Metis Classification Module

Predicting Rain in Australia

Abstract

Australia has been experiencing severe weather changes from the past 2 years. In 2019, the country faced devastating forest fires which burnt tens of millions of acres and animals. Currently, parts of the country are facing massive flooding because of heavy rainfalls. Many lost homes and lives as a result of this unexpected calamity. There is a need to reevaluate how rains are being predicted to better the community for the future. In order to get started, Australian Government Bureau of Meteorology has approached me to build a better model that can predict if it's going to rain the next day or not.

Design

Data was downloaded from Kaggle as a CSV file into pandas. Data was cleaned as part of which nulls were replaced with average values and appropriate data types were assigned. Features were selected based on correlation matrix, VIF scores and distribution of the target variable over the features. Because the impact of predicting rains is related to both forest fires and flooding, the metric to gauge the accuracy of the models is going to be F1. I was able to fit my data into 3 types of models: KNN, Logistic Regression, and Random Forest, with logistic regression as the model with the highest accuracy obtained.

Data

Data has 145460 rows and 23 columns which include location, min/max temperatures, pressure, humidity, windspeed, wind direction and **rain tomorrow(Y/N) .**

Algorithms

- Data Cleaning and EDA
  - Data Cleaning included handling nulls by replacing with values grouped by year and location
  - Correlation matrix, VIFs and pairplot were carefully reviewed. Features which did not contribute to high collinearity were selected.
  - Imbalance of the target variable was noted.
- Modelling
  - I was able to model my data through 3 different algorithms: KNN, Logistic Regression, and Random Forest
  - For KNN and Logistic Regression, I used get_dummies for categorical features and I scaled my numeric features
  - Class imbalance was addressed through stratified test/train/val split and oversampling
  - GridSearchCV was used to tune all the models and best parameters were used to predict on test
  - I ultimately picked Logistic Regression for it's high F1 score and computational efficiency
- Visualizing Data
  - Created confusion matrices, AUC/ROC curves using matplotlib
  - Tableau dashboard was also made to review trends and filter through the years and locations.

Logistic Regression which is my selected model has performed with an F1 score of 0.83

Tools

- Scikit-learn for modeling
- Matplotlib and Seaborn for plotting
- Tableau for interactive visualizations

Communication

Presentation to the class