horizontal line

Average Temperature Prediction

# OVERVIEW

The objective given the Rain in australia data set, was to predict the average temperature from the input features.

# Dataset

The dataset <https://www.kaggle.com/jsphyg/weather-dataset-rattle-package> had 24 columns,

8 object type and 16 column type variables.The visualizations,frequency count and other statistical information is present in the report. The “Rain Tomorrow” and “Risk MM” columns were dropped and were not used as we would not know in advance if it would rain the next day,and the Risk MM was dropped due to the instructions in the dataset information.

The minimum and maximum temperature were combined to form the target variable average temperature and were later dropped from the input features as the targets were built from them, and using them in the input features might have forced the model to simply learn the function average of the two columns, ignoring all the other input features.

# Pretrain Steps

1. Basic statistical information of the data was extracted to gain information about the variables.
2. Visualizations of the data were done to analyze outliers, spread and distribution, and finding out relations between the variables to analyze correlations.
3. From the visualizations some assumptions about the nature of the variables were made which were later on investigated by further separate visualizations, some of which were fairly obvious such as negative correlation between sunshine and cloudiness.
4. The data was treated for missing values, median strategy used for numeric variables, and mode strategy for the categorical values.
5. The variables were analyzed for invalid data such as negative values in evaporation or date.
6. After experimentation with some methods median strategy was used to treat outliers with the Interquartile Range (IQR) criterion used to identify outliers.
7. Scaling of the numerical data and encoding of categorical values was done.
8. Lasso regression coefficients were visualized for feature importance., the Cloudiness at 3 PM, and the temperature at 9 AM and 3 PM were found to be the most important features.
9. A pipeline was implemented to read and perform the processing steps in one module.

# Training

1. A module was implemented to evaluate and train regression models for the task, with Decision tree,Random forest,Ridge and linear regressors included.
2. All models were trained on default hyperparameter setting and the best model from this was to be tuned for best hyperparameters.
3. 70/30 split was used as directed, with R2 score, Mean squared Error and Explained Variance Score were used to evaluate the models.
4. The basic Linear Regression was found to be the best trained model with respect to all the evaluation criteria with a 1.17e-28 MSE, 1.0 R2 score and 1.0 EVS.
5. Grid search was not needed for the hyperparameter search.
6. The evaluation scores were visualized for comparison.

# Deployment

1. The trained model was extracted with pickle library, and saved.
2. Python and Flask were used for the web interface as it provides easy access to the used libraries and frameworks which were to be used in the preprocessing on the live data.
3. The encoding and preprocessing steps had to be the same for the new data for prediction, and hence a standard.csv file after some experiments was created to ensure the consistency in the live data for prediction which is loaded in the app.

# Improvements

Some improvements can be made in the work, such as experimenting with different scaling methods,outlier treatment,selecting top N features to reduce features, writing custom transformer functions for treatment of the mixed nature of the data to build a scikit learn pipeline,efficiently handle live prediction and validations and dropdowns in the html forms.