KLE Technological University

Creating Value
Leveraging Knowledge

School of Computer Science and Engineering

DATA MINING ANALYSIS (18ECSC301)

Report on Course Project

WEATHER SHIFT PREDICTION

(5DMACP09)

TEAM NUMBER: 09

SUBMITTED BY

NAME	USN
BHUVAN M C	01FE19BCS275
PRAJWAL METI	01FE19BCS276
SANKALP PATTANASHETTI	01FE19BCS277
SHREEHARI ALAGAWADI	01FE19BCS279

Faculty In charge

Mr. Shankar Gangisetty

Table of content:

1. ABSTRACT

There has been significant research done on developing methods for improving robustness to distributional shift and uncertainty estimation. In contrast, only limited work has examined developing standard datasets and benchmarks for assessing these approaches. Additionally, most work on uncertainty estimation and robustness has developed new techniques based on small-scale regression or image classification tasks. However, many tasks of practical interest have different modalities, such as tabular data, audio, text, or sensor data, which offer significant challenges involving regression and discrete or continuous structured prediction. In this work, we propose the Shifts Dataset for evaluation of uncertainty estimates and robustness to distributional shift.

2. INTRODUCTION

The Challenge was on Robustness and Uncertainty Under Real World Distributional System. The goal will be to develop models which are robust to distributional shift and to detect such shift via measures of uncertainty in their predictions.

Good performance and generalization on test data imply that the model will perform well in deployment. Unfortunately, this assumption seldom holds in real, “in the wild”, applications of machine learning. In practice, data are subject to a wide range of possible distributional shifts - mismatches between the training data, and test or deployment data.

Training data is not uniformly distributed. Weather prediction models often encounter shifts in climate and degrading performance over time. This challenge would like the models to be more robust to these effects and also to yield uncertainty estimates which tells about accuracy. Duration is July 20, 2021 to October 31, 2021.

3. PROBLEM STATEMENT

Predict the temperature at a particular latitude/longitude and time along with uncertainty, given all available measurements and climate model predictions.

4. RELATED WORKS

We studied the research paper “Shifts: A Dataset of Real Distributional Shift Across Multiple Large-Scale Tasks”. The open-source CatBoost gradient boosting library that is known to achieve state-of-the-art results on various tabular datasets. Here we consider an ensemble-based approach to uncertainty estimation for GBDT models. An ensemble of ten models is trained on the train data from the canonical partition of the Weather Prediction dataset with different random seeds is used as the baseline. The models are optimized with the loss function RMSE With uncertainty that predicts mean and variance of the normal distribution by optimizing the negative log-likelihood. Each model is constructed with a depth of 8 and then is trained for 20,000 iterations at a learning rate of 0.3. Hyperparameter tuning is performed on the dev_in data.

Another paper we studied is Neil Band Research Paper: As the uncertainty measure, we use the total variance (tvar) that is the sum of the variance of the predicted mean and the mean of the predicted variance as our chosen measure of uncertainty. We also compare the ensemble with an individual model, where we use the predicted variance as an uncertainty measure.

5. METHODOLOGY

The problem is solved by carrying out the steps as shown in figure 1. Initial step started with the pre-processing of the data in which data transformation and data reduction is carried which is then followed by the step model detection in which the parameters selection is carried and obtained the best graph out of it. The last step includes the prediction part where the best selected model predicts the output. The predicted output is then plotted and compared with the other resultsUnderstanding the data .

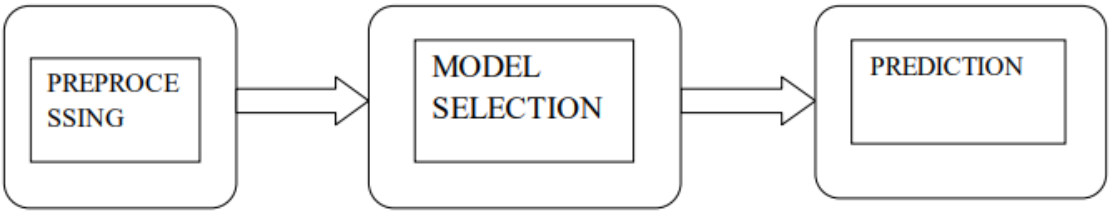

Figure 1 : System Model

6. REVIEWING THE DATASET

6.1 Dataset Description

Size of training data is (3129592, 129) . Size of development data is (50000, 129).

Size of evaluation data is (1048575, 123).

Table 1: Number of samples in the canonical partitioning of Weather Prediction dataset.

	Data	# of samples					
		Total	Tropical	Dry	Mild Temperate	Snow	Polar
Training	`train`	**3,129,592**	416,310	690,284	2,022,998	0	0
Development	`dev_in`	**50,000**	6,641	10,961	32,398	0	0
	`dev_out`	**50,000**	0	0	0	50,000	0
	`dev`	**100,000**	6,641	10,961	32,398	50,000	0
Evaluation	`eval_in`	**561,105**	74,406	123,487	363,212	0	0
	`eval_out`	**576,626**	0	0	0	525,967	50,659
	`eval`	**1,137,731**	74,406	123,487	363,212	525,967	50,659

6.2 List of attributes

- Attributes of train data set
- climate_pressure - climate pressure, mmHg
- climate_temperature - climate temperature, C
- cmc_0_0_0_1000 - temperature at 1000 isobaric level, K
- cmc_0_0_0_2 - temperature at 2m, K
- cmc_0_0_0_2_grad - difference between temperatures on adjacent horizons at 2m, K
- cmc_0_0_0_2_interpolated - temperature at 2m interpolated between horizons, K
- cmc_0_0_0_2_next - temperature at 2m for next horizon, K
- cmc_0_0_0_500 - temperature at 500 isobaric level, K
- cmc_0_0_0_700 - temperature at 700 isobaric level, K
- cmc_0_0_0_850 - temperature at 850 isobaric level, K
- cmc_0_0_0_925 - temperature at 925 isobaric level, K
- cmc_0_0_6_2 - dew point temp at 2m, K
- cmc_0_0_7_1000 - dew point depression at 1000 isobaric level, K
- cmc_0_0_7_2 - dew point depression at 2m, K
- cmc_0_0_7_500 - dew point depression at 500 isobaric level, K
- cmc_0_0_7_700 - dew point depression at 700 isobaric level, K
- cmc_0_0_7_850 - dew point depression at 850 isobaric level, K

- cmc_0_0_7_925 - dew point depression at 925 isobaric level, K
- cmc_0_1_0_0 - absolute humidity from 0 to 1
- cmc_0_1_11_0 - snow depth, m
- cmc_0_1_65_0 - rain accumulated from cmc gentime, mm
- cmc_0_1_65_0_grad - rain accumulated from cmc gentime difference between adjacent horizons, mm
- cmc_0_1_65_0_next - rain accumulated from cmc gentime for next horizon, mm
- cmc_0_1_66_0 - snow accumulated from cmc gentime, mm
- cmc_0_1_66_0_grad - snow accumulated from cmc gentime difference between adjacent horizons, mm
- cmc_0_1_66_0_next - snow accumulated from cmc gentime for next horizon, mm
- cmc_0_1_67_0 - ice rain accumulated from cmc gentime, mm
- cmc_0_1_67_0_grad - ice rain accumulated from cmc gentime difference between adjacent horizons, mm
- cmc_0_1_67_0_next - ice rain accumulated from cmc gentime for next horizon, mm
- cmc_0_1_68_0 - iced graupel accumulated from cmc gentime, mm
- cmc_0_1_68_0_grad - iced graupel accumulated from cmc gentime difference between adjacent horizons, mm
- cmc_0_1_68_0_next - iced graupel accumulated from cmc gentime for next horizon, mm
- cmc_0_1_7_0 - instant precipitation intensivity, mm/h
- cmc_0_2_2_10 - wind U component at 10m, m/s
- cmc_0_2_2_1000 - wind U component at 1000 isobaric level, m/s
- cmc_0_2_2_500 - wind U component at 500 isobaric level, m/s
- cmc_0_2_2_700 - wind U component at 700 isobaric level, m/s
- cmc_0_2_2_850 - wind U component at 850 isobaric level, m/s
- cmc_0_2_2_925 - wind U component at 925 isobaric level, m/s
- cmc_0_2_3_10 - wind V component at 10m, m/s
- cmc_0_2_3_1000 - wind V component at 1000 isobaric level, m/s
- cmc_0_2_3_500 - wind V component at 500 isobaric level, m/s
- cmc_0_2_3_700 - wind V component at 700 isobaric level, m/s
- cmc_0_2_3_850 - wind V component at 850 isobaric level, m/s
- cmc_0_2_3_925 - wind V component at 925 isobaric level, m/s
- cmc_0_3_0_0 - surface pressure, Pa
- cmc_0_3_0_0_next - next horizon surface pressure, Pa

- cmc_0_3_1_0 - sea level pressure, Pa
- cmc_0_3_5_1000 - geopotential height at 1000 isobaric level, gpm (geopotential meter)
- cmc_0_3_5_500 - geopotential height at 500 isobaric level, gpm
- cmc_0_3_5_700 - geopotential height at 700 isobaric level, gpm
- cmc_0_3_5_850 - geopotential height at 850 isobaric level, gpm
- cmc_0_3_5_925 - geopotential height at 925 isobaric level, gpm
- cmc_0_6_1_0 - cloudiness, % from 0 to 100
- cmc_available - is there any data from cmc
- cmc_horizon_h - cmc horizon, h
- cmc_precipitations - avg precipitations rate between adjacent horizons, mm/h
- cmc_timedelta_s, difference between cmc and forecast time, s
- gfs_2m_dewpoint - dew point temperature at 2m, C
- gfs_2m_dewpoint_grad - dew point temperature at 2m difference between horizons, C
- gfs_2m_dewpoint_next - dew point temperature on next horizon, C
- gfs_a_vorticity - absolute vorticity at height 100000 Pa, s-1
- gfs_available - is there any data from gfs
- gfs_cloudness - sum of 3 level cloudiness, from 0 to 3
- gfs_clouds_sea - Cloud mixing ratio at level 1000mb, kg/kg 0.0
- gfs_horizon_h - gfs horizon, h
- gfs_humidity - relative humidity at 2m, %
- gfs_precipitable_water - total precipitable water, kg m^-2
- gfs_precipitations - avg precipitations rate between adjacent horizons, mm/h
- gfs_pressure - surface pressure, mmHg
- gfs_r_velocity - vertical Velocity at 1000mb, Pa/s
- gfs_soil_temperature - soil temperature at 0.0-0.1 m, C
- gfs_soil_temperature_available - is there gfs soil temp data
- gfs_temperature_10000 - temp at vertical level at 10000 hectopascals, C
- gfs_temperature_15000 - temp at vertical level at 15000 hectopascals, C
- gfs_temperature_20000 - temp at vertical level at 20000 hectopascals, C
- gfs_temperature_25000 - temp at vertical level at 25000 hectopascals, C
- gfs_temperature_30000 - temp at vertical level at 30000 hectopascals, C
- gfs_temperature_35000 - temp at vertical level at 35000 hectopascals, C
- gfs_temperature_40000 - temp at vertical level at 40000 hectopascals, C
- gfs_temperature_45000 - temp at vertical level at 45000 hectopascals, C

- gfs_temperature_5000 - temp at vertical level at 5000 hectopascals, C
- gfs_temperature_50000 - temp at vertical level at 50000 hectopascals, C
- gfs_temperature_55000 - temp at vertical level at 55000 hectopascals, C
- gfs_temperature_60000 - temp at vertical level at 60000 hectopascals, C
- gfs_temperature_65000 - temp at vertical level at 65000 hectopascals, C
- gfs_temperature_7000 - temp at vertical level at 7000 hectopascals, C
- gfs_temperature_70000 - temp at vertical level at 70000 hectopascals, C
- gfs_temperature_75000 - temp at vertical level at 75000 hectopascals, C
- gfs_temperature_80000 - temp at vertical level at 80000 hectopascals, C
- gfs_temperature_85000 - temp at vertical level at 85000 hectopascals, C
- gfs_temperature_90000 - temp at vertical level at 90000 hectopascals, C
- gfs_temperature_92500 - temp at vertical level at 92500 hectopascals, C
- gfs_temperature_95000 - temp at vertical level at 95000 hectopascals, C
- gfs_temperature_97500 - temp at vertical level at 97500 hectopascals, C
- gfs_temperature_sea - temp at 2m, C
- gfs_temperature_sea_grad - temperature difference adjacent horizons at 2m
- gfs_temperature_sea_interpolated - gfs_temperature_sea_interpolated between horizons, C
- gfs_temperature_sea_next - next horizon temperature at 2m, C
- gfs_timedelta_s - difference between gfs and forecast time, s
- gfs_total_clouds_cover_high - cloud coverage (between horizons, divisible by 6) at high level, %
- gfs_total_clouds_cover_low - cloud coverage (between horizons, divisible by 6) at low level, %
- gfs_total_clouds_cover_low_grad - difference between low level cloud coverage on adjacent horizons, %
- gfs_total_clouds_cover_low_next - next horizon cloud coverage (between horizons, divisible by 6) at low level, %
- gfs_total_clouds_cover_middle - cloud coverage (between horizons, divisible by 6) at middle level, %
- gfs_u_wind - 10 meter U wind component, m/s
- gfs_v_wind - 10 meter V wing component, m/s
- gfs_wind_speed - wind velocity, sqrt(gfs_u_wind ** 2 + gfs_v_wind ** 2), m/s

- sun_elevation - sun height proxy above horizon (without corrections for precession and diffraction)
- topography_bathymetry - height above or below sea level, m
- wrf_available - is there any data from wrf 1
- wrf_graupel - avg graupel rate between two horizons, mm/h
- wrf_hail - hail velocity on two horizons, mm/h 0.0
- wrf_psfc - pressure, Pa
- wrf_rain - avg rain rate between two horizons, mm/h
- wrf_rh2 - humidity from 0 to 1
- wrf_snow - avg snow rate between two horizons, mm/h
- wrf_t2 - temperature at 2m, K
- wrf_t2_grad - difference between temperatures at 2m on adjacent horizons, K
- wrf_t2_interpolated - wrf_t2_interpolated between horizons, K
- wrf_t2_next - next horizon temperature at 2m, K
- wrf_wind_u - wind U component, m/s
- wrf_wind_v - wind V component, m/s

6.3 Visualization of the data

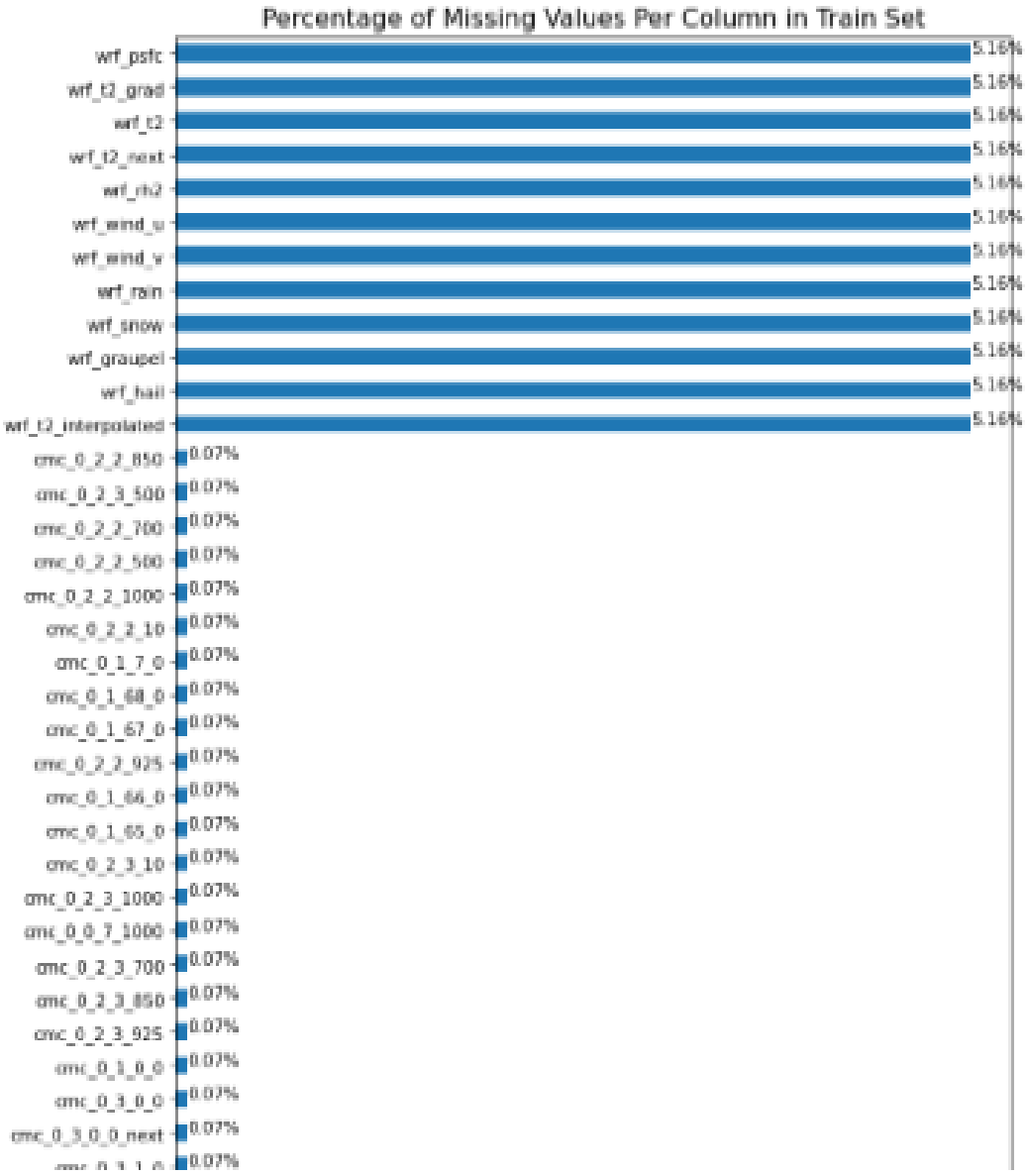

Figure 2 : Percentage of Missing Values per column in the test data set

From figure 2 we can infer that training dataset has some missing value and also WRF station dataset has more missing values than CMC station.

7. DATASET EXPLORATION

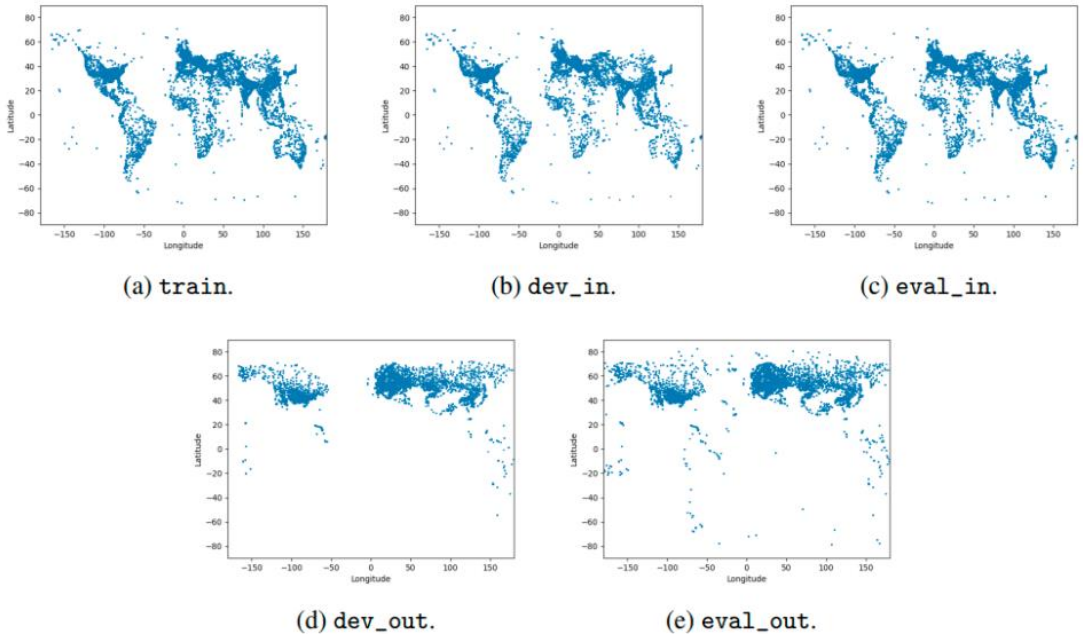

Figure 3 : Location of samples of Weather Prediction Dataset(Distribution of data in dataset)

Figure 3 shows the shift in the samples' locations (latitudes/longitudes) between training, development, and evaluation datasets. The location shift is a natural result of the climate shifts present in the datasets where the training data tends to correspond to warmer parts of the world, whereas the development and evaluation datasets include colder

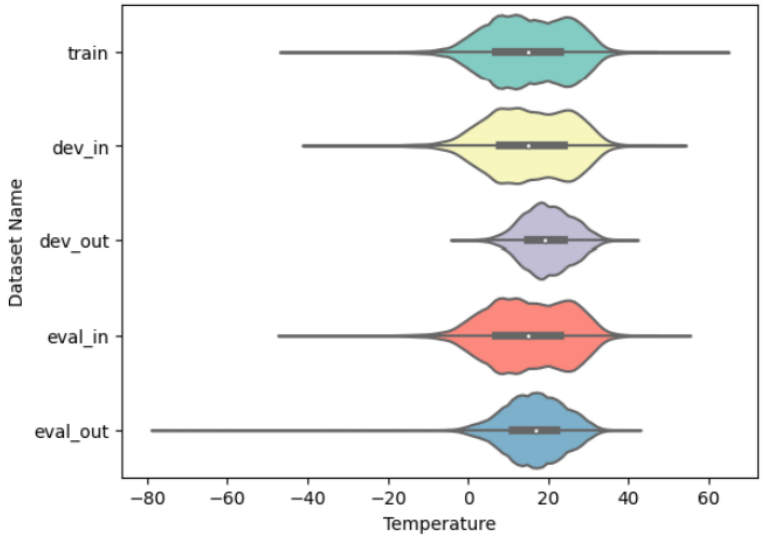

Figure 4 : Distribution of Temperature in the dataset

Figure 4 depicts the shift in the target temperatures between the training, development, and evaluation datasets. It is clear that the temperature distribution is different for dev_out and eval_out compared to the in-domain sets.

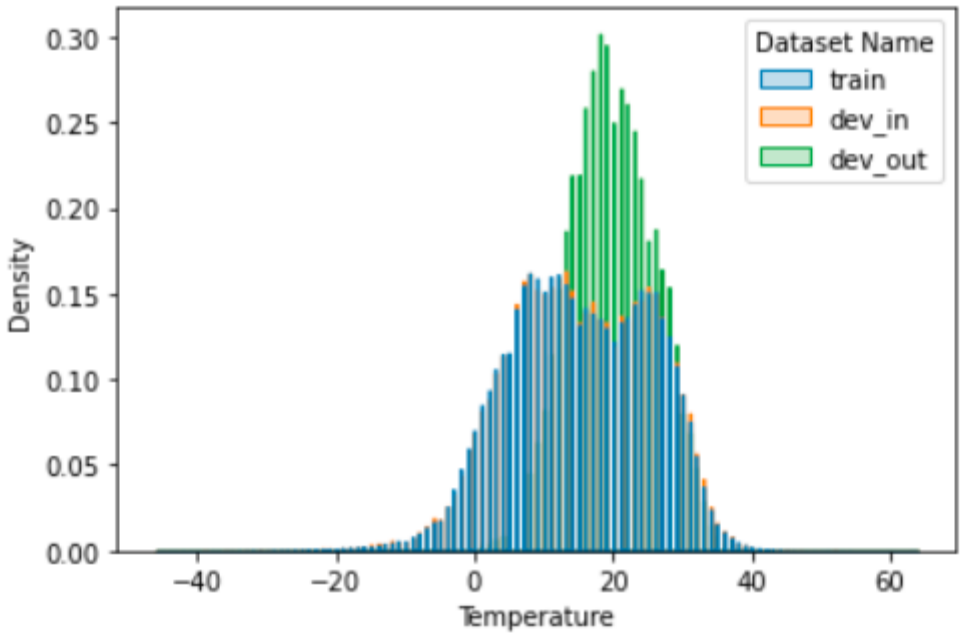

Figure 5 : Distributuion of Temperature in the dataset

From figure 5 we can infer that dev_out data is more narrow and train data is more distributed

8. PRE-PROCESSING OF DATA:

8.1 Data Preprocessing - 1

We found that there was -9999 value in gfs_soil_temperature column where NaN should be there. Therefore we replaced -9999 value with NaN in that column.We checked the percentage of missing values in dataset. We found that there is some missing values in some columns we replaced them with mean of that column.We dropped wrf_hail column as it has all zero value. We dropped gfs_ soil_ temperature, cmc_available, gfs_available, wrf_available, gfs_soil_temperature_available as they are not needed.We found that some temperature values are of celcius and some in kelvin so we converted all celcius scale column to kelvin scale.

Data Preprocessing - 2

We found that there was -9999 value in gfs_soil_temperature column where NaN should be there. Therefore we replaced -9999 value with NaN in that column.We checked the percentage of missing values in dataset. We found that there is some missing values in some columns we replaced them with mean of that column.We dropped wrf_hail column as it has all zero value. We dropped gfs_soil_temperature , cmc_available, gfs_available, wrf_available, gfs_soil_temperature_available as they are not needed.We found that some temperature values are of celcius and some in kelvin so we converted all celcius scale column to kelvin scale. We normalized all columns.

9. LEARNING MODELS

Ridge Regression: The basic idea of ridge regression is to introduce a little bias so that the variance can be substantially reduced, which leads to a lower overall MSE. Multicollinearity can be reduced without completely removing some predictor variables from the model. In general, the predictor variables that are least influential in the model will shrink towards zero the fastest. The advantage of ridge regression compared to least squares regression lies in the bias variance tradeoff.

Lasso Regression : The basic idea of ridge regression is to introduce a little bias so that the variance can be substantially reduced, which leads to a lower overall MSE. Multicollinearity can be reduced without completely removing some predictor variables from the model. Lasso regression and ridge regression are both known as regularization methods because they both attempt to minimize the sum of squared residuals (RSS) along with some penalty term. In cases where only a small number of predictor variables are significant, lasso regression tends to perform better because it's able to shrink insignificant variables completely to zero and remove them from the model.

Elastic net regression : The main purpose of ElasticNet Regression is to find the coefficients that minimize the sum of error squares by applying a penalty to these coefficients. ElasticNet combines L1 and L2 (Lasso and Ridge) approaches. It performs a more efficient regularization process.

Decision Tree : The decision tree breaks down the data set into smaller subsets. The topmost node in the decision tree is the best predictor called the root node. It employs a top to down approach and splits are made based on standard deviation. Just for a quick revision. A higher value of dispersion or variability means greater is the standard deviation indicating the greater spread of the data points from the mean value.

Gradient Boost Regressor : Gradient boosting is an ensemble of decision tree algorithms. A major problem of gradient boosting is that it is slow to train the model. At each step, a new tree is trained against the negative gradient of the loss function, which is analogous to (or identical to, in the case of least-squares error) the residual error.

HistGradient Boost Regressor : Training the trees that are added to the ensemble can be dramatically accelerated by discretizing the continuous input variables to a few hundred unique values. Histogram-based gradient boosting is a technique for training faster decision trees used in the gradient boosting ensemble.

CatBoost Regressor : CatBoost builds upon the theory of decision trees and gradient boosting. CatBoost grows oblivious trees, which means that the trees are grown by imposing the rule that all nodes at the same level, test the same predictor with the same condition. One of CatBoost's core edges is its ability to integrate a variety of different data types, such as images, audio, or text features into one framework. It can handle categorical data.

10. COMPARISON OF MODELS

Table 2 : Comparison of model performance for data preprocessing one and two.

Models	RMSE value 1	RMSE value 2
Ridge Regressor	14.24	**5.04**
Lasso Regression	9.05	**5.4**
HistGradient Boosting Regressor	2.31	3.85
CatBoost Regressor	**2.09**	4.35

From table 2 we can infer that the data after normalization gives better results for linear models but not for tree based models

11. CONCLUSIONS:

We can conclude that Catboost is the most efficient model for the prediction with data pre-processing done for data without normalization .The RMSE score for development data after the submission for first phase of challenge is 2.09. The RMSE score for evaluation data provided during 2nd phase of the challenge is 2.063.

In first Phase we secured the rank of 44 of 127.In second Phase we secured the rank of 15 of 19.

12. REFERENCES

1.https://arxiv.org/pdf/2107.07455.pdf

2.2107.07455.pdf

3. https://towardsdatascience.com/

4. https://machinelearningmastery.com/uncertainty-in-machine-learning/

5. https://blog.paperspace.com/implementing-gradient-boosting-regression-python/#:~:text=%20The%20high%20level%20steps%20that%20we%20follow,function%204%20Minimize%20the%20loss%20function%20More%20

6.Decision Tree Regression: What You Need to Know in 2021 | upGrad blog

7.ElasticNet Regression Fundamentals and Modeling in Python | by Kerem Kargın | MLearning.ai | Medium

8.Introduction to Lasso Regression - Statology

9.What is the Bias-Variance Tradeoff in Machine Learning? (statology.org)