



Brazil Weather Watch: Monitoring Climate Change Through Temperature Forecasting

Group:

Alexander Kapustin

Muhammad Reza

Samuel Pang Shao Heng

Zhanyi Qiu

Introduction: The Amazon Basin

- The Amazon Basin primarily resides in Brazil, sustaining crucial ecosystems and supporting 30+ million people
- Brazil is the largest greenhouse gas emitter in Latin and South America
- They are working towards emission reduction strategies, significant progress in lowering deforestation rates, preserving the region's biodiversity
- Has led to the mitigation of climate change and the sustainability of life in the area



Source: National Geographic Education

Introduction: The Problem

- The Brazilian Amazon experienced a surge in fires in 2023, worsened by a severe drought causing historically low water levels in rivers and lakes.
- Amazonas, the country's largest state, saw a record 3,181 fires from October 1, 2023, double the number from the same period in 2022.
- Manaus, the capital city, has been blanketed in thick smoke, exacerbating respiratory issues and leading to increased medical emergencies.
- Researchers caution that fires now pose the primary threat to the Amazon, undermining environmental protection efforts.
- The fires, fueled by human activities and aggravated by severe drought, are expected to persist in the coming months.
- Beyond environmental harm, severe humanitarian crises have arisen, with communities facing water and food shortages and wildlife experiencing mass casualties.
- Climate change exacerbates these challenges, underscoring the urgent need for coordinated action to mitigate risks to the Amazon Rainforest and its inhabitants.



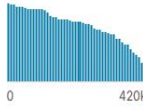
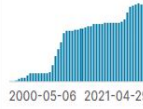
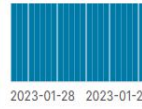
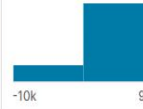
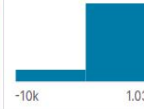



Introduction: The Purpose

- Investigate the impact of global warming on the weather patterns in several regions in Brazil throughout the years and develop a predictive model capable of forecasting future weather conditions.
- By forecasting temperature variations, our goal is to track the progression of global warming and offer valuable insights to stakeholders, including large corporations, the general public, and policymakers.
- Our findings can enable stakeholders to make well-informed decisions concerning policies, investments, and mitigation strategies to address the impacts of climate change in Brazil.

DATASET

- Hourly weather data in Brazil between 2000 and 2021 (via Kaggle)
- Data collected from 623 weather stations across Brazil
- 27 total attributes:
 - Locational attributes (latitude, longitude)
 - Time-based attributes (date, hour)
 - Environmental attributes (air pressure, temperature)
- 8,392,319 rows of data (relatively large dataset)
- Target Variable: airTemp, as we are predicting future temperatures

Data Card	Code (21)	Discussion (6)	Suggestions (0)		
<div>DetailCompactColumn</div>					
10 of 27 columns					
# index	Data	Hora	# PRECIPITAÇÃO TOT...	# PRESSAO ATMOSFE...	# PRESS/
					
0	420k	2000-05-062021-04-29	2023-01-282023-01-29	-10k96	-10k1.03k-10k
139001	2017-12-20	17:00	0.0	897.7	898.6
139002	2017-12-20	18:00	0.0	897.0	897.7
139003	2017-12-20	19:00	0.0	896.3	897.0
139004	2017-12-20	20:00	0.0	895.8	896.3
139005	2017-12-20	21:00	0.0	896.1	896.1
139006	2017-12-20	22:00	0.0	896.6	896.6
139007	2017-12-20	23:00	0.0	897.1	897.2
139008	2017-12-21	00:00	-9999.0	-9999.0	-9999.0
139009	2017-12-21	01:00	-9999.0	-9999.0	-9999.0
139010	2017-12-21	02:00	0.0	898.3	898.4
139011	2017-12-21	03:00	0.0	897.9	898.3
139012	2017-12-21	04:00	0.0	897.5	897.9
139013	2017-12-21	05:00	0.0	897.2	897.5
139014	2017-12-21	06:00	0.0	896.9	897.2
139015	2017-12-21	07:00	0.0	896.9	897.0

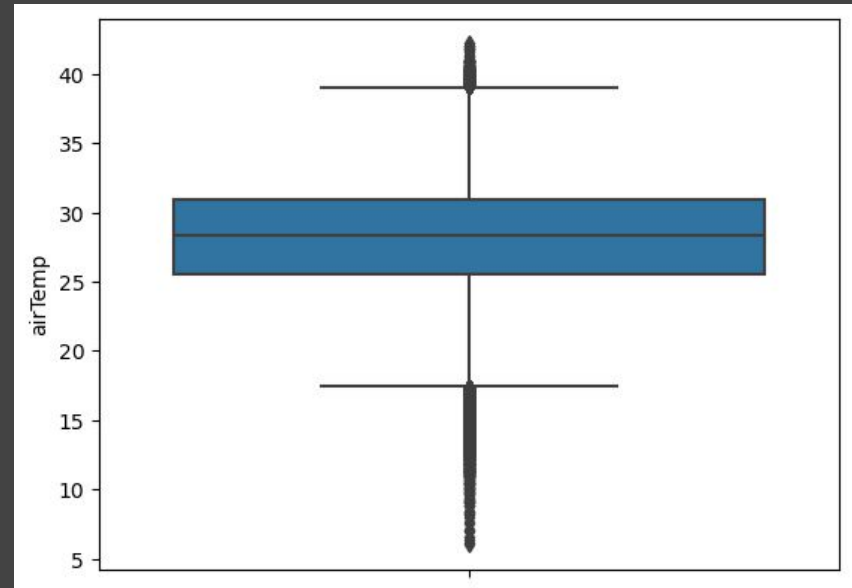
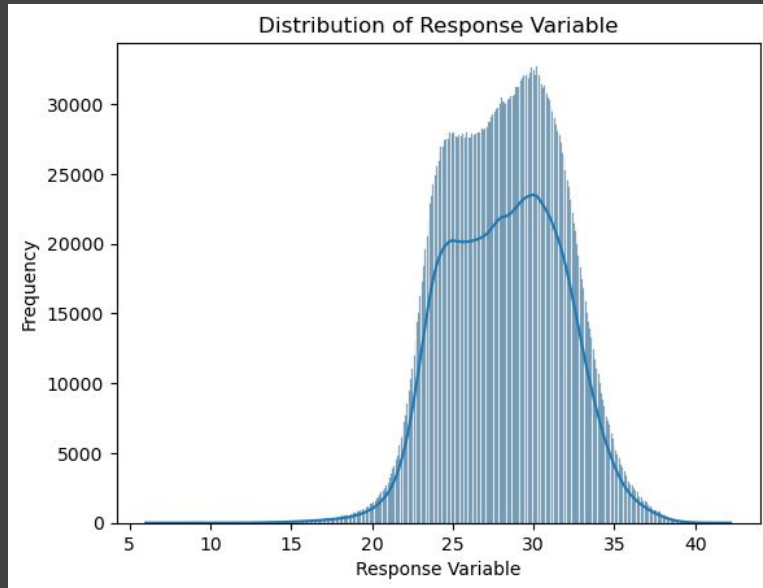
Variables and their meanings

Variable Abbreviation	Meaning
idx	row index
date	(YYYY-MM-DD)
hour	(HH:00)
tPrec	Amount of precipitation in millimetres (last hour)
atmosPStatn	Atmospheric pressure at station level (mb)
prevHrPmax, prevHrPmin	Max/Min air pressure for the last hour (mb)
rad	Solar Radiation (KJ/m2)
<u>airTemp</u>	Air temperature (instant) (°c)
dpTemp	Dew point temperature (instant) (°c)
prevHrMaxTemp, prevHrMinTemp	Max/Min temperature for the last hour (°c)
prevHrMaxDpTemp, prevHrMinDpTemp	Max/Min dew point temperature for the last hour (°c)

Variable Abbreviation	Meaning
prevHrMaxHum, prevHrMinHum	Max/Min relative humid temperature for the last hour (%)
airHum	Relative humidity (% instant)
windDir	Wind direction (radius degrees (0-360))
maxWindSp	Wind gust in metres per second
windSp	Wind speed in metres per second
reg	Brazilian geopolitical regions
state	State (Province)
statn	Station Name
statnCode	Station code
lat	Latitude
long	Longitude
height	Elevation

Data visualization

- We can visualize the distribution of the response variable, airTemp, using a histogram and a boxplot as shown below.
- Significant proportion of the temperatures lie between 25 - 32 degrees.



Data Preprocessing

Data removal:

- NaNs are represented as -9999, thus we converted these values to NaNs and removed rows which had this value.
- (lat, long, height) are variables sufficient to identify a specific location or station, thus, variables such as statnCode, statn, reg and state are removed.

Data Conversion:

- Variables date and hour are considered categorical variables as shown on the right. Conversion to numerical would be ideal, which was done by mapping date_time to a numerical value.

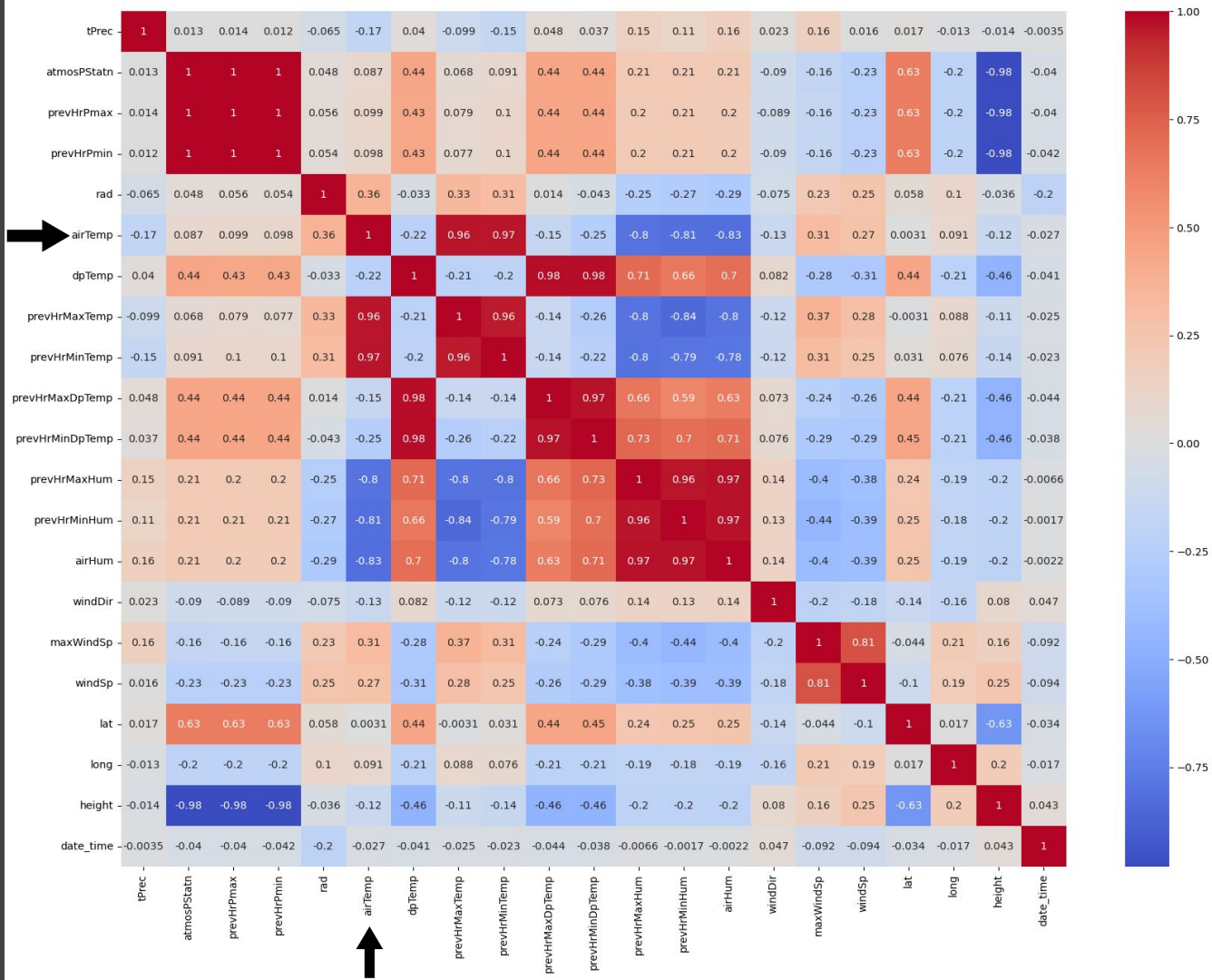
Column Renaming:

- Since the column names were in Portuguese, they were renamed for comprehensibility.

```
RangeIndex: 8392320 entries, 0 to 8392319
Data columns (total 27 columns):
#   Column              Dtype
---  -
0   idx                 int64
1   date                object
2   hour                object
3   tPrec               float64
4   atmosPStatn        float64
5   prevHrPmax          float64
6   prevHrPmin          float64
7   rad                 int64
8   airTemp             float64
9   dpTemp              float64
10  prevHrMaxTemp        float64
11  prevHrMinTemp        float64
12  prevHrMaxDpTemp      float64
13  prevHrMinDpTemp      float64
14  prevHrMaxHum          int64
15  prevHrMinHum          int64
16  airHum                int64
17  windDir               int64
18  maxWindSp             float64
19  windSp                float64
20  reg                   object
21  state                 object
22  statn                 object
23  statnCode             object
24  lat                   float64
25  long                  float64
26  height                float64
```

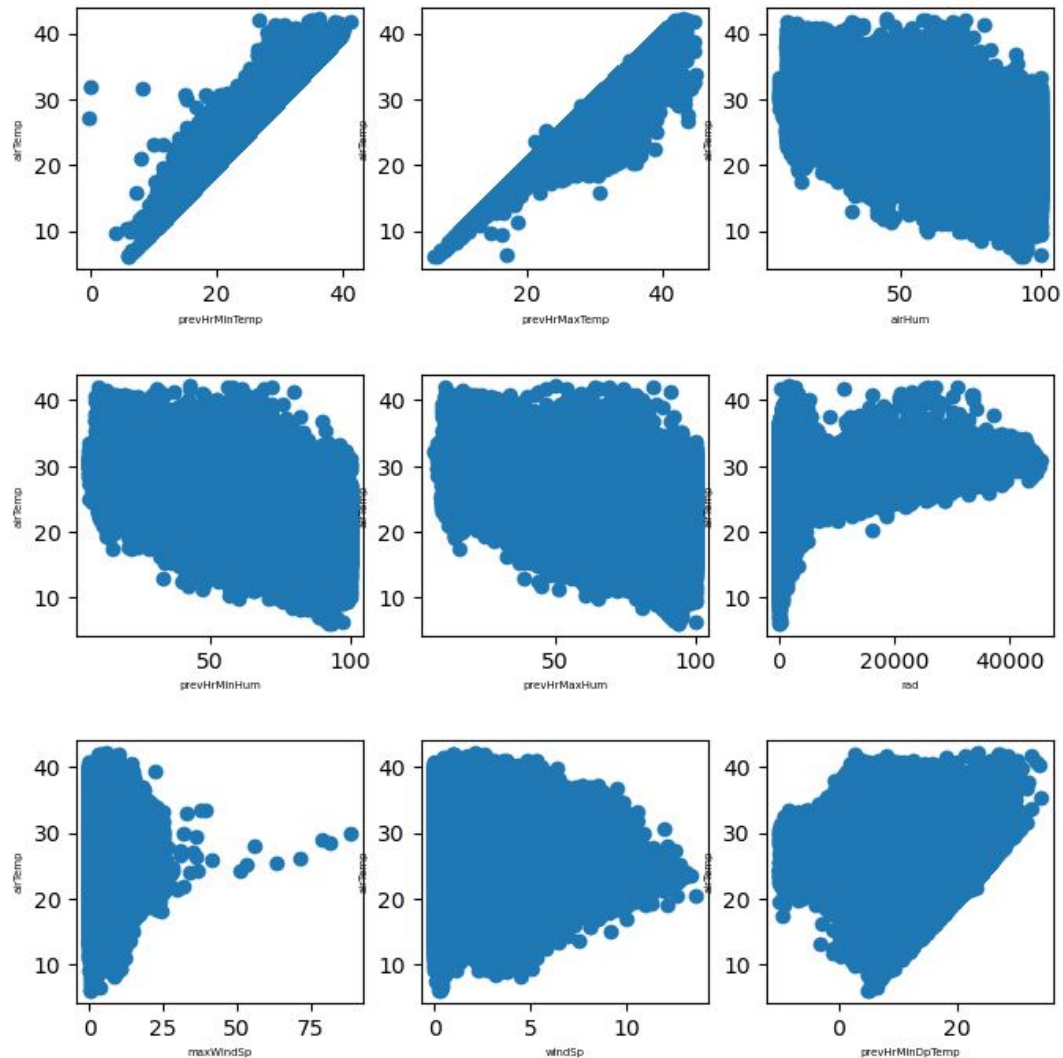

Correlation Heatmap:

- The more intense the color, the greater the correlation
- Atmospheric pressure at the station level perfectly correlated with prevHrPmin and prevHrPmax (corr = 1)
- Many other highly correlated variables (corr >= 0.8)
- We will seek to drop variables that has a correlation value of more than 0.8 with another, leaving 11 predictor vars and 1 response var



Scatter plots (most correlated with response var)

- The magnitude of correlation decreases as we go down for each row, column.
- We can see that especially for the first two graphs, which are prevHrMinTemp and prevHrMaxTemp respectively, there is a relatively linear relationship between these variables and airTemp.



Train Test Split

Goal	Problem	Solution
<ul style="list-style-type: none">- Conduct a time series analysis: Want to train the model based on preceding data to forecast later data- Sought to achieve a 70/30 train test split	<ul style="list-style-type: none">- Data was sorted by statn or (lat, long), not by dates- Each unique (lat, long) does not have the same number of data	<ul style="list-style-type: none">- Grouped data by (lat, long) and sort the data by date_time, before conducting ordered train_test_split for each group- This ensure that all dates in the test set are after those in the training set for each (lat, long)- Ensures we are predicting future temperatures



Forward and Backward Selection

Forward selection

```
selected_features_forward, count_forward = forward_selection(X_train, y_train)
print("Selected features by forward selection:", selected_features_forward)
print("Number of features selected:", count_forward)
```

Backward elimination

```
selected_features_backward, count_backward = backward_elimination(X_train, y_train)
print("Selected features by backward elimination:", selected_features_backward)
print("Number of features selected:", count_backward)
```

Selected features by forward selection: ['atmosPStatn', 'lat', 'windDir', 'rad', 'tPrec', 'dpTemp', 'airHum', 'prevHrMinTemp', 'long', 'windSp']

Number of features selected: 10

Selected features by backward elimination: ['tPrec', 'atmosPStatn', 'rad', 'dpTemp', 'prevHrMinTemp', 'airHum', 'windDir', 'windSp', 'lat', 'long']

Number of features selected: 10

- The forward and backward stepwise selection methods are returning 10 variables as significant, it suggests that each variable in our dataset might be contributing meaningfully to the prediction of air temperature.



Optimal Alpha

- We use K-fold cross validation to find the optimal alpha for the Ridge and Lasso regression model.
- For K-fold cross validation, we set range for the alpha from 0.001 to 10.0.

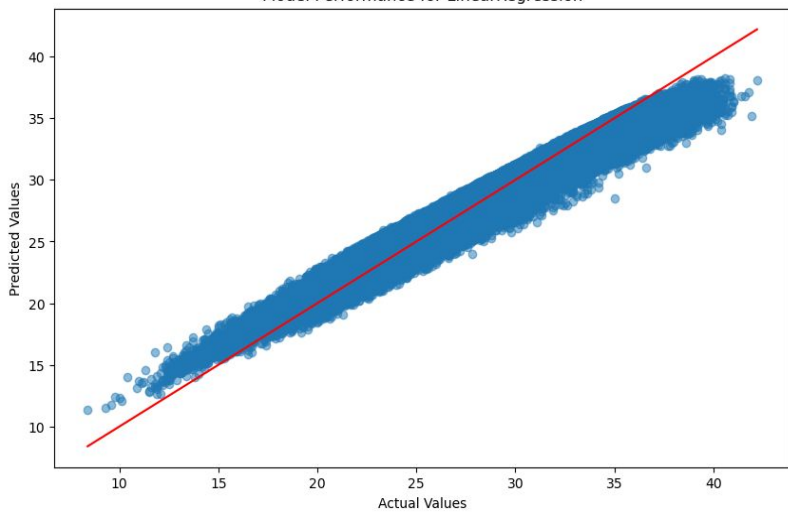
```
Ridge_model = Ridge()
optimal_alpha_ridge = k_fold_cv(Ridge_model, X_train, y_train)
print("Best alpha for Ridge Regression:", optimal_alpha_ridge)
```

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best alpha for Ridge Regression: {'alpha': 10.0}
```

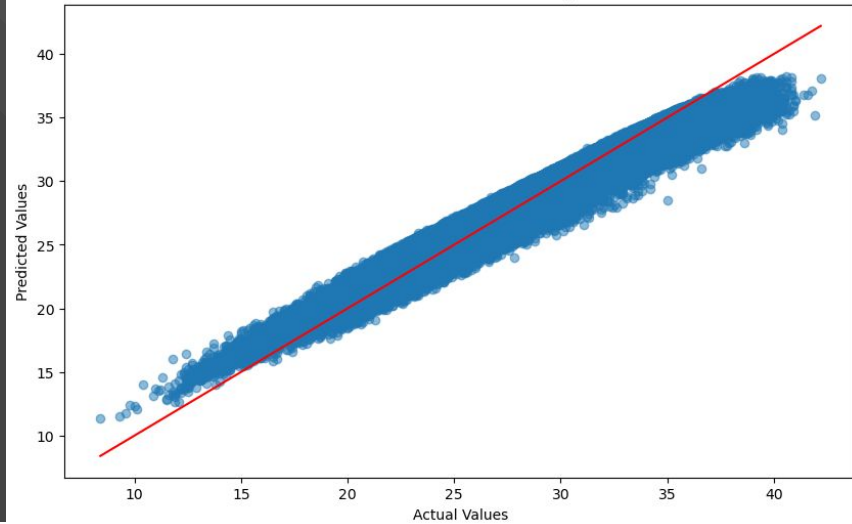
```
Lasso_model = Lasso()
optimal_alpha_lasso = k_fold_cv(Lasso_model, X_train, y_train)
print("Best alpha for Lasso Regression:", optimal_alpha_lasso)
```

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
Best alpha for Lasso Regression: {'alpha': 0.001}
```

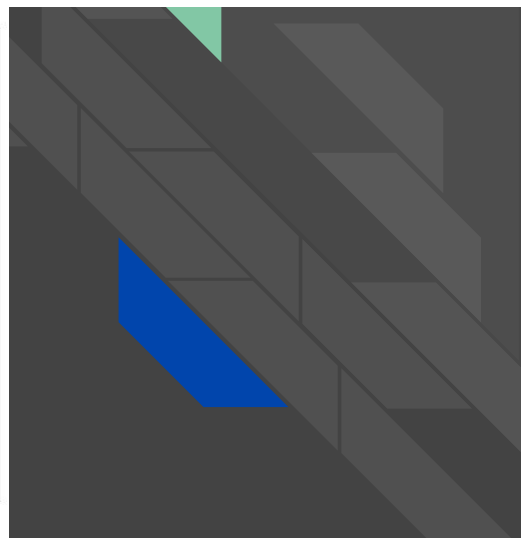
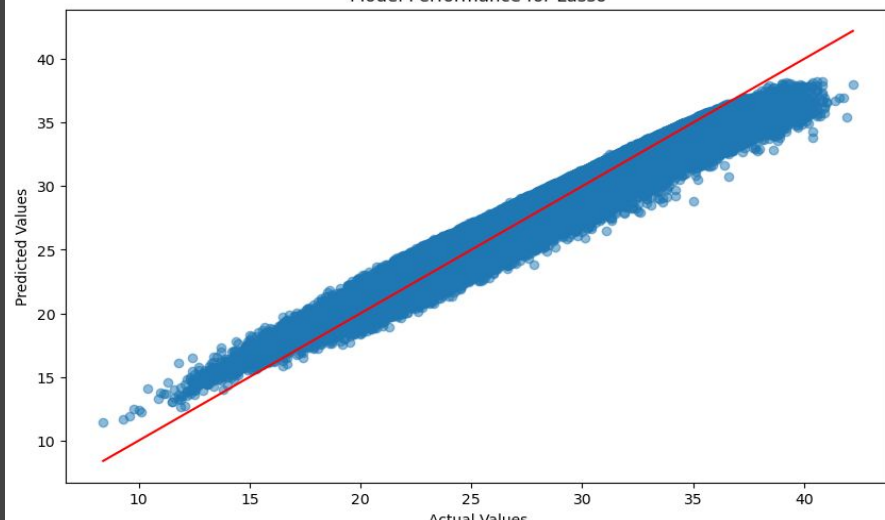
Model Performance for LinearRegression



Model Performance for Ridge



Model Performance for Lasso





Regression Model Selection

```
print("Normalized Linear Regression - MSE:", Nor_linear_mse, "MAE:", Nor_linear_mae, "R2:", Nor_linear_r2)
print("Normalized Ridge Regression - MSE:", Nor_ridge_mse, "MAE:", Nor_ridge_mae, "R2:", Nor_ridge_r2)
print("Normalized Lasso Regression - MSE:", Nor_lasso_mse, "MAE:", Nor_lasso_mae, "R2:", Nor_lasso_r2)
```

```
Normalized Linear Regression - MSE: 0.36822938764092905 MAE: 0.44364334283269685 R2: 0.9724557191078737
Normalized Ridge Regression - MSE: 0.3682269699420409 MAE: 0.4436451123176414 R2: 0.9724558999564956
Normalized Lasso Regression - MSE: 0.3696850463836768 MAE: 0.44797478038145383 R2: 0.9723468329770024
```

- From the above result, we can see that all three model are perform very similar.
- The Linear regression model and the Ridge regression model have almost the same Mean Squared Error(MSE), Mean Absolute Error(MAE) and R-squared (R2) score. Meanwhile, the Lasso regression has a slightly higher MSE and slightly lower R2.
- The difference between the three model are very minimal.
- Considering we are predicting air temperature with many features, Lasso Regression might be more suitable for this dataset.

Polynomial Regression (Using ANOVA) 1/3

Determining the optimal polynomial degree for each predictor variable in the train set through the ANOVA approach:

- The variables "*lat*", "*long*", and "*prevHrMinTemp*" show negligible changes in the RSS across the 5 polynomial degrees, suggesting that degree 1 suffices;
- For the variables "*windSp*", "*atmosPStatn*", and "*tPrec*" the RSS remains relatively similar from degrees 2 to 5, suggesting that degree 2 should be chosen to prevent overfitting.
- For the variables "*dpTemp*" and "*windDir*", the RSS remains relatively constant from degrees 3 to 5, suggesting that degree 3 should be chosen to prevent overfitting.
- For the variable "*airHum*", degrees 4 and 5 show the best performance, suggesting that degree 4 is preferable to avoid overfitting. Similarly, for the variable "*rad*", degrees 4 and 5 demonstrate the best performance, indicating that degree 4 is preferable to avoid overfitting.

```
poly_airHum1 <- lm(y_train ~ airHum, data = X_train)
poly_airHum2 <- lm(y_train ~ poly(airHum, 2), data = X_train)
poly_airHum3 <- lm(y_train ~ poly(airHum, 3), data = X_train)
poly_airHum4 <- lm(y_train ~ poly(airHum, 4), data = X_train)
poly_airHum5 <- lm(y_train ~ poly(airHum, 5), data = X_train)
anova(poly_airHum1, poly_airHum2, poly_airHum3, poly_airHum4, poly_airHum5)
```

```
## Analysis of Variance Table
##
## Model 1: y_train ~ airHum
## Model 2: y_train ~ poly(airHum, 2)
## Model 3: y_train ~ poly(airHum, 3)
## Model 4: y_train ~ poly(airHum, 4)
## Model 5: y_train ~ poly(airHum, 5)
##      Res.Df      RSS Df Sum of Sq      F      Pr(>F)
## 1 2286283 9158342
## 2 2286282 8379938  1    778404 215570.1 < 2.2e-16 ***
## 3 2286281 8355129  1     24809   6870.4 < 2.2e-16 ***
## 4 2286280 8268243  1     86887  24062.3 < 2.2e-16 ***
## 5 2286279 8255543  1     12700   3517.0 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The variable "*airHum*," degrees 4 and 5 demonstrate the best performance, implying that degree 4 is preferable to avoid overfitting.

Polynomial Regression (Using ANOVA) 2/3

RESULTS:

```
# MSE
mse <- mean((y_test - test_predictions)^2)
# MAE
mae <- mean(abs(y_test - test_predictions))

paste("The Mean Squared Error for the model is: ", mse)

## [1] "The Mean Squared Error for the model is: 0.0156540026387341"

paste("The Mean Absolute Error for the model is: ", mae)

## [1] "The Mean Absolute Error for the model is: 0.0888964497745386"

# Predictions for the test set using the poly regression model
test_predictions <- predict(poly_final, newdata = X_test)
# Accuracy of the model on the test set
test_accuracy <- sqrt(mean((y_test - test_predictions)^2))
mean_y_test <- mean(y_test)
paste("The model's predictions are off by ", (test_accuracy/mean_y_test)*100, "%")

## [1] "The model's predictions are off by 0.448997396131852 %"
```

```
## Call:
## lm(formula = y_train ~ poly(lat, 1) + poly(long, 1) + poly(prevHrMinTemp,
## 1) + poly(tPrec, 2) + poly(atmosPStatn, 2) + poly(windSp,
## 2) + poly(windDir, 3) + poly(dpTemp, 3) + poly(airHum, 4) +
## poly(rad, 4), data = X_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.9925  -0.0726  -0.0031   0.0694  14.4240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.839e+01  8.568e-05 331373.290 < 2e-16 ***
## poly(lat, 1)    -5.802e+00  2.233e-01  -25.977 < 2e-16 ***
## poly(long, 1)   -2.980e+00  1.423e-01  -20.941 < 2e-16 ***
## poly(prevHrMinTemp, 1) 1.307e+02  5.604e-01  233.161 < 2e-16 ***
## poly(tPrec, 2)1    -2.218e+00  1.351e-01  -16.415 < 2e-16 ***
## poly(tPrec, 2)2     3.638e-01  1.314e-01    2.769  0.00561 **
## poly(atmosPStatn, 2)1  7.463e+00  2.015e-01   37.035 < 2e-16 ***
## poly(atmosPStatn, 2)2 -2.662e+00  1.668e-01  -15.956 < 2e-16 ***
## poly(windSp, 2)1    -7.795e+00  1.589e-01  -49.056 < 2e-16 ***
## poly(windSp, 2)2    -4.888e+00  1.368e-01  -35.729 < 2e-16 ***
## poly(windDir, 3)1   -8.591e-01  1.368e-01   -6.278 3.43e-10 ***
## poly(windDir, 3)2   -7.879e-01  1.328e-01   -5.932 2.99e-09 ***
## poly(windDir, 3)3    1.457e+00  1.341e-01   10.869 < 2e-16 ***
## poly(dpTemp, 3)1     5.479e+03  7.414e-01  7389.964 < 2e-16 ***
## poly(dpTemp, 3)2    -1.252e+02  2.032e-01  -616.469 < 2e-16 ***
## poly(dpTemp, 3)3    -2.618e+01  1.497e-01  -174.854 < 2e-16 ***
## poly(airHum, 4)1    -8.179e+03  9.884e-01  -8274.637 < 2e-16 ***
## poly(airHum, 4)2     1.855e+03  3.315e-01  5596.586 < 2e-16 ***
## poly(airHum, 4)3    -5.153e+02  1.871e-01  -2754.308 < 2e-16 ***
## poly(airHum, 4)4     1.878e+02  1.541e-01  1218.370 < 2e-16 ***
## poly(rad, 4)1       1.321e+01  1.522e-01   86.777 < 2e-16 ***
## poly(rad, 4)2      -1.869e+01  1.712e-01  -109.153 < 2e-16 ***
## poly(rad, 4)3       1.135e+01  1.468e-01   77.331 < 2e-16 ***
## poly(rad, 4)4      -4.507e+00  1.359e-01  -33.171 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1296 on 2286261 degrees of freedom
## Multiple R-squared: 0.9987, Adjusted R-squared: 0.9987
## F-statistic: 7.493e+07 on 23 and 2286261 DF, p-value: < 2.2e-16
```

Polynomial Regression

(Using ANOVA) 3/3

SUMMARY:

- The results for the Polynomial Regression Model showed that the optimal polynomial degree varied across variables. For instance, variables like air humidity and solar radiation exhibited the best performance at degrees 4 and 5, while variables such as precipitation, atmospheric pressure, and wind speed displayed optimal performance at lower polynomial degrees, suggesting that degree 2 or 3 should be selected to prevent overfitting.
- The final polynomial regression model incorporated the selected polynomial degrees for each variable. Evaluation on the test set revealed a low percentage error, indicating that the model's predictions were relatively close to the actual values. The mean squared error (*MSE*) and mean absolute error (*MAE*) further confirmed the model's performance, with both metrics indicating relatively low errors.
- In conclusion, the developed polynomial regression model demonstrates promising predictive performance for air temperature estimation based on environmental variables. The findings underscore the importance of selecting an appropriate polynomial degree for each predictor variable to balance model complexity and generalization. Further refinements and validations could enhance the model's robustness and applicability in real-world scenarios.



Random Forests

- Differs from linear or polynomial regression as it is a non-parametric model
- We will generate 100 trees to fit the model, but we will limit the number of features to be taken into account to be 6.
- Achieves a lower MSE of 0.036 as well as a lower MAE of 0.11 compared to linear, lasso and ridge regression, but has higher MSE and MAE than polynomial regression.
- Comparing R-squared values of non-parametric models against the R-squared values of parametric models may not be directly applicable or interpretable.

```
RF_weather = RF(max_features=6,  
                random_state=0).fit(X_train, y_train)  
y_hat_RF = RF_weather.predict(X_test)
```

```
print(np.mean((y_test - y_hat_RF)**2))  
print(np.mean(np.abs(y_test - y_hat_RF)))
```

✓ 0.0s

0.03643217300158907

0.10974133908689959



Approach improvements and limitations

Feature selection methodology limitations:

- Omission of some variables before training: Some variables that were omitted may have been relevant. We could try forward and backward selection on this set of variables rather than the limited set.
- Removal of variables that had correlation value above a threshold: Alternatively, we could attempt trying different threshold values to determine which features to omit.
- Best subset selection as an alternative to forward and backward selection on original full set of variables

Given the large size of the dataset, some of these feature selection methods may work for some models (e.g OLS), while some of the methods (best subset selection) may be computationally less feasible for others (e.g polynomial regression). There exists a trade off between dataset size and the type of feature selection method we can use.



Approach improvements and limitations

Data is not evenly distributed across the years due to disproportionate station data:

- Time series analyses are generally structured and proportioned evenly, where each “year” of data has an equivalent number of data points.
- Potentially inhibits the efficiency of an ordered train-test split in the context of time series analysis, where preceding data is used to forecast subsequent values.
- We accounted for this issue and were able to order the split successfully, but it is an interesting perspective that is worth mentioning.



Approach improvements and limitations

Limitations of large dataset on Ensemble Models (Bagging, Random Forests):

- Large computational training time: Random Forests with number of features set to 6 took approximately 25 mins, while training the bagging model was not feasible as the duration extended to 30+ mins without terminating
- Not feasible to conduct multiple fold cross-validation: Hyper parameter tuning via K-fold cross validation would not be feasible given the above large computational time

Limitations of large dataset on Polynomial Models:

- Selecting the polynomial degree through K-fold cross-validation is impractical: processing each variable typically takes an average of 10 minutes, and results vary with each iteration

Conclusion

In summary, after evaluating various machine learning techniques on the dataset, Polynomial Regression and Random Forest emerge as the top-performing models. Polynomial Regression slightly outperforms Random Forest by a narrow margin. Linear, Ridge, and LASSO Regression yield less satisfactory results. Therefore, we would not recommend considering them for predicting temperature variations in Brazil on new datasets. However, it's worth noting that because performing k-fold Cross-Validation on a large dataset like this requires substantial computing power, the models presented may yield different results if supercomputing resources were utilized for the analysis. With regards to our top-performing models, we believe they could be effectively utilized to predict the future weather in Brazil and inform those in power of potential decisions they could make to combat climate change and preserve the environment within the Amazon Basin.

References

<https://education.nationalgeographic.org/resource/amazon-rainforest/>

<https://news.mongabay.com/2023/10/people-and-nature-suffer-as-historic-drought-fuels-calamitous-amazon-fires/>

<https://www.kaggle.com/datasets/PROPPG-PPG/hourly-weather-surface-brazil-southeast-region/data>



Thank you!
We will be taking any Questions