Predicting Extreme Weather in Madrid

FEM11149 - Introduction to Data Science

Daphne, Riya, Vera, Zsofi

20 October, 2025

Introduction

Understanding the factors that influence daily maximum temperature in Madrid is essential for anticipating extreme weather events. Meteorological variables can provide valuable insights for predicting such events one day ahead, which is crucial for energy companies to manage demand, for urban planners to prepare infrastructure, and for public authorities to ensure safety. This analysis focuses on identifying key predictors and evaluating their predictive performance to support effective short-term forecasting and informed decision-making.

Data

The data were obtained from the Copernicus European Regional ReAnalysis (CERRA) project, which provides spatially and temporally consistent historical reconstructions of meteorological conditions across Europe. The dataset covers over three decades and includes multiple atmospheric and surface-level variables with high temporal and spatial resolution. These include measurements of temperature, wind speed and direction, humidity, cloud cover, precipitation, radiation, soil moisture, and surface fluxes, among others. The outcome of interest is the daily maximum temperature, which measures the highest air temperature at 2 meters above the surface within a 24-hour period. This indicator is central for anticipating extreme weather events and understanding variations in energy demand, infrastructure load, and public safety risks. In this report, the maximum temperature variable is analysed in its continuous form, allowing us to study how variations in meteorological variables are associated with differences in temperature extremes, as continuous values capture gradual changes and enable precise estimation of their effects.

Methodology

To predict over time, we organised the data so that each row used the weather from one day to predict the next day's temperature. Hence, the predictor variables (X) include all weather measurements from the current day, and the final observation was removed since there was no next day to predict. The target variable (y) is the maximum temperature on the following day. This formulation reflects the natural structure of forecasting problems and prevents the use of future information in model training. Before analysis, the dataset was cleaned and standardised to ensure comparability across features measured in different units.

A correlation analysis was conducted on the numeric variables to identify which features were most strongly associated with tomorrow's maximum temperature. This step is essential to investigate strong correlations among variables as that would indicate the presence of multicollinearity—one of the key motivations for applying Principal Component Analysis (PCA). To evaluate predictive performance on unseen data, the dataset was divided into a training set consisting first 80% of the observations and a testing set final 20%.

The data was split in a chronological order rather than random sampling since this data is time-ordered, which prevents future data from influencing model estimation, essential for realistic forecasting.

Principal Component Analysis was employed to reduce the dimensionality of the predictor space and to identify latent patterns underlying the meteorological variables. Criteria used to determine the optimal number of components: Cumulative Variance Explained to represent the total variance explained by principal components, by summing the variance of each component; Kaiser threshold of eigenvalues greater than 1 and Scree plot displaying eigenvalues (measure of the variance explained by each component) from largest to smallest on the y-axis against the component number on the x-axis.Principal component regression was used to predict tomorrow's maximum temperature. In this two-stage modelling process, PCA first transforms the correlated predictors into orthogonal principal components, and then a linear regression model is fitted using these components as explanatory variables. We use the number of components determined by the criteria for optimal number of components.

To benchmark the PCR results, a multiple linear regression model was trained using all original predictors without dimensional reduction. This model served as a baseline to assess whether PCA improved performance beyond a standard linear approach. Prediction results from both models were obtained on the testing set, and their performance was compared using three metrics: Root Mean Squared Error (RMSE) to capture the average magnitude of prediction errors, Mean Absolute Error (MAE) to measure the average absolute deviation from observed values, and R-squared to quantify the proportion of variance explained by the model. These metrics were selected to provide an overview of predictive accuracy, model fit, and reliability, providing a fair comparison of the two approaches and an assessment of whether there is a relative benefit of dimensional reduction through PCR. To assess model robustness, a sensitivity analysis was performed using less than, more than and the chosen amount of components. Lastly, to explore model behaviour across different temperature variations, the test observations were divided into ten deciles based on actual maximum temperature values. Mean Squared Error (MSE) was calculated within each decile for PCR models with different amounts of components and for the linear regression model to compare best performing for extreme weather conditions.

Results

The objective of the analysis is to predict tomorrow's maximum temperature (dependent variable) based on the meteorological observations of today (independent variables). First, the correlation analysis is performed to gain insight into the structure of the meteorological data. Figure 1 shows a strong positive correlation between today's and tomorrow's maximum temperature (r 'is approximately' 0.96). In contrast, Figure 2 shows the weak relationship between maximum wind gusts and tomorrow's temperature (r 'is approximately' 0.03). Overall, variables related to temperature show the strongest correlations, while precipitation variables are moderately negatively correlated, and finally wind and snowfall contribute little explanatory power. The full set of correlation plots and the correlation table can be found in the appendix. Since there are strong and similar patterns across the temperature and radiation predictors, it suggests the presence of multicollinearity. This is a key justification for applying PCA, as it reduces correlated variables into a smaller set of independent components, and thus removing multicollinearity and simplifying the model without losing essential information.

To perform PCA, the data was first split chronologically (80-20%) into training and testing data. Next the PCA was performed on the training set, and the number of principal components that will be used for analysis need to be determined based on three common criteria: cumulative variance explained, the Kaiser criterion (eigenvalues $>$ 1) and the scree plot 'elbow' method. In Figure 3 we can see that using three components would explain more than 80% of the variance of the original meteorological variables. Three components thus significantly reduce the dimensionality, suggesting three components for final analysis. In Figure 4 the scree plot shows an elbow after the third or fourth component and the eigenvalues drop below 1 after 4 components, indicating that beyond this point, each additional component explains less variance than a single original variable and thus contributes minimal new information to the model. In conclusion, four components are selected as they provide an optimal trade-off between explanatory power and model simplicity.

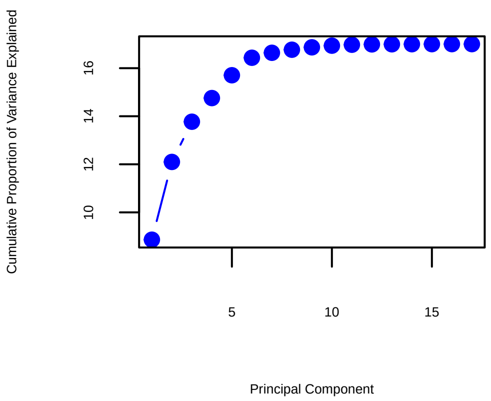


Figure 3: Cumulative Variance Explained by PC

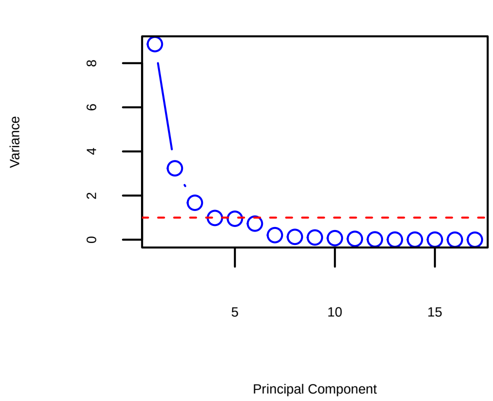
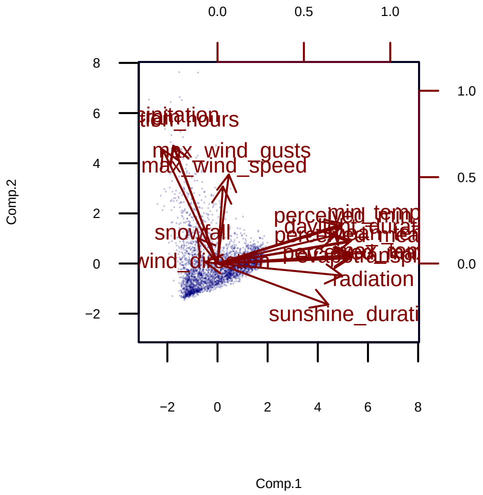


Figure 4: Scree Plot

With the number of principal components established, we next examine their structure and interpretability. The biplot in Figure 5 shows how the original variables contribute to the first two components. PC1 is dominated by variables related to temperature and radiation, as their arrows point strongly to the right along the x-axis, indicating a warm and sunny dimension.

Figure 5: Biplot (PC1−PC2)



Specifically, the loadings for PC1 (as shown in table 4 in the appendix) reveal that all temperature variables such as mean, maximum and minimum temperature (both actual and perceived), as well as radiation related variables, such as daylight duration, sunshine duration and radiation itself all contribute with loadings around 0.30. This suggests that these variables move together in an underlying pattern, with not one single variable driving the component. In the biplot we see that PC2 is more strongly related to precipitation and wind variables, with variables like precipitation and max wind gusts pointing upwards along the y-axis, indicating a wet and windy dimension. The dotted points represent individual days, showing their position based on PC1 and PC2. We can see the data varies gradually along the warm & sunny dimension and for the wet & windy dimension we see more variation towards the wetter and windier side. We further investigated the first principal component, as it alone explains 52% of the total variance. The narrow bootstrap confidence interval (0.517 to 0.529) for PC1 suggests this underlying pattern is robust and consistently shows across datasets that are resampled. These results and the bootstrap histogram can be found in the Appendix. A

Principal Component Regression (PCR) was fitted on the training set using the chosen four components, as well as a Multiple Linear Regression (MLR) with the original predictors to serve as a benchmark (see appendix table 5). The results from both models can be found in table 1, where we can see that the MLR slightly outperforms the PCR with four components across all metrics. The MLR model has a lower RMSE (2.44 vs 2.81) and MAE (1.91 vs 2.5) as well as a higher R^2 (0.93 vs 0.91), which indicates it has more accurate and better fitting predictions. This result is somewhat unexpected, as we expected PCR to perform better by reducing multicollinearity, improving generalization by filtering out noise with dimensionality reduction and by controlling for overfitting as we use less predictors by using fewer components. The MLR performing better suggests that the high predictive power of the original variables outweighed the benefits of dimensionality reduction.

Table 1: Model Performance: PCR vs. Linear Regression

Model	RMSE	MAE	R2
PCR (3 comps)	2.705709	2.128907	0.9143771
PCR (4 comps)	2.808528	2.163033	0.9077460
PCR (5 comps)	2.759551	2.153540	0.9109355
Linear Regression	2.439182	1.912216	0.9304149

To assess the sensitivity of the PCR with 4 components, we also trained two additional models with one fewer and one more component. The results can also be found in table 1, and show that three PCR models perform similarly with the model with three components performing the best, as it has the lowest RMSE and MAE, and highest R^2. The model with five components performs slightly worse, likely due to the addition of a component that captures noise. This suggests a potential shortcoming in choosing four components instead of three based on the scree-plot and Kaiser criterion, as these criteria may not align perfectly with predictive performance. Nonetheless, we still see that the MLR outperforms all the PCR models. These results show that PCR performance is sensitive to the number of chosen components and that one should be careful when balancing explained variance and prediction accuracy. Finally, to critically assess the prediction results, the test data was divided into ten deciles based on the maximum temperature from the coldest to the hottest days. This was then plotted against the MSE of the different models, where a lower MSE suggests a better prediction. The results as shown in figure 6 show that all models perform best in the deciles in the middle, except for the 6th decile. However, the models all have more difficulty predicting rarer, extreme weather events. For cold temperatures, PCR with three components outperforms the other models. After this decile, the PCR models perform similarly, and the MLR performs the best. This is also the case in the last decile at the hottest temperatures, which is the most important for our research question. This reinforces that the linear regression model is superior in predictive accuracy, particularly where it matters most.

Conclusion and Discussion

The analysis shows that today's weather data are a strong predictor of tomorrow's maximum temperature in Madrid, with a "warm-and-sunny" component (driven by temperature and radiation) dominating the signal.The first four principal components explain 86.8% of the variance, indicating that most of the original information can be compressed efficiently. Yet, because today's maximum temperature alone accounts for the bulk of tomorrow's value, PCA yields only modest gains in forecast accuracy while successfully addressing multicollinearity. The PCR model accuracy is highest for mid-range temperatures but falls on extreme hot or cold days, reflecting the rarity and nonlinear nature of those events that linear models miss. The sixth decile also shows a slight error increase, likely due to greater meteorological variability intransitional temperature ranges, small deviations in this densely populated segment disproportionately raise overall error. Since linear regression already outperforms PCR, penalised methods such as ridge or elastic-net should further enhance stability and predictive power by shrinking correlated coefficients without discarding information. Thus, PCA remains valuable for diagnostics, but penalised regression offers a better trade-off between simplicity, robustness, and forecast accuracy.

Appendix

Figure 1: Today mean temp vs Tomorrow's Max Temp

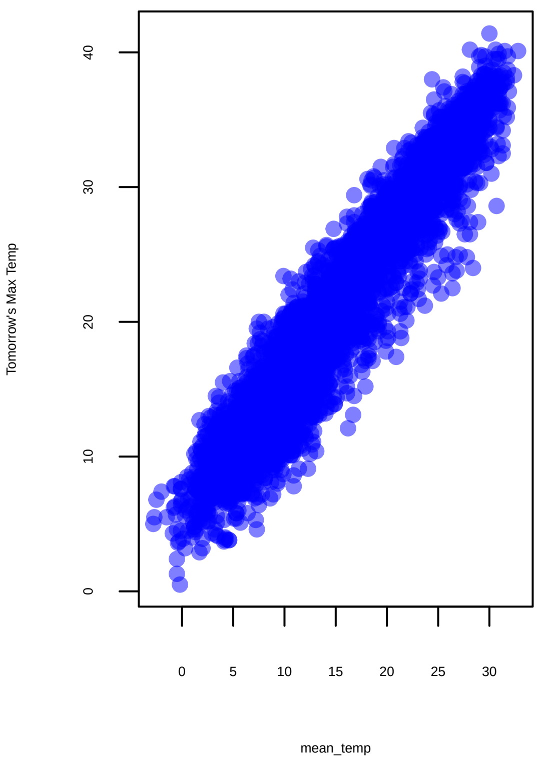
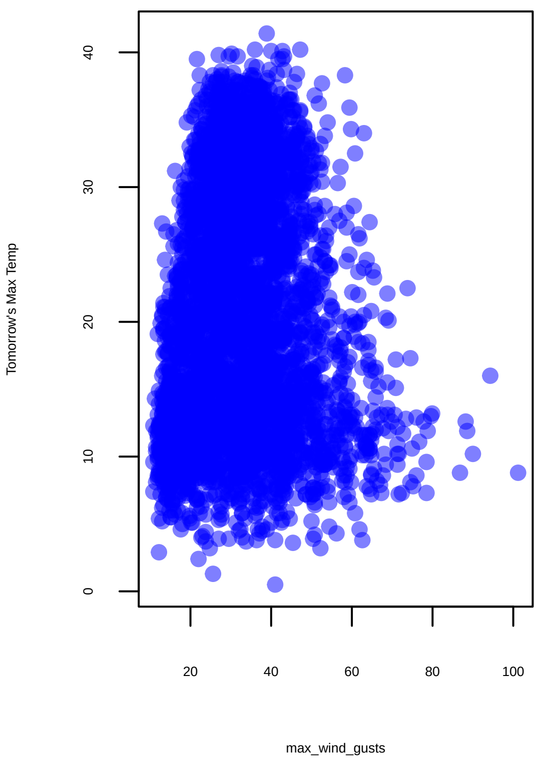


Figure 2: Maximum Wind gusts vs Tomorrow's max temp



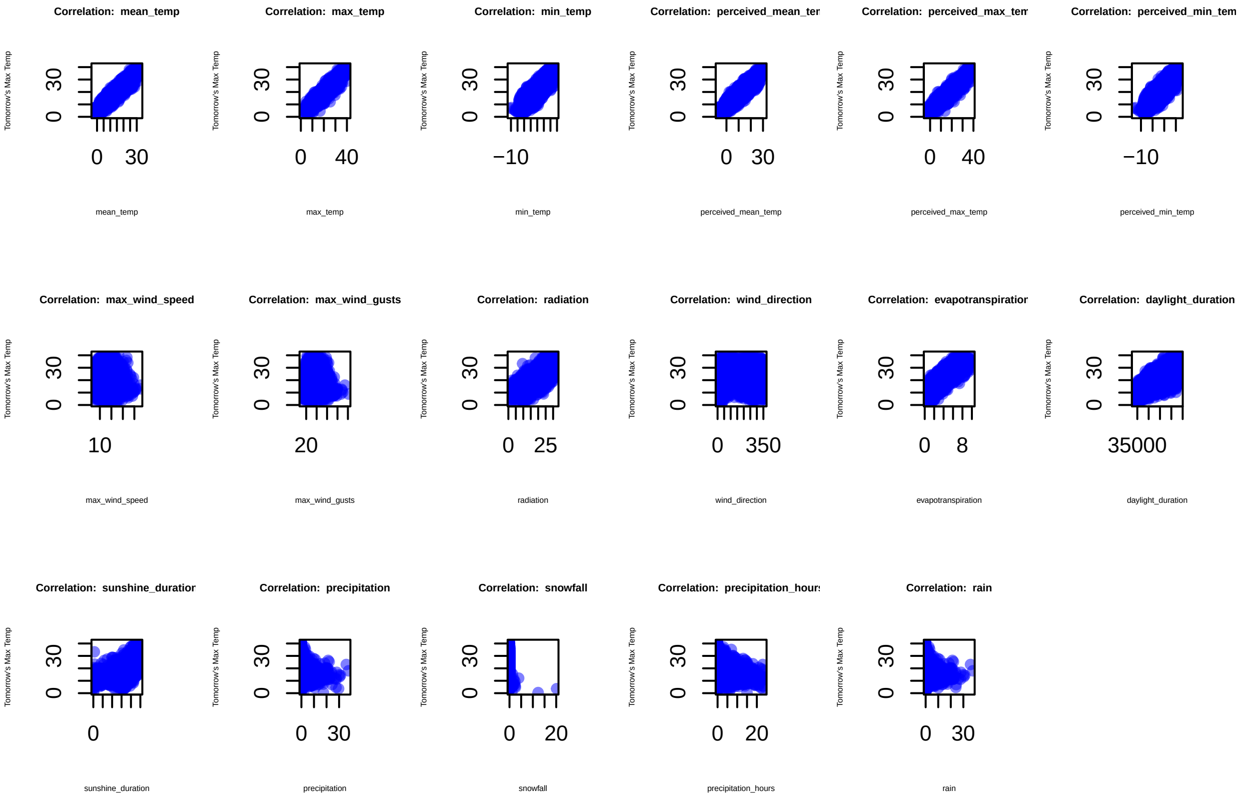


Figure 7: Bootstrap CI − PC1 ratio

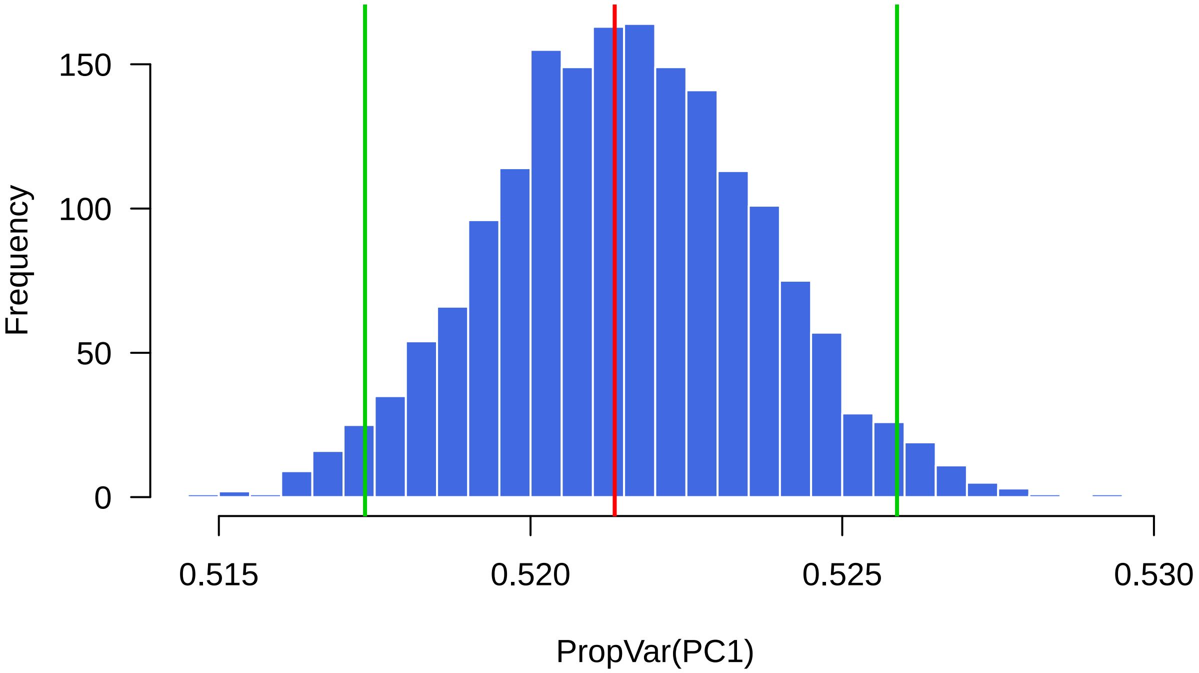


Figure 6

MSE Across Deciles for PCR and LinearRegression

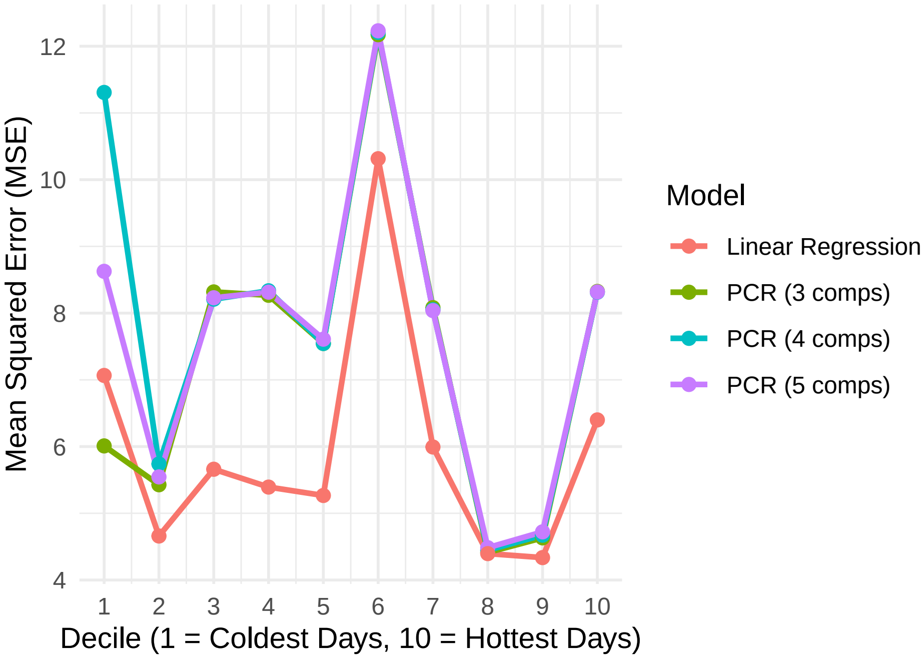


Table 2: Correlation Table for Numeric Varibales

Variable	Correlation	Interpretation
mean_temp	0.9551941	Very strong positive
max_temp	0.9636332	Very strong positive

Variable	Correlation	Interpretation
min_temp	0.8963496	Strong positive
perceived_mean_temp	0.9500798	Very strong positive
perceived_max_temp	0.9609327	Very strong positive
perceived_min_temp	0.8886969	Strong positive
max_wind_speed	-0.0007934	Negligible
max_wind_gusts	0.0277724	Negligible
radiation	0.8273042	Strong positive
wind_direction	-0.0968646	Negligible
evapotranspiration	0.9006286	Very strong positive
daylight_duration	0.7951203	Strong positive
sunshine_duration	0.6915392	Moderate positive
precipitation	-0.2222439	Negligible
snowfall	-0.0714789	Negligible
precipitation_hours	-0.3169115	Negligible
rain	-0.2149513	Negligible

Table 3: Variable Expained by PC1

	Variable	Loading_PC1	Fit_by_PC1
2	max_temp	0.3280981	0.1076484
1	mean_temp	0.3247060	0.1054340
5	perceived_max_temp	0.3241310	0.1050609
11	evapotranspiration	0.3227489	0.1041668
4	perceived_mean_temp	0.3207093	0.1028545
3	min_temp	0.3049892	0.0930184
9	radiation	0.3031546	0.0919027
6	perceived_min_temp	0.3012543	0.0907542
12	daylight_duration	0.2864544	0.0820561
13	sunshine_duration	0.2691713	0.0724532
16	precipitation_hours	-0.1329562	0.0176773
14	precipitation	-0.1075785	0.0115731
17	rain	-0.1060964	0.0112565
15	snowfall	-0.0482012	0.0023234
10	wind_direction	-0.0298747	0.0008925
8	max_wind_gusts	0.0275116	0.0007569
7	max_wind_speed	0.0130809	0.0001711

Linear Regression Coefficient Table

Table 4: Linear Regression Coefficient Table

Term	Estimate	Std_Error	t_value	Pr_t
(Intercept)	-1.4196	0.7053	-2.013	0.0442
mean_temp	1.4697	0.1751	8.395	0.0000
max_temp	0.2210	0.0824	2.683	0.0073
min_temp	-0.1095	0.0990	-1.106	0.2688
perceived_mean_temp	-0.5965	0.1607	-3.711	0.0002
perceived_max_temp	0.1178	0.0729	1.616	0.1061
perceived_min_temp	-0.0534	0.0878	-0.608	0.5433
max_wind_speed	-0.0220	0.0174	-1.264	0.2065
max_wind_gusts	-0.0563	0.0088	-6.365	0.0000
radiation	0.1615	0.0315	5.126	0.0000
wind_direction	-0.0012	0.0004	-3.289	0.0010
evapotranspiration	-0.7762	0.1238	-6.269	0.0000
daylight_duration	0.0001	0.0000	8.124	0.0000
sunshine_duration	0.0000	0.0000	-4.329	0.0000
precipitation	3.6727	10.7953	0.340	0.7337
snowfall	-4.7309	15.3975	-0.307	0.7587
precipitation_hours	-0.0615	0.0204	-3.017	0.0026
rain	-3.6596	10.7966	-0.339	0.7347

Code

```r
# --- DATA PREPARATION -------------------------------------------------------
X <- df[1:(nrow(df)-1), ]                # No "tomorrow" for last observation
y <- df$max_temp[2:nrow(df)]             # Day 1 has no "yesterday"

numeric_X <- X[sapply(X, is.numeric) & names(X) != "Location.ID"]  # Keep only numeric columns
# --- CORRELATION VISUALIZATION ----------------------------------------------
correlations <- sapply(numeric_X, function(col) cor(col, y))
par(mfrow = c(3, 6))
for (colname in names(numeric_X)) {plot(numeric_X[[colname]], y, xlab = colname,
    ylab = "Tomorrow's Max Temp",cex.lab = 0.4, main = paste("Correlation:", colname),
       cex.main = 0.5, pch = 19, col = rgb(0, 0, 1, 0.5))}
# --- TRAIN-TEST SPLIT -------------------------------------------------------
n <- length(y); train_size <- floor(0.8 * n)
X_train <- numeric_X[1:train_size, ]
y_train <- y[1:train_size]
X_test  <- numeric_X[(train_size + 1):n, ]
y_test  <- y[(train_size + 1):n]
# --- PCA ON TRAINING DATA ---------------------------------------------------
res <- princomp(X_train, cor = TRUE, scores = TRUE)
round(res$loadings, 2)
# --- METHOD 1: Cumulative variance explained
pca_var <- data.frame(PC = 1:length(res$sdev^2 / sum(res$sdev^2)),
  Variance_Explained = (res$sdev^2 / sum(res$sdev^2)),Cumulative_Variance = cumsum(res$sdev^2))
plot(pca_var$PC, pca_var$Cumulative_Variance, type = "b", pch = 19, col = "blue",
     xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Explained",
     main = "Cumulative Variance Explained by Principal Components")
abline(h = 0.8, col = "red", lty = 2)
# --- METHOD 2: Scree plot
plot(res$sdev^2, type = "b", col = "blue", ylab = "Variance", xlab = "Principal Component")
abline(h = 1, col = "red", lty = 2)
# --- Biplot
rownames(res$scores) <- rep(".", res$n.obs)
biplot(res, pc.biplot = TRUE, scale = 1, las = 1, col = c(rgb(0, 0, 0.5, 0.25),
       rgb(0.5, 0, 0)), cex = c(0.7, 0.9), main = "Biplot (PC1-PC2)")
# --- VARIANCE EXPLAINED & BOOTSTRAP CI FOR PC1 -----------------------------
ev <- res$sdev^2
prop_pc1_hat <- ev[1] / sum(ev)

X_use <- as.data.frame(X_train)
X_use <- X_use[, colSums(!is.na(X_use)) == nrow(X_use), drop = FALSE]
const_cols <- sapply(X_use, function(z) sd(z) == 0)
if (any(const_cols)) X_use <- X_use[, !const_cols, drop = FALSE]
X_mat <- as.matrix(X_use)

boot_pc1_prop <- function(data, idx) {x <- data[idx, , drop = FALSE]
  ev_b <- try(princomp(x, cor = TRUE)$sdev^2, silent = TRUE)
  if (inherits(ev_b, "try-error")) return(NA_real_)
  ev_b[1] / sum(ev_b)}

set.seed(123)
B <- 2000
fit.boot <- boot(data = X_mat, statistic = boot_pc1_prop, R = B)
```

```r
prop_vec <- fit.boot$t[, 1]
prop_vec <- prop_vec[is.finite(prop_vec)]
ci_pc1 <- quantile(prop_vec, probs = c(0.025, 0.975))

op <- par(mar = c(5, 4, 4, 1) + 0.1)
hist(prop_vec, breaks = 30, col = "royalblue", border = "white",
     main = "Bootstrap CI - PC1 ratio", xlab = "PropVar(PC1)", las = 1)
abline(v = ci_pc1, col = "green3", lwd = 2)
abline(v = prop_pc1_hat, col = "red", lwd = 2)
par(op)
# --- VARIABLE IMPORTANCE FROM PC1 -----------------------------------------
load_pc1 <- res$loadings[, 1]
fit_pc1  <- as.numeric(load_pc1)^2
best_table <- data.frame(Variable = names(load_pc1),Loading_PC1 = as.numeric(load_pc1),
  Fit_by_PC1 = fit_pc1)
best_table <- best_table[order(-best_table$Fit_by_PC1), ]
head(best_table, 10)
# --- MODEL TRAINING --------------------------------------------------------
train_df <- data.frame(y = as.numeric(y_train), X_train)
test_df  <- data.frame(y = as.numeric(y_test),  X_test)

lm_model <- lm(y ~ ., data = train_df)# Linear regression
lm_pred_test <- predict(lm_model, newdata = test_df)

pcr_3 <- pcr(y ~ ., data = train_df, scale = TRUE, ncomp = 3)# PCR with 3 components
pcr_4 <- pcr(y ~ ., data = train_df, scale = TRUE, ncomp = 4)# PCR with 4 components
pcr_5 <- pcr(y ~ ., data = train_df, scale = TRUE, ncomp = 5)# PCR with 5 components
pcr_pred_3 <- as.numeric(predict(pcr_3, newdata = test_df, ncomp = 3))# pred with 3 components
pcr_pred_4 <- as.numeric(predict(pcr_4, newdata = test_df, ncomp = 4))# pred with 4 components
pcr_pred_5 <- as.numeric(predict(pcr_5, newdata = test_df, ncomp = 5))# pred with 5 components
# --- MODEL PERFORMANCE -----------------------------------------------------
rmse <- function(actual, pred) sqrt(mean((actual - pred)^2))
mae  <- function(actual, pred) mean(abs(actual - pred))
r2   <- function(actual, pred) 1 - sum((actual - pred)^2) / sum((actual - mean(actual))^2)

(pcr_sensitivity <- data.frame(Model = c("PCR (3 comps)", "PCR (4 comps)", "PCR (5 comps)"),
  RMSE = c(rmse(y_test, pcr_pred_3), rmse(y_test, pcr_pred_4), rmse(y_test, pcr_pred_5)),
  MAE  = c(mae(y_test, pcr_pred_3),  mae(y_test, pcr_pred_4),  mae(y_test, pcr_pred_5)),
  R2   = c(r2(y_test, pcr_pred_3),   r2(y_test, pcr_pred_4),   r2(y_test, pcr_pred_5))))

lm_metrics <- data.frame(Model = "Linear Regression", RMSE = rmse(y_test, lm_pred_test),
  MAE  = mae(y_test, lm_pred_test),R2   = r2(y_test, lm_pred_test))
(model_comparison <- rbind(pcr_sensitivity, lm_metrics))
# --- DECILE-LEVEL MSE PLOT -------------------------------------------------
set.seed(123)
test_df_mse <- data.frame(max_temp = y_test,pred_pcr3 = pcr_pred_3,
  pred_pcr4 = pcr_pred_4,pred_pcr5 = pcr_pred_5,pred_lm = lm_pred_test) %>%
  mutate(max_temp_jittered = max_temp + rnorm(nrow(.), mean = 0, sd = 0.01))

decile_breaks <- quantile(test_df_mse$max_temp_jittered, probs=seq(0, 1, 0.1),na.rm=TRUE)
test_df_mse <- test_df_mse %>%
  mutate(decile = cut(max_temp_jittered, breaks = decile_breaks, labels = 1:10,include.lowest=TRUE))
```

```r
mse_df <- test_df_mse %>%group_by(decile) %>%summarise(mse_pcr3 = mean((max_temp - pred_pcr3)^2),
  mse_pcr4 = mean((max_temp - pred_pcr4)^2),mse_pcr5 = mean((max_temp - pred_pcr5)^2),
  mse_lm   = mean((max_temp - pred_lm)^2)) %>% ungroup() %>%pivot_longer(cols = starts_with("mse_"),
  names_to = "model", values_to = "mse") %>% mutate(model = recode(model,"mse_pcr3" = "PCR (3 comps)",
  "mse_pcr4" = "PCR (4 comps)", "mse_pcr5" = "PCR (5 comps)","mse_lm"   = "Linear Regression"))

ggplot(mse_df, aes(x = as.numeric(decile), y = mse, color = model)) +
  geom_line(linewidth = 1) + geom_point(size = 2) +
  labs(title = "MSE Across Deciles for PCR (3, 4, 5 comps) and Linear Regression",
       x = "Decile (1 = Coldest Days, 10 = Hottest Days)",y = "Mean Squared Error (MSE)",
       color = "Model") + scale_x_continuous(breaks = 1:10) + theme_minimal()
```