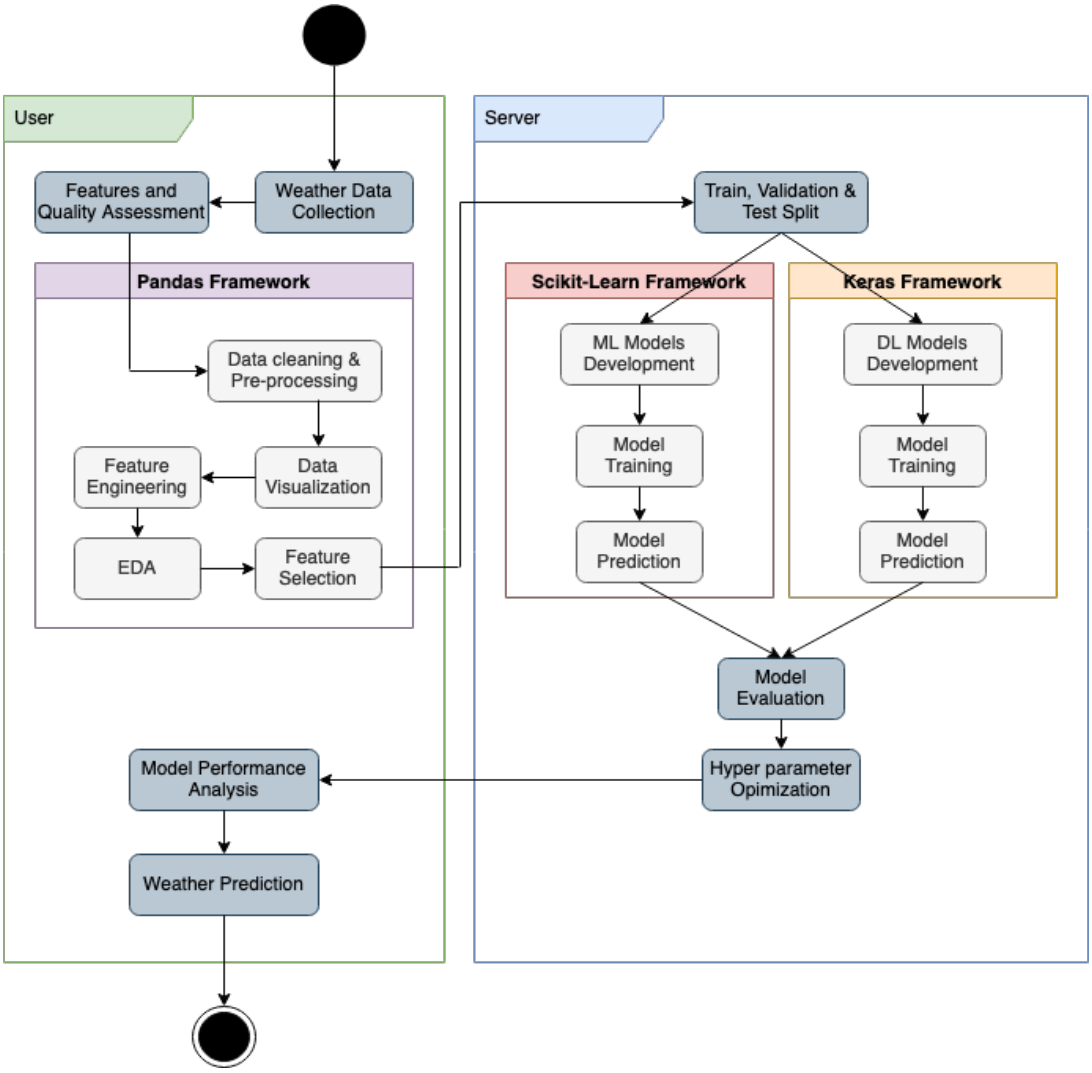
Weather Prediction based on Weather Data using Data Mining

Architectural Decisions Document

1 Architectural Components Overview



Weather Data predictive System Reference Architecture.

1.1 Data Source

1.1.1 Technology Choice

- The dataset is based on the weather records of different cities in Australia along with atmospheric parameters.
- The data has been collected from Kaggle.
- There are mainly two weather datasets i.e.,
    - Weather Dataset [Having the target attribute “RainTomorrow”]
    - Unknown Weather Dataset [Without having target attribute “RainTomorrow”]
    - The Weather dataset consists of 21 features and 1 target variable.

1.1.2 Justification

CSV files are a convenient and effective way to import and export datasets since they are universal and easy to create, read and manipulate in different ways. They can be easily updated by adding new weather records directly from online resources and updating the loaded file in the GitHub repository.

1.2 Enterprise Data

1.2.1 Technology Choice

This component is not needed in this project.

1.2.2 Justification

We are using a CSV file as a data source since our data does not require frequent updates, therefore, a cloud-based solution for enterprise data is not needed.

1.3 Streaming analytics

1.3.1 Technology Choice

This component is not needed in this project.

1.3.2 Justification

The weather dataset includes the weather history of around 99,516 different times in the form of atmospheric records. Therefore, we assume that the size of this sample is representative of the population and that it is sufficient to train classification models to achieve significant prediction accuracy. This means that this data does not lose value quickly but requires real-time updates to achieve the desired performance.

1.4 Data Integration

1.4.1 Technology Choice

- For data preprocessing, we handled the dataset as a Pandas data frame object, a 2-dimensional labeled data structure with the index for rows and columns.
- The dataset needed imputation of missing values in two features which were handled using mean or mode imputation depending on features type. These steps used basic python modules and libraries like NumPy.
- We used OneHotEncoder from sklearn.preprocessing library to scale numeric features and one-hot-encode categorical features respectively.

1.4.2 Justification

Panda's data frame structure is fast and has high productivity and performance, and it is suitable for our two-dimensional dataset. We used the sklearn.preprocessing package to perform the last feature transformation and feature creation steps since it provides several efficient functions for one-hot encoding of data.

1.5 Data Repository

1.5.1 Technology Choice

The CSV file with the dataset is saved in the GitHub repository to be used in the analysis.

1.5.2 Justification

Based on the nature of the data and the objective of the analysis, the dataset will not require frequent updates for future models' training, therefore, the GitHub repository is a sufficient and efficient technology for our case.

1.6 Discovery and Exploration

1.6.1 Technology Choice

In order to visualize the data in a structured manner, the Pandas framework has been used. The NumPy package has also been used.
During the exploration of data, I have used seaborn, and matplotlib Python packages to visualize the distribution of various features to understand their effectiveness for further processing.

1.6.2 Justification

These visualizations provide a brief idea about all the features along with the frequency of the corresponding values. During this phase, the following analysis has been performed.

- Maximum and Minimum Temperature Distribution
- Wind Gust Distribution
- Wind Speed Distribution
- Humidity Distribution
- Pressure Distribution
- Cloud Distribution
- Temperature Distribution

1.7 Actionable Insights

1.7.1 Technology Choice

We have used Scikit-Learn and Keras's framework to develop different machine learning and deep learning models. We have imported the following model classes along with the different accuracy measures –

- LogisticRegression
- KNeighborsClassifier
- DecisionTreeClassifier
- AdaBoostClassifier
- RandomForestClassifier
- accuracy_score
- classification_report
- confusion_matrix
- Dense

- Dropout
- RandomizedSearchCV

1.7.2 Justification

- The machine learning and deep learning classifiers help to predict the future atmospheric condition based on the selected features.
- The accuracy measures will be needed to perform the performance comparisons of these models.
- The classification report and confusion matrix will help to understand the model performance.
- The RandomizedSearchCV will help to find the best values of the hyperparameters in order to optimize the model performance.

1.8 Applications / Data Products

1.8.1 Technology Choice

The data produced for this project is an ipynb file of the analysis generated from the "IBM Advanced Capstone Project - Final.ipynb" Jupyter Notebook, which contains all code chunks and outputs.

1.8.2 Justification

The main motivation behind this work is to build an intelligent system to predict future weather conditions based on the given data. Some major objectives include –

- Exploratory Data Analysis to find the location-wise average trend of wind speed, humidity, pressure, maximum temperature, minimum temperature rainfall.
- Correlation Analysis to select the relevant features from the weather data.
- Predictive Analysis to predict the future rainy days from the weather data.
- Identification of the best classifier to predict the upcoming rainy days from unknown weather data.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

This component is not needed

1.9.2 Justification

The dataset used for this project is public, therefore no information governance and system management are needed.