Regression Modeling of Oral Temperature Using Non-Contact Infrared Thermography

EE 559 Course Project

Data Set: Infrared Thermography

Nissanth Neelakandan Abirami (nissanth@usc.edu)

Sam Devavaram Jebaraj (devavara@usc.edu)

26 April 2024

1. Abstract

This study investigates the Infrared Thermography Temperature dataset, to forecast oral temperature (aveOralM) using a variety of factors such as infrared measurements from face sites and ambient circumstances. The dataset contains 1020 data points and 33 variables, which include gender, age, ethnicity, ambient temperature, humidity, and distance. The objective was to find effective machine learning methods for forecasting oral temperature based on these inputs.

The primary emphasis of the study was feature selection, which involves determining the most significant qualities using methods such as Pearson correlation and sequential forward feature selection. We utilized these chosen properties to develop other machine learning models, such as K-nearest neighbors (1NN), linear regression (without regularization), and support vector regression (SVR). The process subsequently shifted to a non-linear method, employing Polynomial Regression and Random Forest. The models' accuracy and reliability were evaluated using metrics such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE).

The findings revealed that various factors boosted the accuracy of mouth temperature forecasts, establishing infrared thermography as an excellent approach for forecasting oral temperature, with applications in medical diagnostics and fever screening in public and hospital settings.

2. Introduction

2.1. Problem Statement and Goals

Fever screening is critical for detecting people who have high body temperatures, which might indicate sickness. Traditional approaches frequently include contact-based procedures, which may be hazardous in some medical settings[2].

The fundamental purpose of this research is to use infrared thermography, a non-invasive, non-contact technology, as the foundation for fever screening using machine learning. This is accomplished by using linear and non-linear regression models that estimate average mouth temperature using face temperatures and ambient factors from the infrared thermography dataset.

Furthermore, the best model would be tested on previously unknown data and assessed using performance criteria. This might be particularly useful in hospital settings, where contactless temperature monitoring is favored for safety and efficiency.

2.2. Literature Review

[2] Infrared Thermography for Measuring Elevated Body Temperature: Clinical Accuracy, Calibration, and Evaluation

Wang et al. investigated various calibration methods, such as constant offset, ordinary linear regression, Deming regression, weighted linear regression, binning, and segmented linear regression, on 1000 subjects, collecting temperature data from various facial locations and reference oral temperatures.

The study assessed clinical accuracy by considering clinical bias (Δcb), standard deviation of Δcb ($\sigma\Delta$cb), repeatability, root-mean-square difference, and sensitivity/specificity.

Sublingual mouth temperatures were thought to be more accurate as a reference location because of their association with core body temperature. Clinical accuracy was best at inner canthi or full-face maximum temperatures, with a Δcb of ±0.03°C, $\sigma\Delta$cb less than 0.3°C, and sensitivity/specificity between 84% and 94%.

[3] Regression model for predicting core body temperature in infrared thermal mass screening

Limpabandhu et al.'s article focuses on using fever screening to avoid the spread of COVID-19. The study involves matching together subsequent characteristics and testing several regression models, including linear regression, regression trees, support vector machines, tree ensembles, and Gaussian Process regression.

To improve the model's capacity to effectively capture the underlying patterns and connections in the data, the four features with the greatest correlation values were selected, as well as three matched features with the lowest correlation scores.

Linear regression was the most successful, with a root mean square error (RMSE) of 0.285°C and a mean absolute error (MAE) of 0.240°C, showing that it is suitable for forecasting core body temperature from face temps. Other models, like regression trees and SVMs, exhibited lower accuracy due to overfitting or sensitivity to noisy data.

3. Approach and Implementation

3.1. Dataset Usage

The Infrared Thermography Temperature dataset mirrors the UCI version [1] but the columns "aveOralF", "Cosmetics", "time", and "date" are dropped from the original dataset.

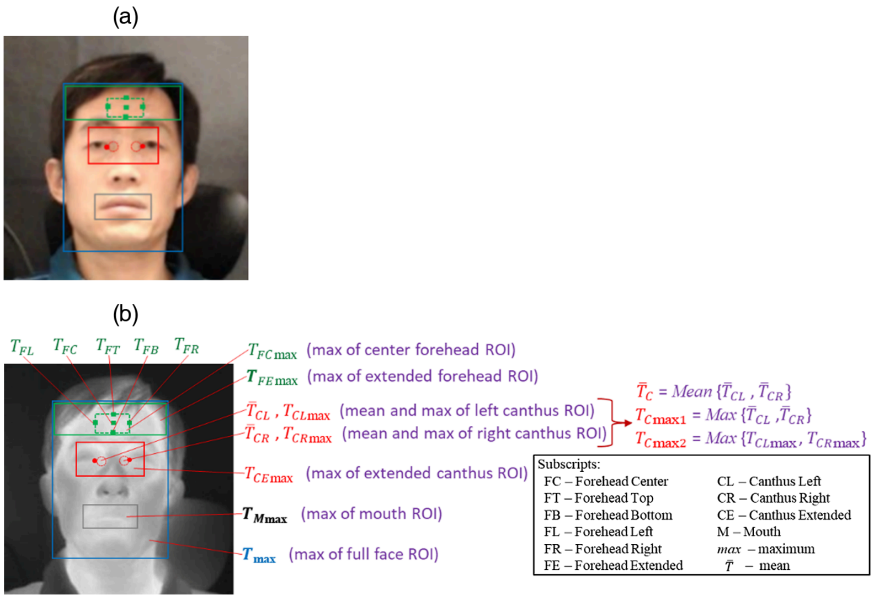

Figure 1: Delineated facial regions on (a) visible-light and (b) thermal images [4]

This study focuses solely on (Forward Looking Infrared) FLIR data from both groups 1 and 2. As a result, the total number of 33 characteristics comprises 26 temperature readings from various locations of the subject's forehead and canthus, as seen in Fig 1. When measurements were gathered across four rounds, the dataset included demographic information as well as environmental parameters.

Furthermore, the dataset was split into distinct training and testing datasets at a 2.3:1 ratio. To achieve an unbiased performance assessment, only the test dataset was used for the final evaluation. The dataset was used in the parts listed below:

Data preprocessing: The training set had 1261 null values. KNN Imputer was used to replace the null values. Statistical analysis was used to enhance the degrees of freedom in order to extract critical discoveries from existing key features.

Feature Selection: Using Pearson correlation, we identified the most important features, evaluated individual features with linear regression, and utilized sequential forward feature selection to simplify the model's input.

Model Development: Used a range of machine learning models, both linear and nonlinear, to predict oral temperature. The training dataset was subsequently divided into cross-validation sets of size 142 (710 samples / 5 folds) using K-fold cross-validation. Cross-validation is performed via a sequential loop that iterates across each fold. A total of eight models were used and tested.

Testing Models: An unseen dataset of 310 samples was also preprocessed and feature-engineered; columns of chosen features from the training set's dimensionality reduction were utilized to reproduce them in the testing dataset.

The test dataset was run through eight regression models and assessed using measures such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). R-squared (R^2) was calculated to understand how independent factors influenced the variability of the dependent variable.

3.2. Preprocessing

Following the elimination of empty columns, the features are appropriately aligned with their respective column headings. The feature dimensionality

increases to 115, with each round's feature represented as <feature>_<round number>.

Preprocessing required many processes to ensure data uniformity and cleanliness. The following approaches were used:

Data Imputation: The KNNImputer technique uses proximity to resolve missing values in a dataset's feature space. When n_neighbors=5, it finds the five nearest neighbors of a record with missing data and calculates the average. [5]

One-hot Encoding: The DataFrame's category columns are encoded using the OneHotEncoder to produce binary vectors [6]. Even age, with year ranges like "21-30" rather than numerical figures, is encoded.

Standardization: The sklearn.preprocessing module's StandardScaler is used to normalize features by removing the mean and scaling to unit variance, as seen below:

$$z = (x - u) / s$$

where:

x represents the original feature value,

u denotes the mean of the training samples,

s signifies the standard deviation of the training samples,

z stands for the standardized feature value.

This ensures that all features have the same scale, preventing features with larger scales from dominating the learning process during model training [7].

3.3. Data Visualization

Figure 2(a) shows a box plot created using the Matplotlib tool that depicts the distribution of oral temperatures among males and females. The box indicates the interquartile range (IQR), which shows where the middle half of the data falls. Outliers, shown as solitary dots outside the whiskers, are data points that deviate considerably from the norm. Interestingly, the IQR ranges for both genders are identical.

Figure 2(b) shows a histogram of the number of men and females. The dataset shows a larger proportion of females.

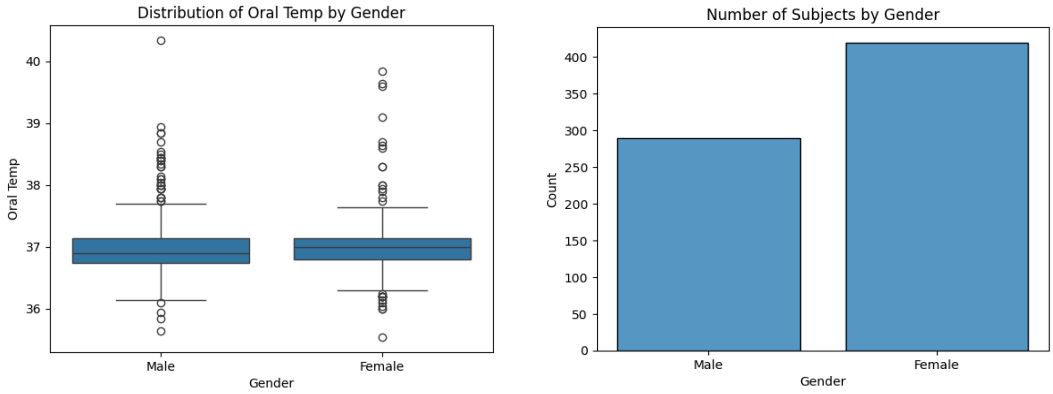

Figure 2(a) (left) - Box plot of AvgOralTemp by gender. Figure 2(b) (right) - Histogram of genders

Similarly, the plots for Ethnicity provide insight into the link between temperature distributions and ethnic makeup, as illustrated in Figure 3. The whisker lengths of box plots vary among ethnic groups, with white ethnicity having the highest prevalence.

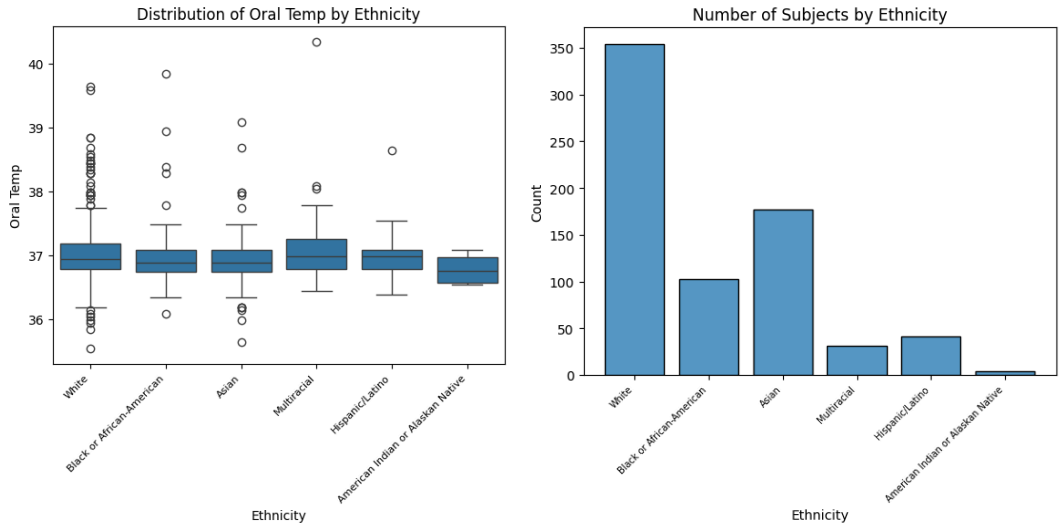

Figure 3(a) (left) - Box plot of AvgOralTemp by Ethnicity. Figure 3(b) (right) - Histogram of Ethnicity

3.4. Feature engineering and dimensionality adjustment

To get a more refined collection of features, a set of essential characteristics specified in Wang et al.'s paper[2] was employed to perform feature extraction using the statistical qualities of those corresponding to

temperatures aggregated by both gender and ethnicity. This study uses the mean, minimum, maximum, and median of relevant temperatures for each group to generate new columns in the training dataset. The characteristics used were T_FHC_Max_(Temperature Forehead Center Maximum), T_FHCC_(Temperature Forehead Center), T_LC_(Temperature Left Cheek), and canthiMax_(Maximum temperature measured at the canthi of the eyes) [1].

However, this increases the degrees of freedom from 116 to 244. Thus, feature reduction is required to escape the curse of dimensionality [8]. High-dimensional data increases computing demands, requiring longer training durations and greater memory capacities.

This is done by combining the three strategies listed below. Each approach generates a list of significant values, and the intersection of these three sets selects the most common relevant characteristics, which are then pushed into model selection.

Pearson Correlation Coefficient: A basic yet useful tool for identifying characteristics with a strong linear link to the objective. This involved calculating the Pearson correlation coefficients using the formula,

$$r = \frac{n\sum(xy) - \sum(x)\sum(y)}{\sqrt{[n\sum(x^2) - \sum(x^2)][n\sum(y^2) - \sum(y^2)]}}$$

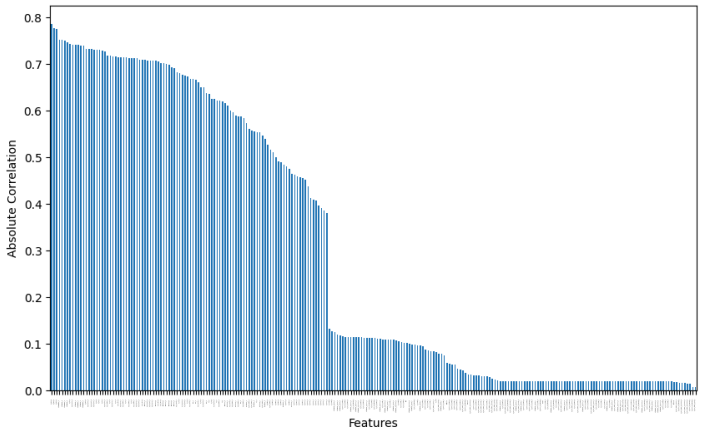

Figure 4: Bar Graph - Absolute Correlation with Target Column for Each Feature

The correlation coefficient of zero indicates no link. Furthermore, a correlation value of -1 or 1 indicates a significant negative or positive link [9]. From the preceding figure, only characteristics with absolute

correlation values larger than 0.2 were preserved, capturing only the top features with a significant linear association with the outcome.

Feature Performance Analysis: Linear regression was used to assess the performance of each feature independently using cross-validation. The negative mean squared error (MSE) for each feature was calculated using 5-fold cross-validation. Features with negative MSE larger than the criterion of -0.21 were selected.

Sequential Forward Selection: To identify the most valuable subset of features, a sequential forward feature selector was applied. This technique begins with an empty set and adds features one by one, based on their influence on negative MSE [10], until the target number of features, 100, is obtained. The junction of these three techniques yielded the final set of key traits. This lowered the degrees of freedom from 244 to 46 key characteristics, as seen in the plot below:

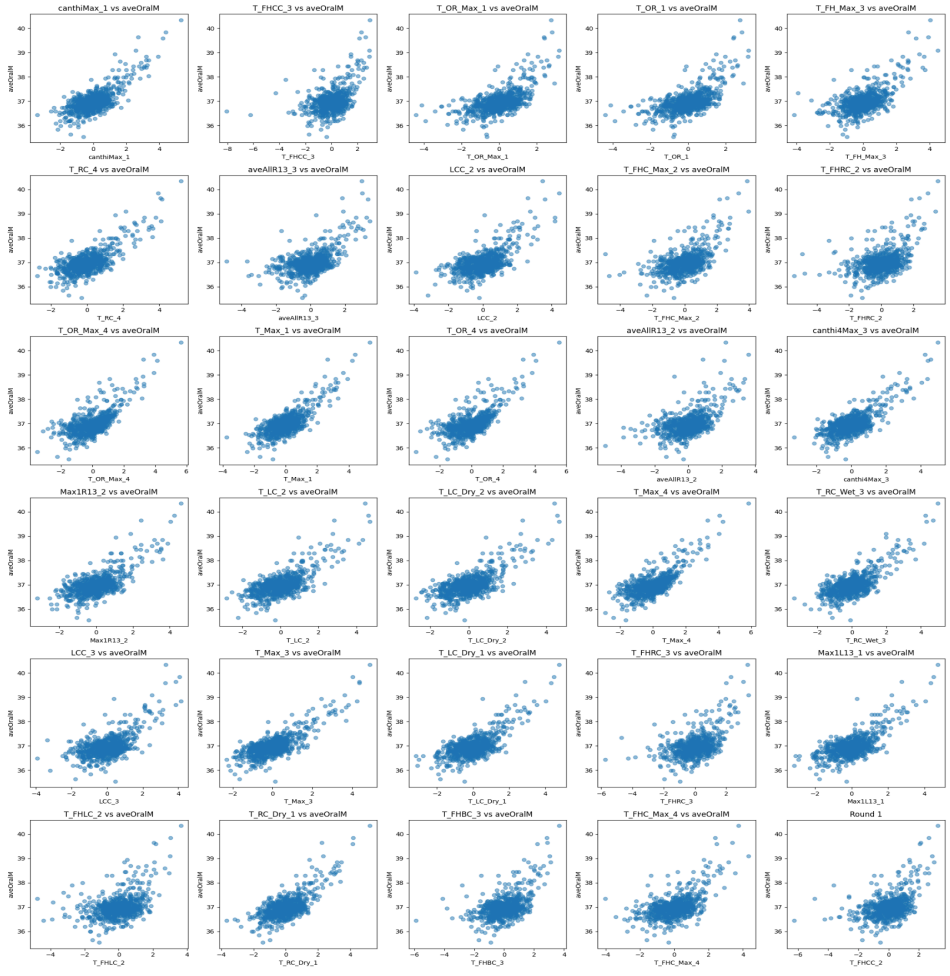

Figure 5: Scatter plots of 30 out of 46 best features vs target variable 'aveOralM'

3.5. Training, Regression, and Model Selection

The study used a variety of machine learning approaches, from simple baselines to more complicated models. To guarantee robust assessment, each model's training dataset goes through K-Fold cross-validation with 5 splits and shuffling. As a result, Nc (Number of constraints) remains constant in all circumstances at 710 - (710/5) = 568 samples. Here is a summary of the models utilized and how they were implemented.

Trivial System: The Trivial System is a simple reference model that consistently predicts the mean value of the target output from the training dataset.

- The Trivial System relies on a single parameter: the mean value of the training set's target variable. During the fit() procedure, the algorithm calculates this mean value, which is then predicted for all inputs during the predict() phase.
- Because the Trivial System has no free parameters other than the computed mean, it lacks degrees of freedom in the usual sense. It always returns the same result, regardless of limitations or data points. As a result, even when the testing dataset is used for prediction, it returns the mean value of the target column from the training dataset.

K-Nearest Neighbors (1NN): The K-Nearest Neighbors (1NN) system is a non-parametric regression approach that predicts a target value using the values of its nearest neighbors in the training set.

- The approach weights each neighbor and computes a weighted average to estimate the expected value [11]. This implementation makes predictions using the closest neighbor (K=1) algorithm.
- The D.O.F is about N/k [12], where N (568 for a validation set) is the number of data points and k is set to 1.

Linear Regression: Linear Regression is a traditional regression approach that develops a linear connection between input data and output variables.

- The linear regression model aims to identify the best-fit line across the data, reducing the gap between the observed and anticipated outcomes.
- Each feature adds a degree of freedom since the model develops a distinct coefficient for it. Hence, D.O.F = 46. [13]

Support Vector Regression (SVR): Support Vector Regressor (SVR) is a regression technique based on Support Vector Machines. It aims to discover the hyperplane that best matches the data while accounting for a margin of error known as epsilon.

- For a linear kernel, SVR computes the linear connection between features and targets in the same way as the prior model, but with a margin.
- If a linear kernel is used, the degrees of freedom are bound to the number of features, which is 46. The regularization parameter constrains the model by balancing data fitting and generalization.
- The RBF kernel is a popular choice for non-linear data, as it allows for more intricate interactions between features and targets. When implemented with default settings, a gamma value is used to adjust the curvature of the decision border.

Quadratic Polynomial Regression: Polynomial regression is a type of regression analysis in which the connection between the independent and dependent variables is expressed as an nth-degree polynomial.

- The degree of 2 is selected to generate more words. Even when the data is translated into polynomial terms, the regression procedure remains linear.
- Additional words increase the number of characteristics. Given the initial 46 characteristics, a degree-2 polynomial transformation increases the degrees of freedom to 1128. [C(D+n, D) = 1128]

Non - Linear Transformation through RBF and Linear Regression: Non-Linear Transformation using RBF and Linear Regression: Radial Basis Function transformation is a popular approach for transforming input features into a new representation, with each new feature representing the response from a radial basis function.

- Gamma determines the dispersion of the radial basis function. The optimal value of gamma is chosen through the equation below:

$$\gamma = \frac{1}{8\alpha^2} \qquad \alpha = \frac{\Delta}{M^{1/D}}$$

where α is the average spacing between basis function centers,
Δ= extent (width) of feature space in any one dimension
D = number of features (dimensionality of unaugmented feature space)
M = number of basis function centers.

- The number of clusters (K) obtained using K-means indicates the centroids utilized in the RBF transformation [14]. The best values achieved for K and gamma are 90 and 0.028125, respectively. Therefore, the D.O.F is 90.

Random Forest Regressor + Hyperparameter Tuning [beyond the scope of EE 559]: Random Forest Regressor is one of the ensemble learning methods. The core premise of ensemble learning is that a set of weak learners combine to become a strong learner, increasing the model's accuracy. [15]

In the case of this model, it generates a collection of decision trees from a randomly chosen portion of the training data. Each subset corresponds to a decision tree node, and the split criteria are specified by the dataset's characteristics. The Random Forest Regressor trains these decision trees and then combines their votes to determine the final class of the test item. [16]

RandomizedSearchCV is an effective and adaptable method for hyperparameter tweaking in machine learning models. In comparison to GridSearchCV [17], it efficiently explores the search space and delivers the optimum parameters based on a given performance indicator by randomly choosing a subset of hyperparameter combinations, making it particularly ideal for time-consuming model training.

Because the Random Forest Regressor requires a few parameters, such as the number of estimators, maximum depth, minimum samples, and so on, these values were tuned using RandomizedSearchCV tuning.

4. Results and Analysis: Comparison and Interpretation

This comparison includes a variety of models, ranging from trivial and baseline to more complex regression and machine learning approaches. The training occurs through K-Fold cross-validation with 5 splits and shuffling (Nc = 568). The results presented are based on the training dataset during model selection:

Model	Avg Train RMSE	Avg Train MAE	Avg Train MSE
Linear Regression	0.2416716332	0.1899542362	0.05840517829
Trivial Systems	0.4872352768	0.3158600104	0.237398215
1NN	0	0	0

SVR Linear	0.2459425524	0.188141448	0.06048773909
SVR Rbf	0.1859250291	0.1376534691	0.03456811646
Polynomial	0	0	0
RBF + Regression	0.2271624958	0.1762370036	0.05160279951
RandomForestRegressor	0.1385783987	0.1102306195	0.01920397258

Table 1: Training RMSE, MAE, and MSE during Model Selection

Observations:

The Tuned Random Forest Regressor outperforms all other measures on the training set, suggesting good handling of both linear and nonlinear patterns as well as resilience to overfitting.

Trivial Systems and SVR with Linear Kernel have larger errors, indicating that they are less successful at capturing the complexity of the data.

1NN and Polynomial Regression provide zero errors, which may imply overfitting to the training data unless the findings are from a controlled or simple instance. This may be due to the fact that the condition of Nc > (3-10) D.O.F is breached with D.O.F being equal to 568 and 1128 in 1NN and Polynomial regression respectively.

Model	Avg Val RMSE	Avg Val MAE	Avg Val MSE
Linear Regression	0.2663640468	0.2099209115	0.07094980545
Trivial Systems	0.4839243647	0.3160925412	0.2341827907
1NN	0.3553499824	0.2683802817	0.12627361
SVR Linear	0.2682838105	0.2084161535	0.07197620296
SVR Rbf	0.2861336941	0.2095437152	0.08187249088
Polynomial	0.915624246	0.5835490584	0.8383677598
RBF + Regression	0.2455577622	0.1897905355	0.06029861456
RandomForestRegresso	0.2564860051	0.195082741	0.0657850708

Table 2: Validation of RMSE, MAE, and MSE during Model Selection

Observations:

RBF + Regression has the lowest validation errors, indicating that it generalizes well to previously unknown data and may be the best-performing model based on validation findings. The Random Forest Regressor likewise similarly well.

Polynomial Regression and 1NN have much higher errors indicating overfitting as training errors are nil, but returns with significant loss on validation sets.

Trivial Systems because of its simplicity limits in capturing data complexity. Other linear models perform well in both training and validation sets.

310 testing samples were also preprocessed and columns of significant features from the training set's dimensionality reduction were utilized to reproduce them in the testing dataset. Table 3 shows the testing results with an additional metric R^2 to understand how independent variables change the variability of the dependent variable. This is defined by the formula:

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

Model	Test MSE	Test RMSE	Test MAE	Test R^2
Linear Regression	0.0810866	0.2847571	0.2183031	0.7319615
Trivial System	0.3118289	0.5584164	0.3530418	-0.0307766
1NN	0.1301532	0.3607676	0.2788710	0.5697676
SVR - Linear	0.0829811	0.2880644	0.2189440	0.7256991
Polynomial	1.1545235	1.0744876	0.6125641	-2.8163740
RBF + Regression	0.0693928	0.2634252	0.1980927	0.7706162
Random Forest	0.0592889	0.2434932	0.1851068	0.8040155
SVR - Rbf	0.0709616	0.2663862	0.2000699	0.7654305

Table 3: Test RMSE, MAE, MSE and R^2 during Inference

Observations:

The Random Forest Regressor has the lowest error as well as the greatest R^2, showing good generalization and performance on the test set.

RBF transformation with Linear Regression and SVR - RBF Kernel also perform well, with low errors, indicating the existence of non-linear patterns in the test data.

Trivial System and Polynomial Regression perform the worst, with high errors and negative R^2 values, indicating overfitting or failure to generalize to test data. 1NN being 100% accurate to the training dataset surprisingly performs better than the Trivial System. As observed in Fig. 6, the best model is the Random Forest Regressor, which is out of EE559's scope and the second best with a 0.02 difference in RMSE is RBF Kernel transformation followed by linear regression. Moreover, SVR with linear kernel performs slightly poorer than the normal linear regression, indicating the presence of margins of error degrades the generalization.

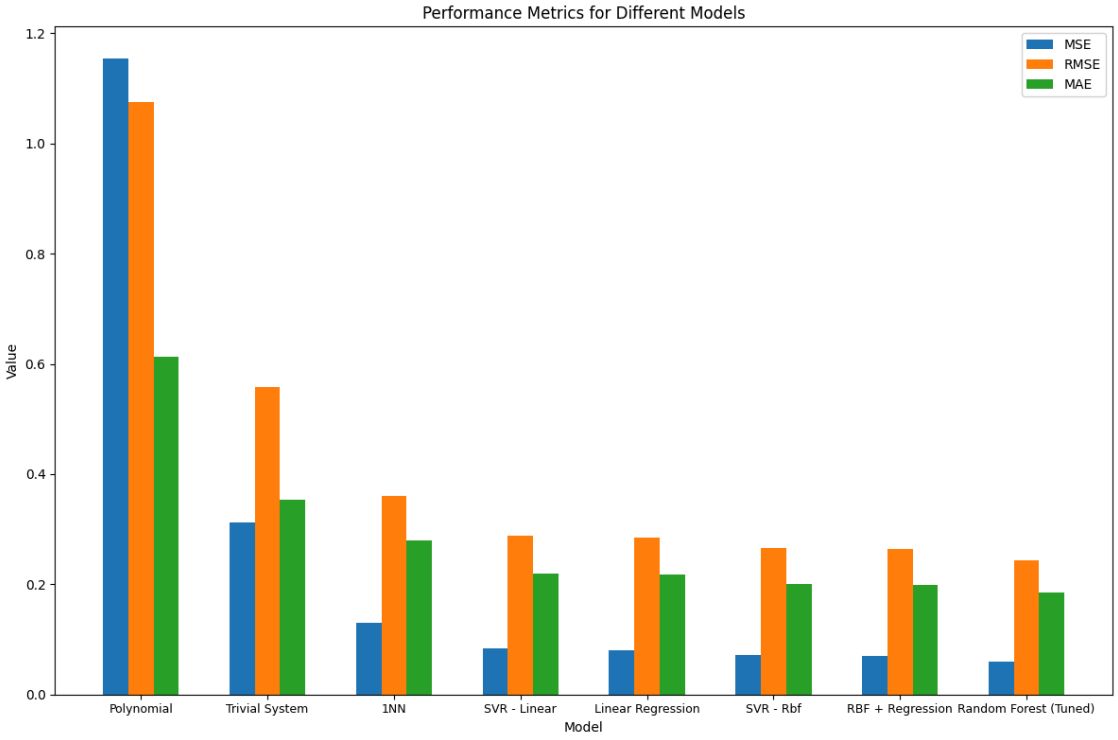

Figure 6: Errors from Testing Dataset in descending order (Worst to Best)

Best Performing System:

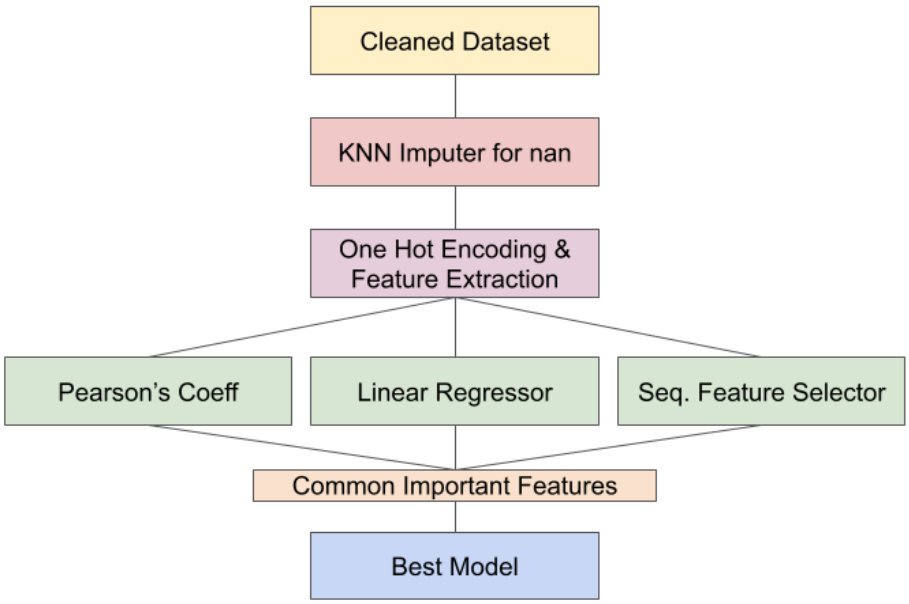

Figure 7: Best Performing System for Inference. The best models are Random Forest Regressor (outside EE559 Scope) and RBF Transformation with Linear Regression (EE559 Scope)

Model: Random Forest Regressor (outside EE559 Scope)

Parameters: Tuned parameters through RandomizedSearchCV:

n_estimators: Defines the number of decision trees to be used in the random forest. Chosen value = 581

max_depth: Specifies the maximum depth of each decision tree in the random forest. Chosen value = 9. More depth can lead to overfitting.

min_samples_split: The minimum number of samples required to split a node during the tree-building process. Chosen value = 3.

min_samples_leaf: Specifies the minimum number of samples required to be at a leaf node. Chosen value = 1.

max_features: Determines the maximum number of features to consider when looking for the best split at each node. Chosen = 'sqrt ' (the square root of the total number of features).

Test Performance:

Test MSE (Mean Squared Error): The MSE value of 0.059 is the lowest error rate, indicating that the Random Forest performs well in terms of prediction accuracy.

Test RMSE (Root Mean Squared Error): RMSE is just the square root of MSE, representing the standard deviation of prediction errors. An RMSE of 0.2435 confirms that the Random Forest model has good reliability in predictions.

Test MAE (Mean Absolute Error): MAE measures the average absolute difference between predicted and actual values. An MAE of 0.1851 suggests consistent performance across test cases.

Test R^2 (R-squared): An R^2 of 0.8040 suggests that the Random Forest model explains 80.4% of the variance in the test data, indicating strong predictive capability and good generalization.

Model: RBF + Regression (EE559 Scope)

Parameters : Best K =90 , Best Gamma = 0.028125

Test Performance:

Test MSE (Mean Squared Error): The Test MSE is 0.0694, slightly higher than Random Forest yet significantly lesser than SVR with RBF kernel.

Test RMSE (Root Mean Squared Error)**:** This model has a Test RMSE of 0.2634, indicating a relatively low error rate.

Test MAE (Mean Absolute Error): An MAE of 0.1981, suggests little deviation of predictions from the actual oral temperature.

Test R^2 (R-squared): A Test R^2 of 0.7706, shows strong explanatory power like that of Random Forest Regressor.

Key Attributes of the Best-Performing Models:

RBF + Regression: This model's flexibility allows for effective handling of complex data, reflected in its best performance. However, the RBF transformation during inference takes a longer time (1.882 seconds)

Random Forest Regressor: An ensemble approach with high degrees of freedom, Random Forest tends to be robust against overfitting. Its results indicate strong performance along with a faster inference time (0.245 seconds)

Both models surpassed the model provided base paper [3] in terms of RMSE as well as MAE by a margin of 0.2 and 0.4 respectively. The base paper had an average RMSE of 0.283.

5. Libraries used and what you coded yourself

The project employs various libraries for data preprocessing, feature engineering, model algorithms, cross-validation, and evaluation. Here's a summary of the libraries used for each component of the model pipeline:

Data Preprocessing and Feature Engineering

Pandas: Used for data manipulation, including creating and modifying DataFrames, renaming columns, one-hot encoding, reshaping, and general preprocessing tasks.

Numpy: Provides support for numerical operations, reshaping arrays, and basic statistics. Trivial system was built as a class with just the numpy.mean() function.

Data Visualization

Matplotlib and Seaborn: Used to create plots and visualize data.

Machine Learning Algorithms

Scikit-learn: The primary library for machine learning, offering a range of algorithms for classification and regression. It provides various models, including:

K-Nearest Neighbors (KNN) with k=1: A simple classification or regression algorithm that finds the closest data point. Scikit-learn provides support for KNN through the KNeighborsRegressor classes.

Linear Regression with No Regularizer: A basic linear regression model without regularization techniques like L1 or L2.

Polynomial Regression: Scikit-learn offers support for polynomial features with the PolynomialFeatures class. The study analyzed the quadratic expansion of features.

Non-Linear Transformation through RBF and Regression: This algorithm involves non-linear transformations like the Radial Basis Function (RBF). Scikit-learn provides LinearRegression, the RBF transformation however was done using numpy.

Random Forest Regressor: Scikit-learn provides support for Random Forest through the RandomForestRegressor and RandomForestClassifier classes with a handful of parameters to tune.

Cross-validation and Model Training

Scikit-learn: The KFold class is used to implement cross-validation. The library also provides metrics to evaluate model performance, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2).

These libraries form the backbone of the code in the notebook, supporting tasks from data preprocessing to model training and evaluation.

6. Contributions of each team member

Nissanth Neelakandan Abirami:

- Ensured data integrity by establishing proper column headers and removing empty columns.
- Handled missing data by imputing null values using KNN imputer
- Implemented one hot encoding for categorical data of Gender, Age, and Ethnicity.
- Produced insightful visualizations including box plots, histograms, and scatter plots in subplots, illustrating the relationship between features and target variables
- Employed a variety of techniques such as 1NN, SVR - Linear and RBF Kernel, and Polynomial regression.

Sam Devavaram Jebaraj:

- Ensured data consistency through standardization procedures.
- Conducted feature extraction and engineered new features to enhance model performance.
- Utilized Pearson correlation to perform dimensionality reduction, linear regression as well as Sequential Feature selector for identifying and retaining essential features.
- Employed models including Trivial, Linear regression, and RBF transformation.
- Leveraged RandomizedSearch for hyperparameter tuning in Random Forest regression.

7. Summary and conclusions

The study investigated the Infrared Thermography Temperature dataset with the primary goal of forecasting oral temperature (aveOralM) using a variety of parameters.

Key Results

	Random Forest Regressor	RBF + Regression
Train RMSE	0.1385783987	0.2271624958
Val RMSE	0.2564860051	0.2455577622
Test RMSE	0.2434932	0.2634252
Inference time (s)	0.245	1.882

The Random Forest Regressor was the best-performing model, with the lowest set of errors and the lowest inference time. Radial Basis Function transformation followed by Linear Regression achieved the second-best results, but with a cost of inference time, almost 8-9 times. Both models outperformed the baseline paper [3] by a considerable margin, achieving improvements ranging from 0.2 to 0.4 in terms of RMSE.

Trivial System and Polynomial Regression had the poorest performance, with high test RMSE and negative R^2 values, suggesting poor generalization and overfitting. If inference time was taken into consideration with allowance of trading off negligible error difference, the third best model, Support Vector Regression with Radial Basis Function kernel would be considered.

Future work can involve usage of deep neural networks to perform regression and better feature engineering and hyperparameter tuning using Bayesian Optimization.

References

[1] Infrared Thermography Temperature - UCI Machine Learning Repository

[2] Wang, Quanzeng, et al. "Facial and oral temperature data from a large set of human subject volunteers" (version 1.0.0). PhysioNet (2023), https://doi.org/10.13026/3bhc-9065

[3] Limpabandhu C, Hooper FSW, Li R, Tse Z. Regression model for predicting core body temperature in infrared thermal mass screening. IPEM Transl. 2022 Nov-Dec;3:100006. doi: 10.1016/j.ipemt.2022.100006. Epub 2022 Jul 15. PMID: 35854880; PMCID: PMC9284542.

[4] Yangling Zhou, Pejman Ghassemi, Michelle Chen, David McBride, Jon P. Casamento, T. Joshua Pfefer, Quanzeng Wang, "Clinical evaluation of fever-screening thermography: impact of consensus guidelines and facial measurement location," J. Biomed. Opt. 25(9) 097002 (12 September 2020) https://doi.org/10.1117/1.JBO.25.9.097002

[5] 6.4. Imputation of missing values — scikit-learn 1.4.2 documentation

[6] https://datagy.io/sklearn-one-hot-encode/

[7] sklearn.preprocessing.StandardScaler — scikit-learn 1.4.2 documentation

[8] The Curse of Dimensionality. Why High Dimensional Data Can Be So… | by Tony Yiu | Towards Data Science

[9] https://www.wallstreetmojo.com/pearson-correlation-coefficient/

[10] sklearn.feature_selection.SequentialFeatureSelector — scikit-learn 1.4.2 documentation

[11] Understanding K-Nearest Neighbors (KNN) Regression in Machine Learning | by NANDINI VERMA | Medium

[12] L4_knn (ubc.ca)

[13] df.pdf (cmu.edu)

[14] USC EE559 - HW 7 - Spring 2024

[15] Random Forest Regression. A basic explanation and use case in 7… | by Nima Beheshti | Towards Data Science

[16] sklearn.ensemble.RandomForestRegressor — scikit-learn 1.4.2 documentation

[17] Hyperparameter Tuning Showdown: Grid Search vs. Random Search — Which is the Ultimate Winner? | by Hestisholihah | Medium