1. **Rainfall Prediction – Weather Forecasting**

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere at a given place and using meteorology to project how the atmosphere will change. Rain Dataset is to predict whether or not it will rain tomorrow. The Dataset contains about 10 years of daily weather observations of different locations in Australia. Here, predict two things:

1. **Problem Statement:**

a) Design a predictive model with the use of machine learning algorithms to forecast whether or not it will rain tomorrow.

b) Design a predictive model with the use of machine learning algorithms to predict how much rainfall could be there.

# 2. Data Preprocessing:

Real-world data is often messy, incomplete, unstructured, inconsistent, redundant, sprinkled with wacky values. So, without deploying any Data Preprocessing techniques, it is almost impossible to gain insights from raw data.

Data preprocessing is a process of converting raw data to a suitable format to extract insights. It is the first and foremost step in the Data Science life cycle. Data Preprocessing makes sure that data is clean, organize and read-to-feed to the Machine Learning model.

# 3. Finding Categorical and Numerical Features in a Data set:

Cardinality check for Categorical features:

* The accuracy, performance of a classifier not only depends on the model that we use, but also depends on how we preprocess data, and what kind of data you’re feeding to the classifier to learn.
* Many Machine learning algorithms like Linear Regression, Logistic Regression, k-nearest neighbors, etc. can handle only numerical data, so encoding categorical data to numeric becomes a necessary step. But before jumping into encoding, check the cardinality of each categorical feature.
* Cardinality: The number of unique values in each categorical feature is known as cardinality.
* A feature with a high number of distinct/ unique values is a high cardinality feature. A categorical feature with hundreds of zip codes is the best example of a high cardinality feature.
* This high cardinality feature poses many serious problems like it will increase the number of dimensions of data when that feature is encoded. This is not good for the model.
* There are many ways to handle high cardinality, one would be feature engineering and the other is simply dropping that feature if it doesn’t add any value to the model.

# 4. Handling Missing Values:

Machine learning algorithms can’t handle missing values and cause problems. So they need to be addressed in the first place. There are many techniques to identify and impute missing values.

If a dataset contains missing values and loaded using pandas, then missing values get replaced with NaN(Not a Number) values. These NaN values can be identified using methods like isna() or isnull() and they can be imputed using fillna().

This process is known as Missing Data Imputation.

# 5. Outliers detection and treatment:

What is an outlier?

An Outlier is an observation that lies an abnormal distance from other values in a given sample. They can be detected using visualization(like boxplots, scatter plots), Z-score, statistical and probabilistic algorithms, etc. Outlier Treatment to remove outliers from Numerical Features:

**6. Exploratory Data Analysis:**

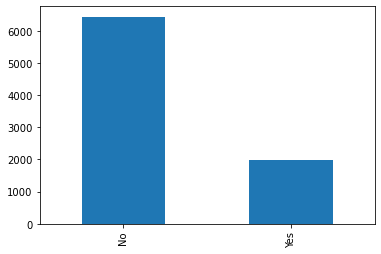
Exploratory Data Analysis(EDA) is a technique used to analyze, visualize, investigate, interpret, discover and summarize data. It helps Data Scientists to extract trends, patterns, and relationships in data.

Univariate Analysis:

* Exploring target variable:

rain['RainTomorrow']**.**value\_counts()**.**plot(kind**=**'bar')

<AxesSubplot:>



Looks like the Target variable is imbalanced. It has more ‘No’ values. If data is imbalanced, then it might decrease the performance of the model. As this data is released by the meteorological department of Australia, it doesn’t make any sense when we try to balance the target variable, because the truthfulness of data might decrease. So, let me keep it as it is.

Bi-variate Analysis:

* Sunshine vs Rainfall:

# 7. Encoding of Categorical Features:

Most Machine Learning Algorithms like Logistic Regression, Support Vector Machines, K Nearest Neighbours, etc. can’t handle categorical data. Hence, these categorical data need to converted to numerical data for modeling, which is called Feature Encoding.

There are many feature encoding techniques like One code encoding, label encoding. But in this particular blog, I will be using replace() function to encode categorical data to numerical data.

# 8. Correlation:

Correlation is a statistic that helps to measure the strength of the relationship between two features.

It is used in bivariate analysis. Correlation can be calculated with method corr() in pandas.

# 9. Splitting data into Independent Features and Dependent Features:

For feature importance and feature scaling, we need to split data into independent and dependent features.

**X = rain.drop(['RainTomorrow'],axis=1)**

**y = rain['RainTomorrow']**

In the above code,

* X – Independent Features or Input features
* y – Dependent Features or target label

Feature Importance:

* Machine Learning Model performance depends on features that are used to train a model. Feature importance describes which features are relevant to build a model.
* Feature Importance refers to the techniques that assign a score to input/label features based on how useful they are at predicting a target variable. Feature importance helps in Feature Selection.

We’ll be using ExtraTreesRegressor class for Feature Importance.

This class implements a meta estimator that fits a number of randomized decision trees on various samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

**10. Splitting Data into training and testing set:**

*#train\_test\_split() is a method of model\_selection class used to split data into training and testing sets.*

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X,y, test\_size **=** 0.2, random\_state **=** 0)

*#Length of Training and Testing set:*

print("Length of Training Data: {}"**.**format(len(X\_train)))

print("Length of Testing Data: {}"**.**format(len(X\_test)))

Length of Training Data: 6740

Length of Testing Data: 1685

# 11. Feature Scaling:

Feature Scaling is a technique used to scale, normalize, standardize data in range(0,1). When each column of a dataset has distinct values, then it helps to scale data of each column to a common level. Standard Scaler is a class used to implement feature scaling.

# 12. Model Building:

Using a Logistic Regression algorithm to build a predictive model to predict whether or not it will rain tomorrow in Australia.

* Logistic Regression: It is a statistic-based algorithm used in classification problems.
* It allows us to predict the probability of an input belongs to a certain category.
* It uses the logit function or sigmoid function as a core.

According to the Data science community, logistic regression can solve 60% of existing classification problems.

# 13. Model Training:

Sklearn library has a module called linear\_model, which provides LogisticRegression class to train a model or a classifier and test it.

**14. Evaluating Model Performance:**

accuracy\_score() is a method used to calculate the accuracy of a model prediction on unseen data.

In [33]:

**from** sklearn.metrics **import** accuracy\_score

print("Accuracy Score: {}"**.**format(accuracy\_score(y\_test,y\_pred)))

Accuracy Score: 0.8367952522255193

**15. Receiver operating characteristic(ROC) curve:**

* The ROC curve is an evaluation metric used in binary classification problems to know the performance of the classifier.
* It is a curve plotted between True Positive Rate(TPR) and False Positive Rate(FPR) at various thresholds.
* ROC graph summarizes all the confusion matrices produced at different threshold values.
* ROC curve is used to determine which threshold value is best for Logistic Regression in order to classify classes.

# 16. Cross-Validation:

Let’s find out whether model performance can be improved using Cross-Validation Score.

**from** sklearn.model\_selection **import** cross\_val\_score

scores **=** cross\_val\_score(classifier\_logreg, X\_train, y\_train, cv **=** 5, scoring**=**'accuracy')

print('Cross-validation scores:{}'**.**format(scores))

print('Average cross-validation score: {}'**.**format(scores**.**mean()))

Cross-validation scores:[0.82937685 0.83456973 0.83308605 0.83679525 0.81973294]

Average cross-validation score: 0.8307121661721067

The mean accuracy score of cross-validation is almost the same as the original model accuracy score which is 0.8445.

So, the accuracy of the model may not be improved using Cross-validation.

**17. Results and Conclusion:**

* The logistic Regression model accuracy score is 0.84. The model does a very good job of predicting.
* The model shows no sign of Underfitting or Overfitting. This means the model generalizing well for unseen data.
* The mean accuracy score of cross-validation is almost the same as the original model accuracy score. So, the accuracy of the model may not be improved using Cross-validation.

**18. Save Model and Scaling object with Pickle:**

Pickle is a python module used to serialize and deserialize objects. It is a standard way to store models in machine learning so that they can be used anytime for prediction by unpickling.

Here scaling object is stored in a pickle file, which can be used to standardize real-time and unseen data fed by users for prediction.

**import** pickle

**with** open('scaler.pkl', 'wb') **as** file:

pickle**.**dump(scaler, file) *# here scaler is an object of StandardScaler class.*

Save the logistic regression model in a pickle file, so that it can be used for predictions later.