTC, 10UTC, and 20UTC data is all available
'valid' [datetime]	Remove datetime feature, replace with year, month, and day features

Table 1: Some adjustments to raw IEM station dataset. Adjustments that affected 1 data point not included.

Since trends are important for forecasting, we compute the differences between 4UTC and 10UTC observations to get rates of change of features over the 6-hour period leading up to the forecast initialization time. Each of these changes is included as a separate feature in the dataset (such that the station-based features include the 10UTC value of an observation and the 4UTC-10UTC change for each observation). Finally, we also make a standardized version of the dataset by taking z-scores of the components of each data point with respect to corresponding feature means and standard deviations - this permits more meaningful comparison of coefficient magnitudes in linear models.

We select relevant fields from the HRRR model's 10UTC run (10-hour forecast for 20UTC) at the surface level and 800 millibar (mb) pressure level (relatively close to the elevation of Mount Washington in a standard atmosphere), including temperature, wind speed, dewpoint, precipitation, and wind components (Young, 2017). We then average these fields over the domains listed in Figure 1 for each time considered in our dataset. Data cleaning is not a challenging task for NWP data (since missing data does not exist, beyond dates for which *Herbie* encounters errors accessing archived data), but we do apply some postprocessing to the raw model data before consideration in our model, such as transforming wind components into 8

wind direction categories (N, NW, W, etc.) and using one-hot encoding to represent them. Further, we note that spatial changes in fields may provide useful information, so we compute changes in fields between each of the six 90km x 90km regions considered and their neighboring points (including diagonal) and add these changes as features. As feature names within the dataset may not be easily understandable, we include the following table outlining the structure of a selection of feature names for reference.

Example Feature Name	Feature Description	Source Dataset
t800[x][y]	Forecast difference in spatial mean 800mb temperature between region [x] and point [y]	HRRR Model
tdsfc[x]	Forecast surface dewpoint temperature at region [x]	HRRR Model
u/v/wssfc[x]	Forecast surface u-component/v-component/total speed of wind at region [x]	HRRR Model
wd_5am_NE	One-hot encoding of whether observed wind direction was northeast at 5/6am EST/EDT	IEM Obs.
FG_6hrchange	Difference between one-hot encodings of fog presence at 11pm/12am and 5am/6am EST/EDT	IEM Obs.

Table 2: Examples of some feature names, descriptions, and source datasets

Exploratory Data Analysis

We perform some initial exploration of the dataset by identifying the feature that is most strongly correlated with the label (in this case, the observed wind speed at 20UTC on the summit). We find that this feature is ***ws8001***, the forecast wind speed at 800mb for the grid cell over Mt. Washington, and we include a scatterplot of this feature's value vs. wind speed below:

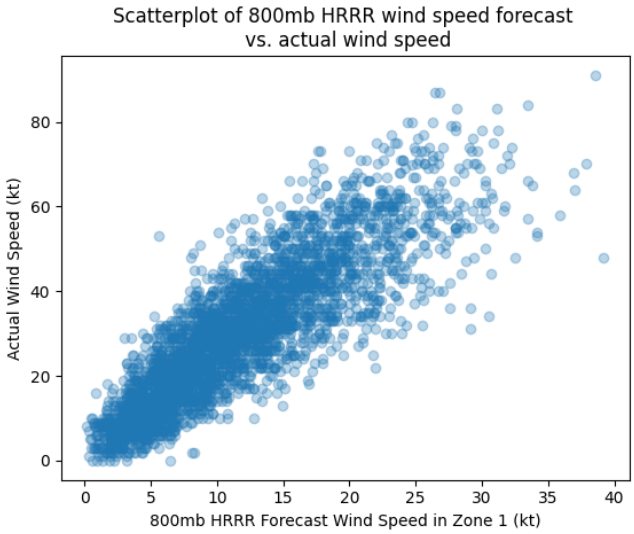

Figure 2: Scatterplot of 800mb HRRR forecast wind speed for grid cell over Mt. Washington and observed wind speed at the summit.

We note that this single feature, a model-based forecast value, has a correlation coefficient of 0.86 with actual wind speed. However, from inspection of Figure 2, we note that the linear relationship may somewhat degrade for high wind speeds. Thus, to continue our initial analysis, we investigate correlations between all features and the wind speed labels, as well as their *changes* when restricting our dataset to high wind speeds (wind speed > 40kt). The increased difficulty of predicting higher wind speeds with single predictors drives our inclusion of the ***Predicting High Winds*** section in this report.

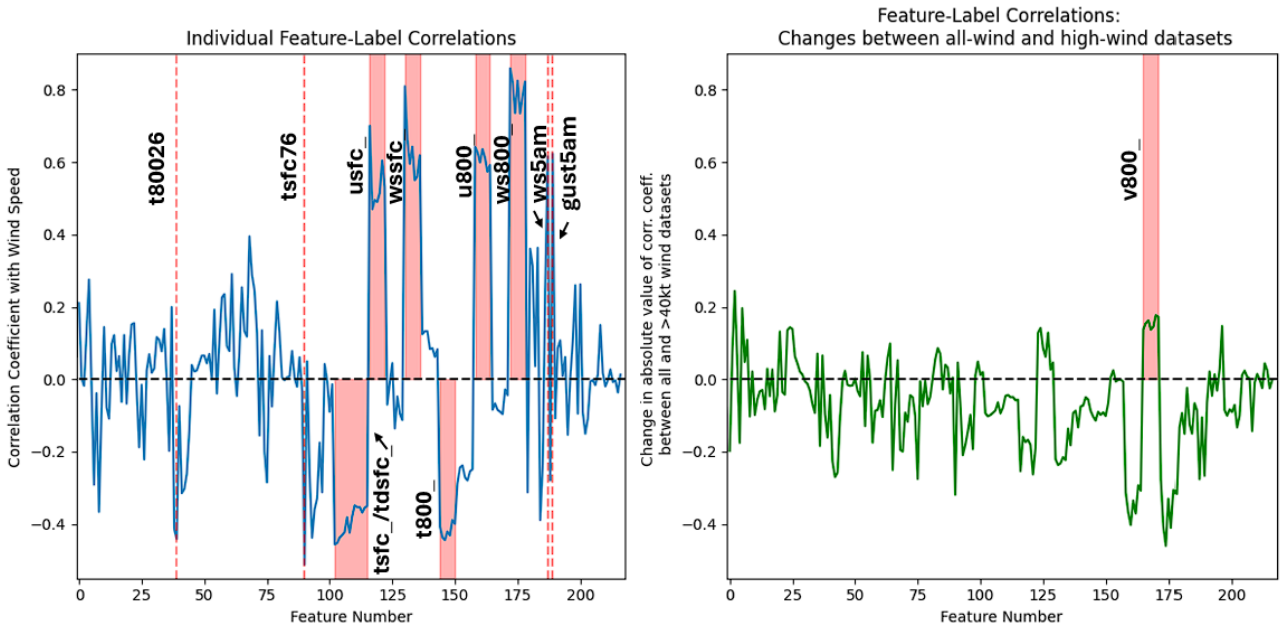

Figure 3: Feature-label correlation for all features [left], and change in absolute value of feature-label correlation when dataset restricted to include only points with wind speeds > 40kt [right]. Individual features (dashed) or groups of features (filled) with noteworthy correlations / changes highlighted and labeled. Underscores indicate variable number within feature name for group of features.

The above figure indicates that most variables highly-correlated with wind speed involve either surface or 800mb wind speeds from HRRR forecasts, earlier observed wind speeds from the summit, or temperatures (likely a proxy for season). Interestingly, we see negative correlation between *t80026* and *tsfc76* and wind speed, since these temperature differences are likely negative when cold fronts are moving through the region (which frequently bring higher winds). We note that the correlation magnitude between 800mb v-wind components (north-south radial wind) increases when the dataset is filtered to remove winds below 40kt, perhaps suggesting that the forecasted v-component of wind matters more in differentiating high from very-high wind events than in differentiating low from high wind events.

Linear Models

We first consider predicting wind speed from linear models applied to all available features and data points. We split the dataset via a 60%-20%-20% Train-Test-Validate random split for analysis of model performance. Performance of linear models is measured using the scikit-learn default metric, the coefficient of determination (R^2). R^2 values near 1 indicate that the model fits the data nearly perfectly, whereas R^2 values near (or below) zero indicate that the model

performs nearly the same as (or worse than) simply predicting the mean ("3.3. Metrics and scoring", n.d.).

Ordinary Least Squares

We first consider ordinary least-squares regression with the single ***ws8001*** predictor and all predictors, respectively, fitted to non-standardized data [Results shown in Figure 4]. We perform 5-fold cross validation (CV) using the training and validation sets combined, as well as training the model on the training dataset and testing on the validation set.

Model	Validation R^2	5-Fold CV Mean R^2	5-Fold CV StDev R^2
Single-Feature OLS	0.716	0.732	0.019
All-Feature OLS	0.841	0.840	0.008

Table 3: Single-feature (ws8001) and all-feature ordinary least squares model performance. Second column indicates validation-set R^2 of model trained on training set and evaluated on validation set. Third and fourth columns indicate mean and st. dev. of R^2 from a 5-fold cross validation using the training and validation sets combined.

We find that OLS with all features considered outperforms the single-feature OLS by approximately 0.13 and 0.11 (in R^2 score) when tested on the validation set and in the 5-fold cross-validation mean, respectively. The standard deviation of the 5-fold cross-validation R^2 for the all-feature model is approximately half that of the single-feature model (but both are small compared to the means), indicating that we do not seem to be overfitting the data with the linear models. We next standardize the data (to allow for more meaningful comparison of coefficients of the linear model) and run OLS, yielding results nearly identical to the all-feature OLS in the above table. Coefficient values of this linear model are generally on the order of 10^{13}, with similarly high standard deviations of the coefficients (from a 50-sample bootstrap, with each sample containing half of the data points) on the order of 10^{12}. These values suggest very high model variance, with perturbations in individual features resulting in extremely large differences in model predictions. However, the mean R^2 of the bootstrapped models remains 0.82 ± 0.006 ($Mean \pm 1SD$). Perhaps the spatial relationships and cross-correlation between variables ensure that even unseen data are likely to receive reasonable predictions despite the high model variance. We still, however, would prefer a model with smaller coefficients (and thus less sensitivity to small feature perturbations), and we explore regularized regression in the next section.

Lasso and Ridge Regression

To prevent the large coefficients that we see in the OLS model (which may result in extremely poor generalization of the model to outliers), we apply two regularization techniques to the quardatic-loss linear models: ridge and lasso regression. Ridge regression minimizes the quadratic loss function plus a scaled euclidean norm of the parameter vector w ($argmin_w ||y - Xw||_2^2 + \alpha||w||_2^2$), where w is the parameter vector, y are the training labels, and X are the training data points (He, 2024). This configuration heavily penalizes large coefficients in w and thus should reduce sensitivity of model predictions to perturbations in features. As shown in Figure 5 [left plot], from $\alpha = 0.05$ to $\alpha \approx 100$, we see very little change in mean 5-fold cross-validation R^2 score, indicating that we achieve good model generalization for a wide range of regularization parameters (α). Further, Figure 5 [right plot] indicates that coefficients on

each feature are *significantly* smaller than in OLS for the $\alpha = 5$ case. We select $\alpha = 5$ since coefficient magnitudes are relatively small and mean cross-validation R^2 is relatively good, although many different α values would be suitable here. Results for the $\alpha = 5$ ridge regression validation R^2 and 5-fold CV mean / st. dev. R^2 are included in Table 4.

We next consider Lasso Regression, a method that finds the minimum of the loss function plus a scaled 1-norm (sum of absolute values of entries) of the feature coefficient vector w (He, 2024). Lasso regression solves $argmin_w||y - Xw||_2^2 + \lambda||w||_1$, resulting in less sensitivity to higher w_i (compared to ridge) and a tendency to arrive at a sparse coefficient vector w (He, 2024). Figure 6 [left] shows CV mean R^2 and the number of nonzero feature coefficients as a function of λ. A tradeoff between CV performance and regularization parameter is somewhat more clear here than in the ridge case, but a relatively good sparse solution can be obtained even for relatively small λ. Figure 6 [right] shows feature coefficient magnitudes for lasso regression with $\lambda = 1$ (chosen for a balance between relatively high accuracy and sparsity). Figure 6 [right] suggests that (unsurprisingly), ***ws8001*** is the most important predictor of wind speed, with other surface forecast wind components/speed and a forecast 800mb temperature features also seemingly important. Comparison of feature importance by coefficient magnitude here is somewhat reasonable since the data are standardized. Results for the $\lambda = 1$ case, included in Table 4, suggest that lasso regression with $\lambda = 1$ may have slightly poorer performance on both the validation set and in the 5-fold CV mean when compared to ridge regression, but both models show a similar standard deviation of R^2 score over the 5-fold CV. We evaluate the best model (Ridge Regression) on the test dataset and obtain $R^2 = 0.850$.

Model	Validation R^2	5-Fold CV Mean R^2	5-Fold CV StDev R^2	Test R^2
Ridge Reg. ($\alpha = 5$)	0.839	0.840	0.007	0.850
Lasso Reg. ($\lambda = 1$)	0.823	0.829	0.008	—

Table 4: Ridge and lasso regression model performance for specified parameters. Second and fifth columns indicate validation-set and test-set R^2 of model trained on training set. Third and fourth columns indicate mean and st. dev. of R^2 from a 5-fold cross validation using the training and validation sets combined.

Tree-Based Models

We next consider tree-based regression models. Some parameter tuning was performed on decision tree, bagged decision tree, and feature-subsampled random forest models, and model results at optimal parameter settings (within the ranges considered) are included in Table 5. Figures illustrating the models' performance are included in the appendix. Three experiments were performed to tune features with random_state set to 1:

- (1) Run DecisionTreeRegressor. Iterate through max_depth in [1, 20] and test l_2 and l_1 splitting criteria. Keep all other parameters as defaults.
- (2) Run sets of 50 bagged regression decision trees. Iterate through max_depth in [1, 5] and test l_2 and l_1 splitting criteria. Also test bootstrap sample proportion (max_samples) in [0.2, 0.7]. Keep all other parameters as defaults.
- (3) Run 50-tree random forests with max_samples = 0.7. Iterate through max_depth in [1, 5] and test l_2 and l_1 splitting criteria. Also test feature subsample proportion

(max_features) in [0.2, 0.7]. Keep all other parameters as defaults.

Model	Validation R^2
Decision Tree w/ L1 Splitting Criterion (max_depth = 4)	0.789
Bagged Trees w/ L1 Splitting Criterion (max_depth = 5, max_samples = 0.7)	0.842
Random Forest w/ L2 Splitting Criterion (max_depth = 5, max_features = 0.7)	0.846

Table 5: Validation-set R^2 values for the optimal parameter combinations in each of the listed experiments.

We find that tree-based models achieve very similar validation R^2 scores to our linear models. The Random Forest Regressor (experiment 3) has the highest validation R^2 and achieves a test R^2 of 0.849. Using scikit-learn, we also find the Mean Decrease Impurity of features in our DecisionTreeRegressor (experiment 1) and RandomForestRegressor (experiment 3) in Figure 9 ("Feature importances with a forest of trees", n.d.). As expected, surface and 800-mb wind speed features dominate the feature importance for both the decision tree and random forest.

Predicting High Winds

We next consider the problem of predicting high winds, defined here as those over 40 knots. We rerun the linear models from the ***Linear Models*** section on a smaller dataset containing data points from high-wind days. Results from each of the models are included in the below table and Figures 10 and 11 are analogous to Figures 5 and 6 for the all-wind-data case. From Figure 10 [left], we see that the validation R^2 is quite insensitive to large α (permitting large regularization parameters in the ridge regression case and small feature coefficients). Further, $\lambda = 0.6$ provides a good balance between sparsity and valdiation accuracy (visually) from Figure 11 [left], and similar nonzero features are selected in the high-wind and all-wind cases as indicated by Figure 11 [right].

Model	Validation R^2	5-Fold CV Mean R^2	5-Fold CV StDev R^2	Test R^2
Single-Feature OLS	0.388	0.359	0.079	—
All-Feature OLS	0.387	0.458	0.074	—
Ridge Reg. ($\alpha = 5$)	0.407	0.484	0.074	—
Lasso Reg. ($\lambda = 1$)	0.528	0.530	0.030	—
Ridge Reg. ($\alpha = 300$)	0.518	0.555	0.048	—
Lasso Reg. ($\lambda = 0.6$)	0.559	0.555	0.029	0.516

Table 6: OLS, ridge and lasso regression model performance for specified parameters. Models trained/evaluated on dataset filtered to select wind speeds > 40kt. Second and fifth column indicate validation-set and test-set R^2 of model trained on training set. Third and fourth columns indicate mean and st. dev. of R^2 from a 5-fold cross validation using the training and validation sets combined.

We find that, of the 6 models considered, lasso regression with $\alpha = 0.6$ performs the best at maximizing R^2 when tested on the validation set, improving the single-feature OLS R^2 value by approximately 0.17 (with test-set R^2 slightly lower at 0.516). While predictability is still much worse than that of models applied to the unfiltered dataset, including multiple features can still meaningfully improve model performance. We also note an interesting result - while the cross-validation scores from the all-feature OLS model are still reasonable, the mean validation set

accuracy of 50 all-predictor OLS models trained on bootstrapped samples of size equal to half of the training set size is approx. -10×10^{22} with even higher-magnitude standard deviation. This indicates the sort of overfitting problems that can arise without regularization in linear models.

Conclusions and Future Directions

We find that both linear regression techniques and tree-based models can improve 10-hour forecasts (scored by R^2) of winds on Mt. Washington by about 0.1-0.15 [R^2] compared to the single best predictor. In the linear model case, while the all-feature OLS model has the best validation R^2, its extremely large coefficients suggest that we should use a regularized model instead: thus, we choose an optimal ridge regression $\lambda = 5$ model which has a test R^2 of 0.85. For the tree-based models, parameter tuning yields an optimal validation-R^2 (and similar test R^2) of 0.846 with a tuned RandomForestRegressor. Lastly, high winds are significantly harder to accurately predict for models (optimal validation R^2 of 0.559 for lasso regression), but significant improvement over single-feature model predictions is still possible. We note that there are some limitations to this analysis - not all vertical levels of the model were considered, and there may be some other model or observational features that are better predictors. Nevertheless, this model could be used to relatively confidently, even if only slightly, improve short-term wind speed predictions for Mt. Washington (relative to predictions made from only surface and 800mb model output), and thus, perhaps after some further improvements, may be useful to forecasters. Lastly, we note that both linear and tree-based models indicate that model-forecast wind speeds are, by far, the most important predictors. This is a testament to the accuracy of forecast models (even if they do not *forecast* the wind speed at the summit well using their surface or 800mb wind fields, model output is well-*correlated* with the summit wind speed.)

Moving forward, similar techniques could be applied to variables that could perhaps be less-accurately forecasted by models themselves. For example, cloud cover forecasts from weather models are generally quite variable (even for short-term forecasts) and are likely challenging in environments where topography may induce local-scale cloud cover changes. Application of machine learning techniques to prediction of cloud cover, ice accumulation, precipitation, or other perhaps harder-to-forecast variables could prove to be an interesting extension of this work, and one in which machine learning tools may be even more valuable.

Weapon of Math Destruction

Given the nature of our data involving weather and meteorology, our models are — in no sense — Weapons of Math Destruction. Indeed, our meteorological data was collected by reputable organizations, and there are no ethical concerns regarding the data collection process. Similarly, our data does not use any kind of proxy methods for analysis and predictions that may cause concern regarding model fairness and bias. Our models are also safe from creating any kind of "defeating feedback loop", since they would simply be used to improve forecasting (O'Neil, 2017).

References

3.3. Metrics and scoring: Quantifying the quality of predictions. (n.d.). Retrieved May 10, 2024, from https://scikit-learn/stable/modules/model_evaluation.html

Blaylock, B. (2024, March). Herbie: Retrieve NWP Model Data — Herbie 2024.3.0 documentation. Retrieved March 18, 2024, from https://herbie.readthedocs.io/en/stable/

Feature importances with a forest of trees. (n.d.). Retrieved May 11, 2024, from https://scikit-learn/stable/auto_examples/ensemble/plot_forest_importances.html

He, H. (2024, April). L17&L18: Regularization. https://canvas.cornell.edu/courses/62820/files/10318062?module_item_id=2534854

Higher Summits Forecast. (2024, March). Retrieved March 18, 2024, from https://mountwashington.org/weather/higher-summits-forecast/

IEM :: Download ASOS/AWOS/METAR Data. (n.d.). Retrieved March 18, 2024, from https://mesonet.agron.iastate.edu/request/download.phtml?network=NH_ASOS

National Oceanic and Atmospheric Administration. (2024, March). NOAA High-Resolution Rapid Refresh (HRRR) Model. https://registry.opendata.aws/noaa-hrrr-pds/

O'Neil, C. (2017). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group.

Young, T. (2017, November). International Standard Atmosphere (ISA) Table [Section: A _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118534786.app1]. In *Performance of the Jet Transport Airplane: Analysis Methods, Flight Operations and Regulations* (pp. 583–590). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118534786.app1

Further notes regarding references: Ideas for some techniques (i.e., plotting the correlation between features and labels within the ***Exploratory Data Analysis*** section) were influenced by looking at some past ORIE4741 projects on GitHub. Documentation for the Python packages used in the project (i.e., scikit-learn) was used when writing the code for models and data analysis.

Appendix

Additional Note

This project shares some similarity to a project that I (Ben Moose) worked on during a summer internship last semester - using weather model data to try to predict turbulence. However, this project uses new models, new station observation datasets, and feature transformation methods to answer a different meteorological question. Thus, it is substantially different in content than my past project, excluding the data acquisition process for NWP data.

Linear Models Figures

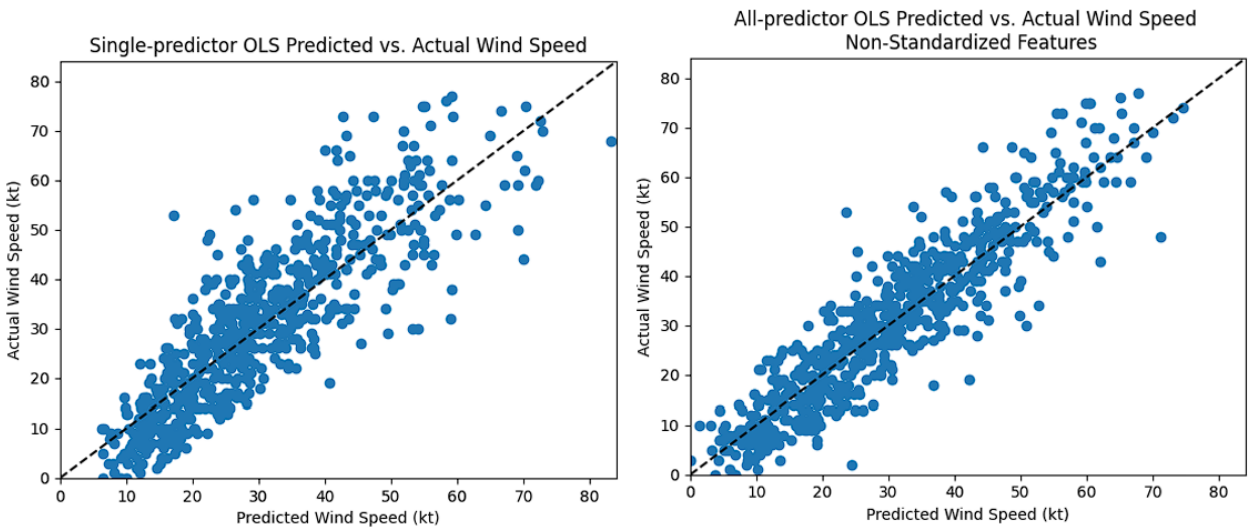

Figure 4: Scatterplots of OLS-predicted wind speed vs. actual wind speed for single-predictor OLS (using only ws8001 feature) [left] and all-feature OLS [right]. Dashed black line indicates perfect 1:1 fit.

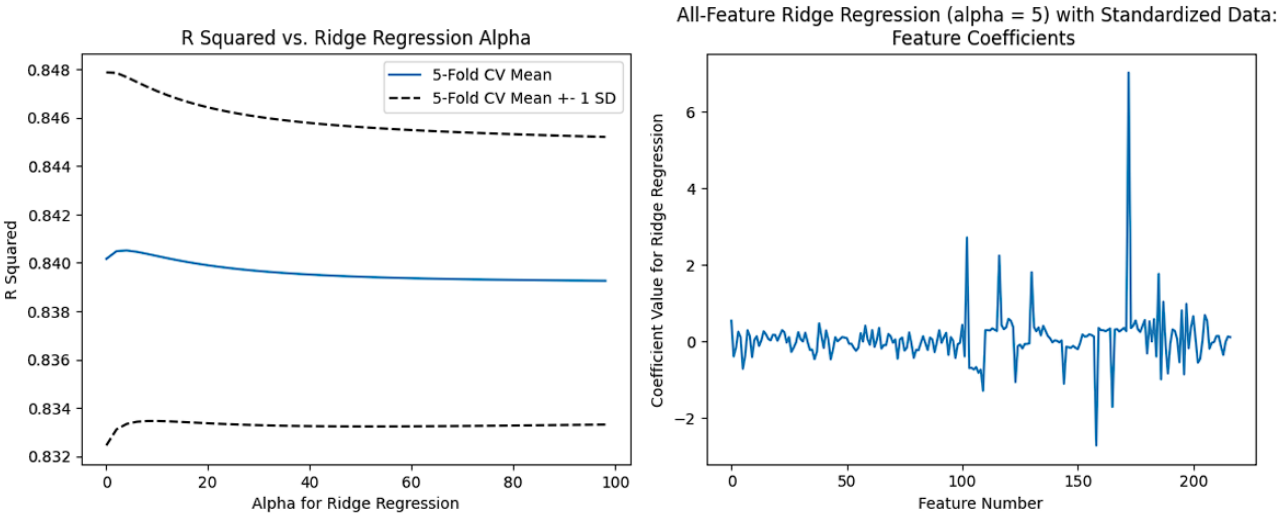

Figure 5: [Left] Mean and +/- 1SD R^2 from 5-fold cross-validation performed on training + validation dataset as a function of ridge regularization parameter α. [Right] Coefficient values corresponding to each feature from ridge regression (fit to only training set) with $\alpha = 5$. No wind speed filtering applied to dataset.

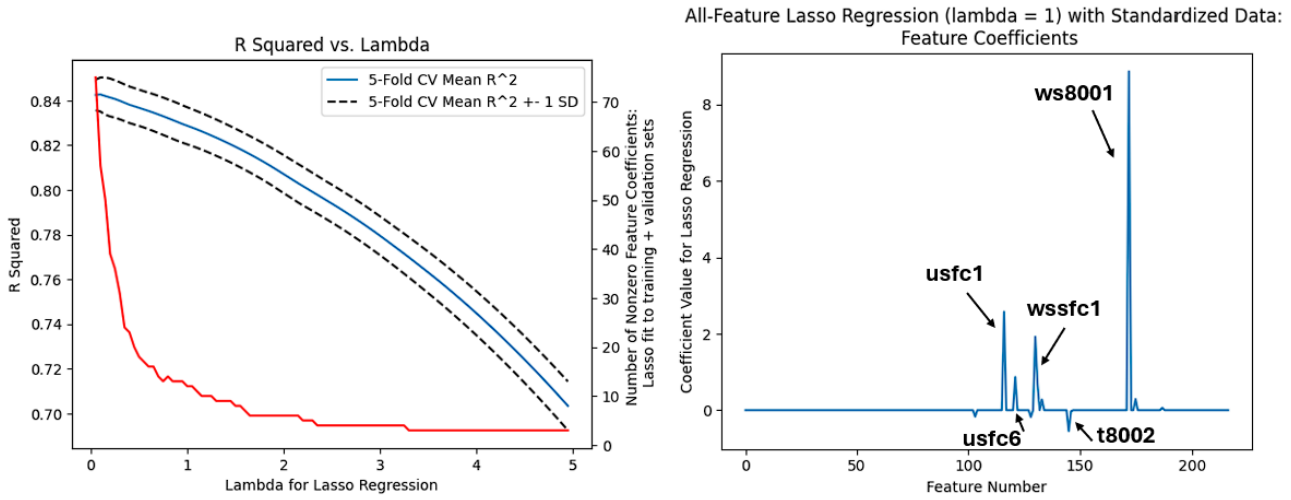

Figure 6: [Left] Mean and +/- 1SD R^2 from 5-fold cross-validation performed on training + validation dataset as a function of lasso regularization parameter λ. Red line (right axis) indicates number of nonzero feature coefficients in model trained on train+validation sets. [Right] Coefficient values corresponding to each feature from lasso regression (fit to only training set) with $\lambda = 1$. No wind speed filtering applied to dataset.

Tree-Based Models Figures

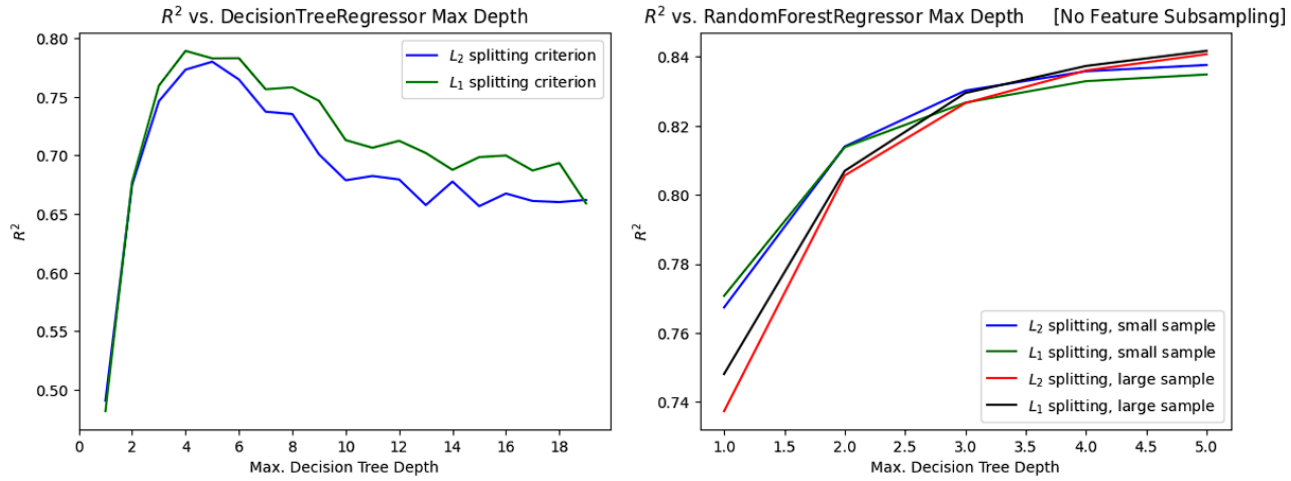

Figure 7: [Left] Parameter tuning results for adjusted max. tree depth and splitting criterion for DecisionTreeRegressor (Experiment 1). [Right] Parameter tuning results for adjusted max. tree depth, splitting criterion, and max_samples (0.2 = small, 0.7 = large) for bagged trees (Experiment 2).

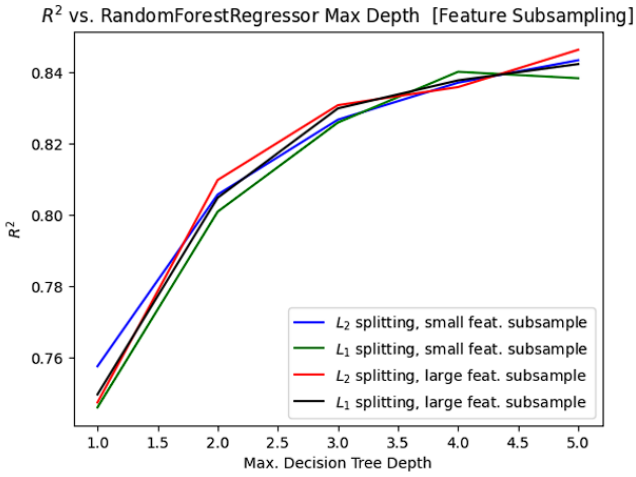

Figure 8: Parameter tuning results for adjusted max. tree depth, splitting criterion, and max_features (0.2 = small, 0.7 = large) for RandomForestRegressor (Experiment 3).

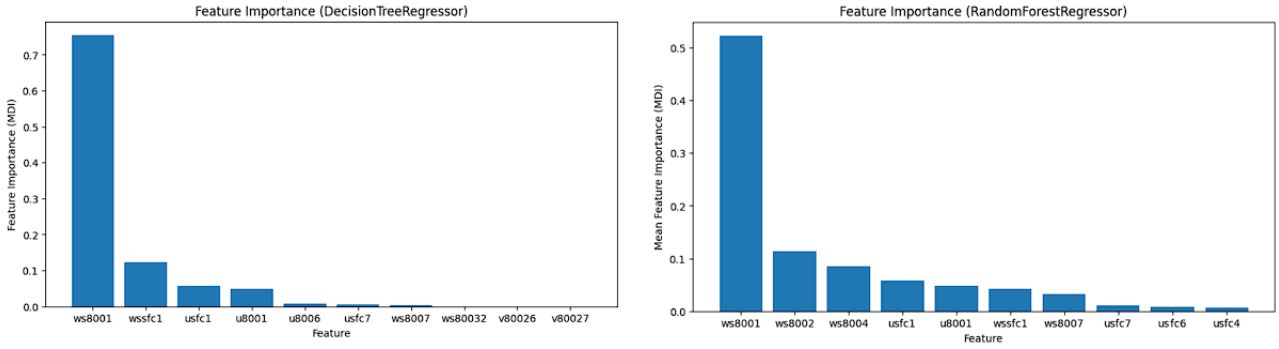

Figure 9: Mean Decrease Impurity (feature importance) for top-10 most important features from optimal-parameter (see Table 5) DecisionTreeRegressor and RandomForestRegressor ("Feature importances with a forest of trees", n.d.)

Predicting High Winds Figures

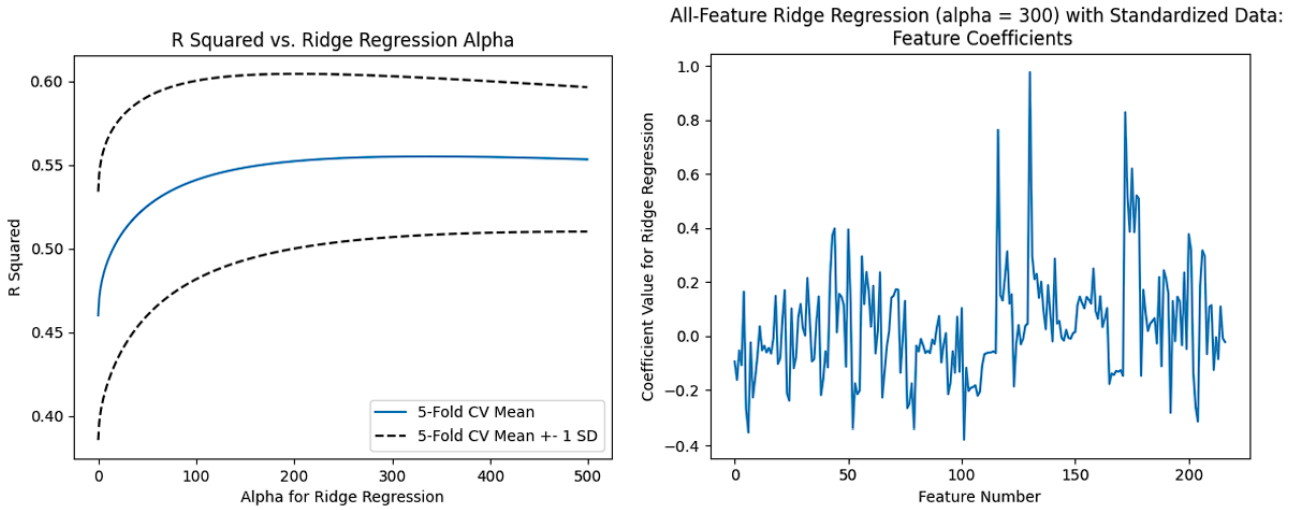

Figure 10: [Left] Mean and +/- 1SD R^2 from 5-fold cross-validation performed on training + validation dataset as a function of ridge regularization parameter α. [Right] Coefficient values corresponding to each feature from ridge regression (fit to only training set) with $\alpha = 300$. Models created/evaluated on dataset filtered to select wind speeds $>$ 40kt.

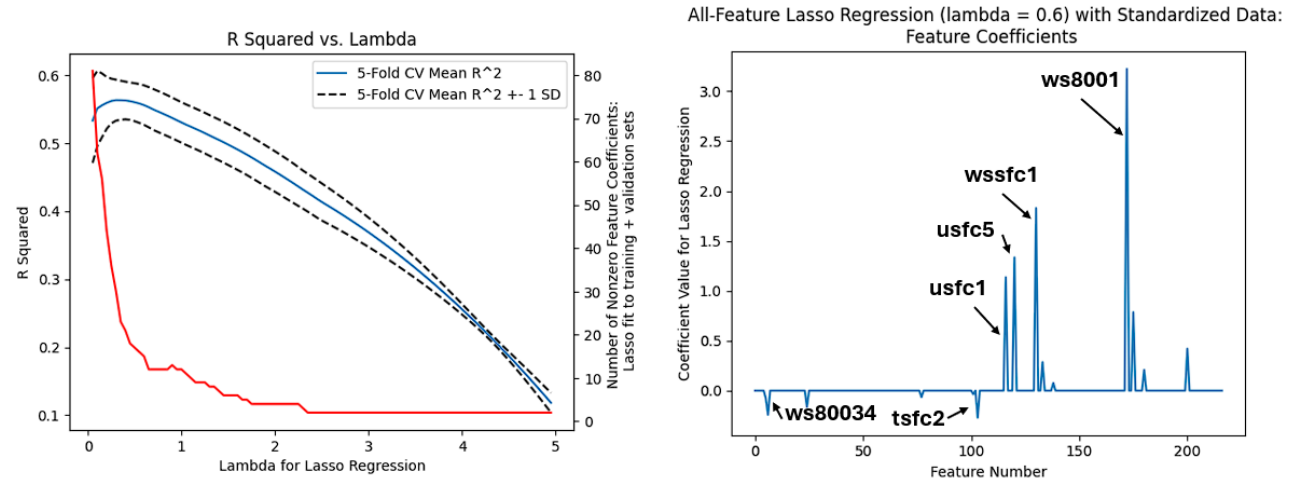

Figure 11: [Left] Mean and +/- 1SD R^2 from 5-fold cross-validation performed on training + validation dataset as a function of lasso regularization parameter λ. Red line (right axis) indicates number of nonzero feature coefficients in model trained on train+validation sets. [Right] Coefficient values corresponding to each feature from lasso regression (fit to only training set) with $\lambda = 0.6$. Models created/evaluated on dataset filtered to select wind speeds $>$ 40kt.