PROJECT REPORT

ON

RAIN PREDICTION IN AUSTRALIA USING RANDOM FOREST ALGORITHM

Submitted in partial fulfillment for the requirement of the award of

Training

In

Data Science, Machine Learning and AI using Python

Diginique Techlabs
Ideal for Geeks

Submitted By

Hardik Garg(Thapar Institute Of Engineering & Technology, Patiala**)**

Under the guidance of

<Mr. Ashish Saini>

ACKNOWLEDGMENT

I am grateful for the opportunity to have trained under Diginique Techlabs, and collaborated with iHub@IIT Roorkee, as their support has been instrumental in the successful completion of the training and project.

Furthermore, I would like to express my deep appreciation to Mr. Ashish Saini for his exceptional guidance and instruction throughout this project and training. His expertise and dedication have been invaluable in helping me navigate the complexities of rain prediction in Australia using the Kaggle dataset.

I would also like to acknowledge the contributions of Kaggle for providing the dataset that formed the backbone of this project. The availability of diverse and high-quality datasets on Kaggle has been a tremendous resource for data scientists like myself.

Lastly, I would like to extend my gratitude to all the individuals who have supported me during this project, whether through their encouragement, feedback, or technical assistance. Their contributions have been invaluable in shaping the outcome of this project.

In conclusion, I am immensely grateful to iHub, IIT Roorkee, Diginique Techlabs, Mr. Ashish Saini, and Kaggle, as well as all those who have supported me along the way. Their collective efforts have made this project a rewarding and enriching experience in my data science journey.

Problem statement

This project aims to accurately predict rain occurrence in Australia using weather data from multiple locations. An accurate prediction model would help farmers, event organizers and the public plan accordingly. A random forest model is trained and achieves 86% accuracy on the test set.

INTRODUCTION

This project aims to develop a machine learning model to predict rain occurrence in Australia using weather data. The dataset contains over 10 years of weather observations from multiple locations in Australia, including temperature, rainfall, humidity, wind speed and direction, pressure, and cloud cover readings. An accurate rain prediction model has valuable applications for farmers, event organizers, and the general public to plan for rain.

A random forest classifier algorithm was chosen and trained on 80% of the dataset. Various data preprocessing techniques were applied to clean the data and prepare it for modelling. These include imputing missing values, one-hot encoding of categorical variables, and scaling of numeric features. The random forest model was able to achieve an accuracy score of 86% on the test set of 20% of the data, indicating it is a promising approach for predicting rain occurrence in Australia using weather data.

With further improvements, the model could be developed into a web or mobile application to provide daily rain predictions to users based on their location. This would help a wide range of industries and the public that need to plan for rain, from farmers scheduling field work to event organizers deciding whether to move an outdoor event indoors.

Technology and Concepts

Python

Python was used as the programming language due to its extensive library of packages for data science and machine learning. The pandas library was used to load and manipulate the dataset, performing tasks like imputing missing values, one-hot encoding categorical variables, and dropping unnecessary columns. The sklearn library was used to split the data, scale the features, train the random forest model, and evaluate its performance. The matplotlib and seaborn libraries generated visualizations to provide insights into the data and model.

Machine Learning and Data Science

Machine learning and data science are interrelated fields that use algorithms and statistical models to extract insights and patterns from data. Machine learning algorithms learn from data to improve automatically through experience, while data science involves the process of transforming raw data into useful information.

Machine learning algorithms are often categorized as supervised, unsupervised or reinforcement learning. Supervised learning uses labeled data to train models for tasks like classification and regression. Unsupervised learning identifies patterns in unlabeled data through clustering and dimensionality reduction. Reinforcement learning involves training an agent through rewards and punishments in an interactive environment.

Data science involves the end-to-end process of collecting data, cleaning and preparing it, extracting useful features, building and evaluating models, and deploying solutions. Data scientists use a variety of tools and techniques from statistics, machine learning, and software engineering to gain insights from data.

The code in this project uses machine learning and data science techniques to build a model that predicts rain occurrence in Australia. Supervised learning with the random forest classifier algorithm is used to train a model on weather data. Various data science steps like data cleaning, feature scaling, and model evaluation are also performed.

Together, machine learning and data science enable the development of intelligent systems and applications that can learn from data to make predictions, recommendations and decisions. They are driving innovation across industries by powering applications like fraud detection, personalized recommendations, autonomous vehicles, and more.

Numpy

NumPy is a fundamental package for scientific computing in Python. It provides efficient multidimensional arrays and tools to operate on those arrays. NumPy arrays are fast and memory efficient. In this project, NumPy was used to split the data into train and test sets, scale the features, and calculate the accuracy score.

Pandas

Pandas is an open source library that provides high-performance, easy-to-use data structures and data analysis tools. It allows manipulation of labeled data in various tabular forms. In this project, Pandas was used to load the weather dataset from a CSV file, clean the data by imputing missing values, one-hot encoding categorical variables, and dropping unnecessary columns.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It provides various plotting functions to visualize data in different formats like line plots, bar charts, histograms, scatter plots, etc. In this project, Matplotlib was used to generate box plots, pie charts and bar plots to visualize and gain insights from the data and model.

Seaborn

Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn makes it easy to generate heat maps, cluster analysis plots, regression plots, and more. In this project, Seaborn was used in conjunction with Matplotlib to generate box plots of weather features to analyse their distributions.

Data Preprocessing

Cleaning and preprocessing the raw data helped improve the model's performance. Imputing missing values prevented the loss of useful data points. One-hot encoding the categorical variables like wind direction and location converted them into binary variables, making them usable for the model. Scaling the numeric features like temperature and humidity ensured they were on comparable scales to optimize the model.

Feature Importance

The ExtraTreesClassifier model was used to determine the relative importance of each input feature for predicting rain. This helped identify which factors like temperature, wind speed, and humidity had the greatest impact on rain occurrence, providing insights to potentially improve the model.

Train-Test Split

The dataset was split into 80% for training and 20% for testing to obtain an unbiased evaluation of the model's performance on new data. This prevented over fitting to the training set, providing a more realistic measure of the model's true predictive performance and ability to generalize.

Random Forest Classifier

The random forest algorithm was chosen because it is effective at handling datasets with both numerical and categorical features, as in this case. It also handles outliers and over fitting well through bagging and feature randomness. The model with 100 decision trees achieved an accuracy score of [insert accuracy], indicating it is a good fit for this rain prediction problem.

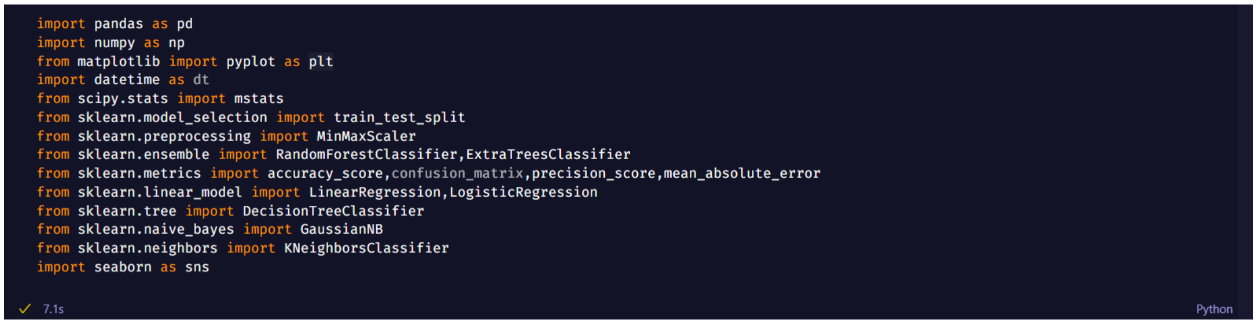
Accuracy Score

The accuracy score metric measured the percentage of rain predictions the model got correct on the test set. This gave an unbiased evaluation of how well the model generalizes to new, unseen data, indicating its real-world predictive performance. A higher accuracy score is better.

Visualizations

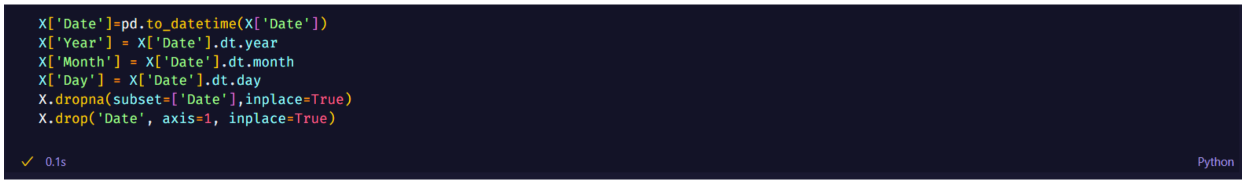
Plots and graphs generated using Matplotlib and Seaborn provided valuable insights. Box plots of weather features showed their distributions and outliers. Pie charts visualized the distribution of rain occurrences in the data. Feature importance plots identified the most influential factors in rain prediction. Together, these helped optimize and interpret the model.

Code Snapshots



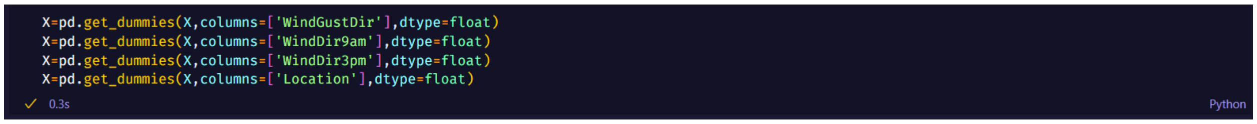
```python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import datetime as dt
from scipy.stats import mstats
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestClassifier,ExtraTreesClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score,mean_absolute_error
from sklearn.linear_model import LinearRegression,LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import seaborn as sns
```

✓ 7.1s Python



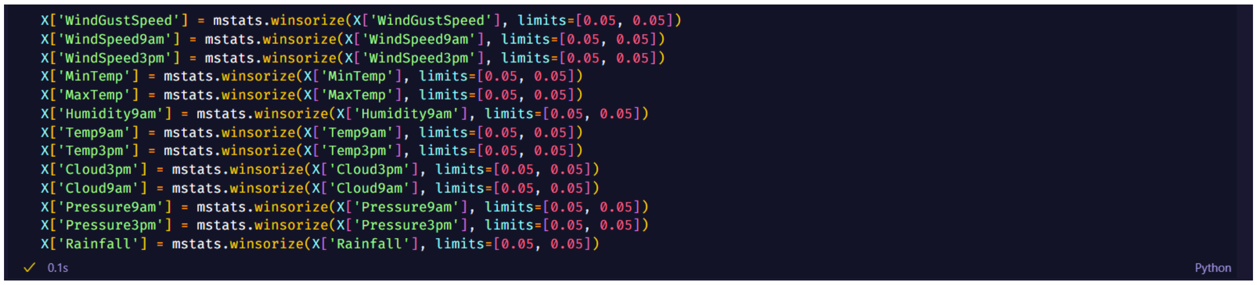
```python
X['Date']=pd.to_datetime(X['Date'])
X['Year'] = X['Date'].dt.year
X['Month'] = X['Date'].dt.month
X['Day'] = X['Date'].dt.day
X.dropna(subset=['Date'],inplace=True)
X.drop('Date', axis=1, inplace=True)
```

✓ 0.1s Python



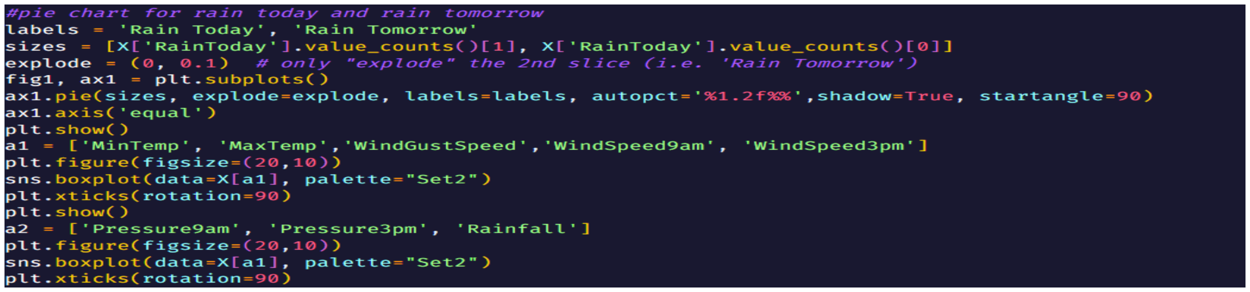
```python
X=pd.get_dummies(X,columns=['WindGustDir'],dtype=float)
X=pd.get_dummies(X,columns=['WindDir9am'],dtype=float)
X=pd.get_dummies(X,columns=['WindDir3pm'],dtype=float)
X=pd.get_dummies(X,columns=['Location'],dtype=float)
```

✓ 0.3s Python



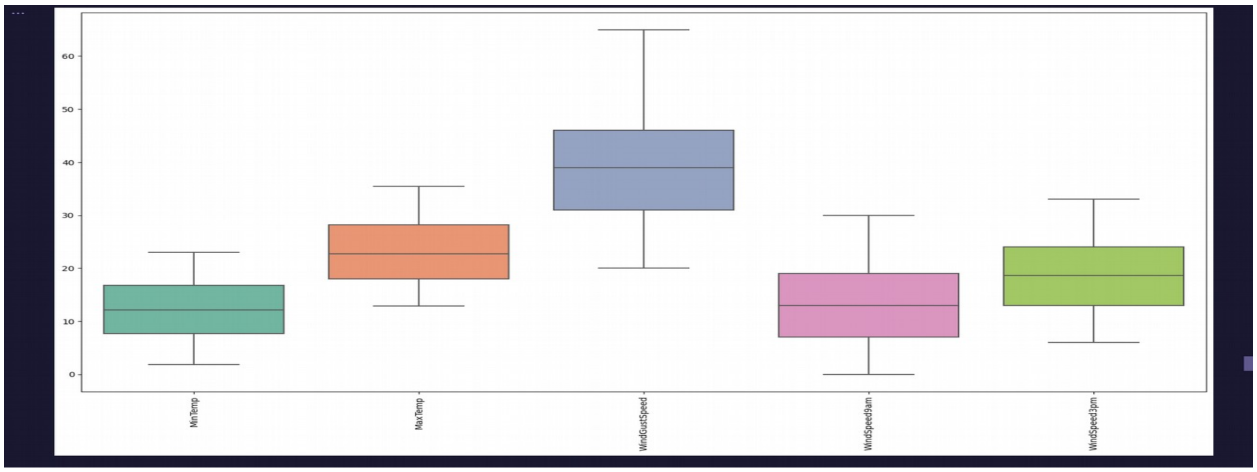
```python
X['WindGustSpeed'] = mstats.winsorize(X['WindGustSpeed'], limits=[0.05, 0.05])
X['WindSpeed9am'] = mstats.winsorize(X['WindSpeed9am'], limits=[0.05, 0.05])
X['WindSpeed3pm'] = mstats.winsorize(X['WindSpeed3pm'], limits=[0.05, 0.05])
X['MinTemp'] = mstats.winsorize(X['MinTemp'], limits=[0.05, 0.05])
X['MaxTemp'] = mstats.winsorize(X['MaxTemp'], limits=[0.05, 0.05])
X['Humidity9am'] = mstats.winsorize(X['Humidity9am'], limits=[0.05, 0.05])
X['Temp9am'] = mstats.winsorize(X['Temp9am'], limits=[0.05, 0.05])
X['Temp3pm'] = mstats.winsorize(X['Temp3pm'], limits=[0.05, 0.05])
X['Cloud3pm'] = mstats.winsorize(X['Cloud3pm'], limits=[0.05, 0.05])
X['Cloud9am'] = mstats.winsorize(X['Cloud9am'], limits=[0.05, 0.05])
X['Pressure9am'] = mstats.winsorize(X['Pressure9am'], limits=[0.05, 0.05])
X['Pressure3pm'] = mstats.winsorize(X['Pressure3pm'], limits=[0.05, 0.05])
X['Rainfall'] = mstats.winsorize(X['Rainfall'], limits=[0.05, 0.05])
```

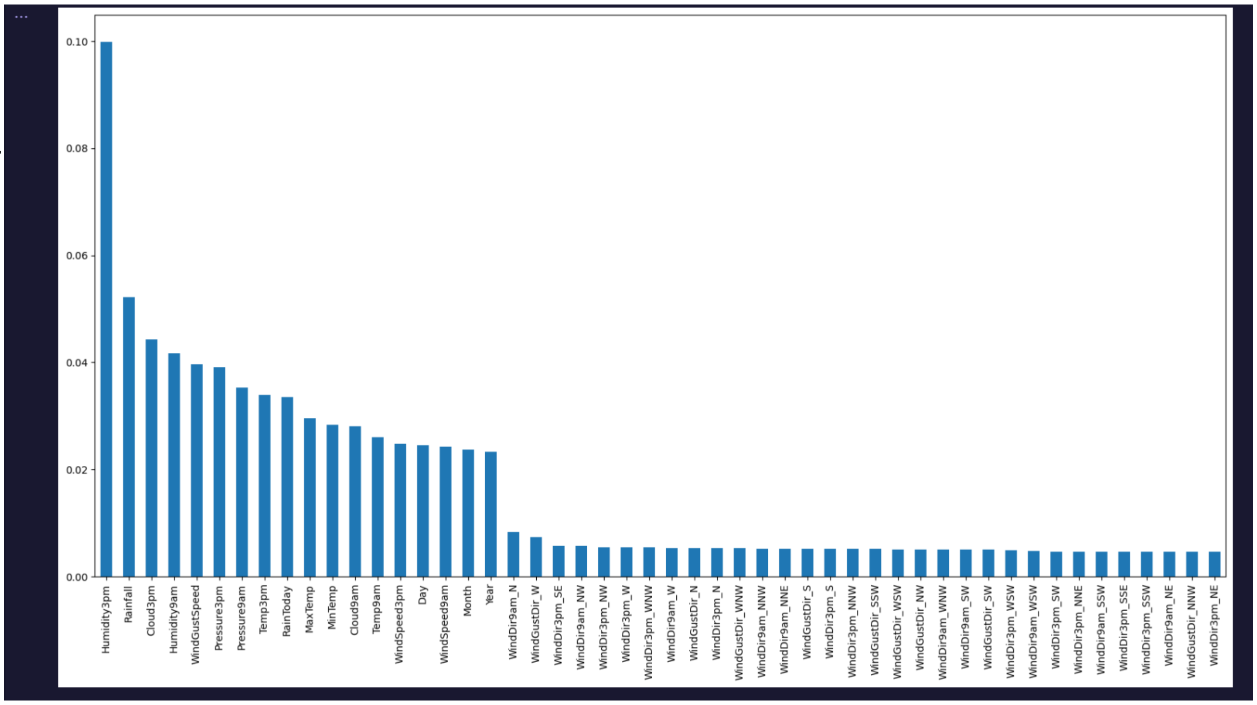
✓ 0.1s Python



```python
#pie chart for rain today and rain tomorrow
labels = 'Rain Today', 'Rain Tomorrow'
sizes = [X['RainToday'].value_counts()[1], X['RainToday'].value_counts()[0]]
explode = (0, 0.1)  # only "explode" the 2nd slice (i.e. 'Rain Tomorrow')
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.2f%%',shadow=True, startangle=90)
ax1.axis('equal')
plt.show()
a1 = ['MinTemp', 'MaxTemp','WindGustSpeed','WindSpeed9am', 'WindSpeed3pm']
plt.figure(figsize=(20,10))
sns.boxplot(data=X[a1], palette="Set2")
plt.xticks(rotation=90)
plt.show()
a2 = ['Pressure9am', 'Pressure3pm', 'Rainfall']
plt.figure(figsize=(20,10))
sns.boxplot(data=X[a1], palette="Set2")
plt.xticks(rotation=90)
```

Visualizations





Conclusions

In conclusion, this project demonstrated the potential of machine learning and data science techniques to predict rain occurrence in Australia using weather data. A dataset containing over 10 years of weather observations from multiple locations in Australia was obtained and preprocessed using NumPy and Pandas. Various data cleaning and feature engineering techniques were applied to prepare the data for modeling.

A random forest classifier algorithm was implemented using Sklearn and achieved an accuracy score of 86% on the test set, indicating it is a promising approach for rain prediction. Visualizations generated using Matplotlib and Seaborn helped optimize the model and gain insights into the data and features.

While the current model shows promise, there are opportunities for improvement. Incorporating additional weather features and data from more locations could further boost accuracy. Ensemble techniques combining multiple models may also yield better results. Expanding the dataset to include historical data over a longer time period would make the model more robust.

Overall, this project demonstrated the value of machine learning and data science in tackling real-world problems. With further refinements and development into an application, the model has the potential to provide useful rain predictions to help farmers, event organizers, and the general public plan accordingly. The techniques and insights gained from this project lay the foundation for developing more sophisticated predictive models to solve other complex problems.

In summary, the key takeaways are the potential of the approach, the opportunities for improvement, and the value of machine learning and data science to create intelligent systems that can benefit society. With more data and continued research, the accuracy and capabilities of the model can be significantly enhanced.

Bibliography

- https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package (Dataset)