AML Report

Challenge 1: Weather Prediction

EURECOM

Group 7: Emerson CARDOSO, Sameer HANS, Deepika SIVASANKARAN

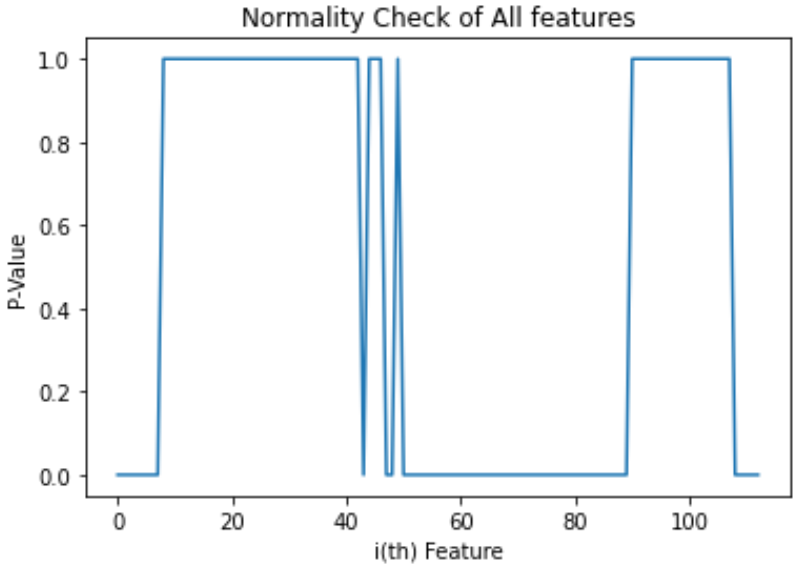
1. Introduction

The challenge aims to build models to predict the weather in distinct locations around the world. A CSV dataset (train dataset) containing columns denoting various features are given, using which we predict the output fact_temperature in the test dataset. The procedural way implemented to achieve the aim is analyzing the data for incompleteness, selecting of most correlated features and training various models to obtain the best performing model.

2. Data Analysis

The target variable fact temperature contains values in the range [-45.0, 60.0].

a) **Distribution**:



The data distribution for all the features are checked for normal distribution using the Shapiro Wilk Normality test. Features that are deviating from the normal bell have a p-value of 0, while the others have 1.0.

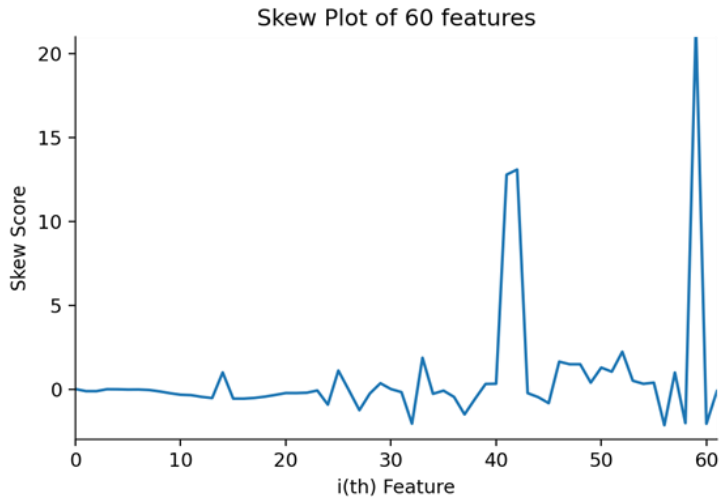
b) **Completeness:** The dataset containing missing values which are filled using the mean of each column.

c) **Numerical features:** The dataset contains only numerical features and no categorical features. The dataset is normalized since each column contains data points that fit in different ranges. Hence to speed up the model training process, the data is normalized using the respective column mean values and are unstandardized after predictions.

d) Futile Columns: wrf_available is discarded since it only provides information about other wrf columns. Similarly, gfs_soil_temperature_available only provides data about the

availability of soil temperature

e) **The skewness of data:** Skewness determines how biased a dataset will be due to it being imbalanced and deviating from the symmetry of the Normal bell curve. The overall dataset is very slightly skewed with a skew score of **-0.077248,** with individual columns going as high as 21.0. The skew scores are seen below



3. Data Preparation

Before feeding the dataset to our model we need to do some preprocessing in the dataset to address the problems that would arise due to incompleteness and find the best features that could maximize the model performance. The steps are as follows:

1. Impute all the columnar missing values from our dataset with the mean values of each column.
2. Find the features with a high correlation with the fact temperature
3. Using univariance to calculate the features with a strong relationship with the fact temperature.
4. Calculate the mutual information between the fact temperature with the rest of the features.
5. We combine the three techniques of feature selection (correlation matrix, embedded method using LassoCV and Univariate selection) in order to select the best features for us.
6. Because we select models that do make an assumption about the dataset, we choose to Standardize our dataset by subtracting the mean and dividing by the standard deviation.

4. Model Selection

Because our data is continuous and the target variable is also continuous the problem in our hand is regression. For this problem we tried four different regression algorithms with the following parameters.

a) **Lasso** – It is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a pivotal point. This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

b) **Decision Tree Regressor -** Decision Tree is used for classification as well as regression. It contains a Tree like structure with the branches representing various decisions on features to be made and the leaf nodes indicating a target variable.

c) **Random forest Regressor** – It is a supervised learning algorithm that uses ensemble learning methods for regression. We thought that it could be a good model to predict the temperature as it is one of the models that work well with continuous values and as an additional precaution it automates missing values present in the data. It is also expected to reduce overfitting in decision trees and helps to improve the accuracy.

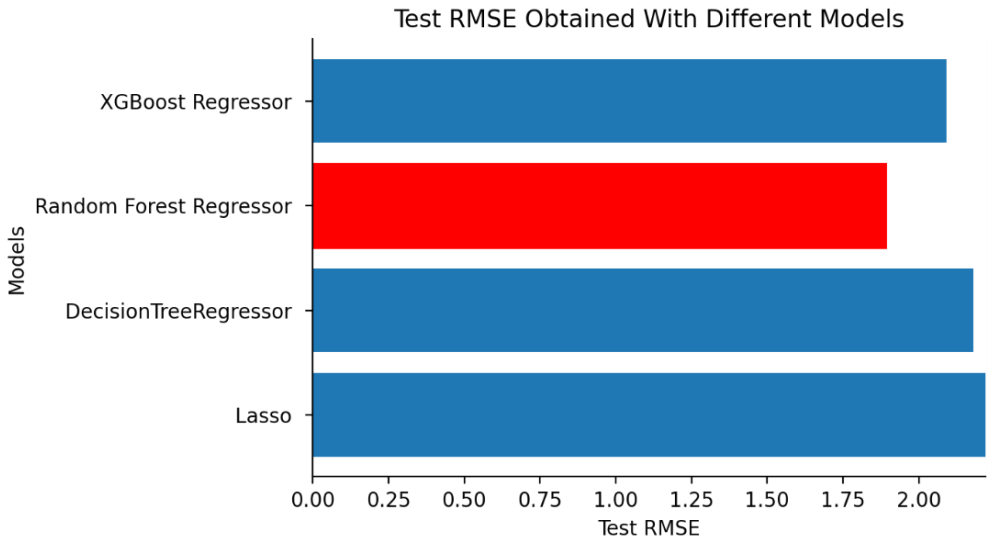
d) **XGBoost -** XGBoost or Extreme Gradient Boosting is an implementation of gradient boosted decision trees. Similar to random forest, trees are created in sequence and each tree is trained individually on the features. The features are provided with weights which are then updated when a tree makes an incorrect prediction. The new weights are fed into the successive tree and so on, until the mistakes made by the previous trees are negligible.

Note: Several other models were tested on the data but crashed due to RAM issues. The models tested, which crashed are: Support Vector Machine, KNN, BayesianRidge, SGDRegressor with MinMaxScaler. Imputing techniques that crashed: MICE, KNN

5. Results and Discussion

After performing our experiment on the training set, we notice that the Random Forest Regressor achieved a better performance compared to all other models.

Visualised as a plot, we see the test RMSE obtained for each model.



The table below shows in detail the performance of the models.

Model	Parameters	Train RMSE	Val RMSE	Test RMSE
Lasso	alpha=0.001	2.301	2.355	2.22
DecisionTree Regressor	min_samples_split=20, random_state=42, max_depth= 70, min_samples_leaf= 50	2.077	2.362	2.18
Random Forest Regressor	n_estimators = 50, random_state = 50, bootstrap= True, max_depth= 80, max_features= 'auto', min_samples_leaf= 4, min_samples_split=10,	1.097	2.011	1.89
XGBoost Regressor	n=100, max depth=5, subsample=0.8, colsample_bytree=0.8	2.275	2.308	2.09

The high performance of Random Forest is it being an ensemble model due to which it uses many decision trees individually that train on the dataset and using a majority voting system, it provi des with an output. It is also an efficient method since the chances of overfitting which are bound to occur with decision trees are avoided here.

6. Conclusion

Thus, with correct feature selection and less complex models chosen by us, we find that Random Forest Regressor with the mentioned parameters predicts the temperature better considering the RMSE and Cross validation score. Further, if given some more time, we would have tried to optimize the hyperparameters of the models, optimize the RAM utilisation and would have tried a few more stacked average models to improve the performance of the model.