# DATA-DRIVEN WEATHER FORECAST: OPTIMIZING WEATHER PREDICTIONS WITH MACHINE LEARNING

## B.SC PHYSICS PROJECT REPORT
## COMPUTATIONAL PHYSICS

**PREPARED BY**

ABHIRAMI BS – RGAVSPH001
ABHISHEK PARTHAN – RGAVSPH004
ABINA K - RGAVSPH007
MUHAMMED FARIS P- RGAVSPH017
MUHAMMED SHINAS- RGAVSPH006

**PROJECT ADVISOR**

MR. SANAL V

SREE GOKULAM
ARTS AND SCIENCE COLLEGE

**DEPARTMENT OF PHYSICS**

**SREE GOKULAM ARTS AND SCIENCE COLLEGE**

# DECLARATION

WE DO SOLEMNLY AFFIRM THAT THIS PRESENTED WORK AS TITLED "**A DEEP DIVE INTO MACHINE LEARNING AND ITS NOVEL APROACH TO WEATHER FORECASTING"** PROVIDED BY US TO THE UNIVERSITY OF CALICUT IN CONSIDERATION OF THE DEGREE OF BACHELOR OF SCIENCE IN PHYSICS, IS OUR ORIGINAL WORK, CONDUCTED IN THE PREMISES OF SREE GOKULAM ARTS AND SCIENCE COLLEGE, CALICUT. THE PRESENTED DISSERTATION HAVE NOT BEEN ISSUED TO ANY OTHER ISTITUITION OR UNIVERSITY FOR THE AWARD OF ANY DEGREE. WHILE MAINTAINING EDUCATIONAL INTEGRITY AS PRESENTING THE WORK, ANY SOURCES OF INFORMATION USED HAVE BEEN DULY AKNOWLEDGED AND REFERENCED.

# ACKNOWLEDGEMENTS

# *ABSTRACT*

*THE PRIMARY GOAL OF THIS PROJECT IS TO SET UP A WEATHER PREDICTION ALGORITHM USING MACHINE LEARNING TECHNIQUES, AIMING TO IMPROVE ITS EFFICIENCY AND PRECISION. BY LEVERAGING THE PROVIDED DATA, THE OBJECTIVE IS TO GAIN FURTHER UNDERSTANDING OF MACHINE LEARNING AND ITS APPLICATION IN WEATHER PREDICTION, ULTIMATELY ENHANCING THE MODEL'S ACCURACY TO PRODUCE MORE PRECISE FORECASTS AS THE CONCLUSIVE OUTPUT.*

# DATA-DRIVEN WEATHER FORECAST: OPTIMIZING WEATHER PREDICTIONS WITH MACHINE LEARNING

# Index

# Introduction to Machine Learning

"Machine intelligence is the last invention that humanity will ever need to take" As ever so harmoniously asserted by Nick Bostrom, puts into perspective the revolutionary force of machine learning that drives the technology of the present and the future, for posterity. Machine learning is a subject of AI, that uses statistical methods to enable a computer system to make certain decisions, to execute an operation or a task. In simple terms, machine learning provides computers the ability to learn without explicitly being programmed. Machine learning algorithms are trained to find relationships and patterns in the data. These programs are designed in such a manner that it can learn and enhance from the amount of data provided. The term machine learning was coined in 1959 by Arthur Samuel, an IBM employee and pioneer in the field of computer gaming and artificial intelligence. machine learning has come a long way since its humble beginnings in the early 1940's,noteably the neural network with electric circuit developed by Warren McCulloch and Walter Pitts in 1943.This model demonstrated that it was possible for two computers to exchange information without any human interaction. A most profound discovery as it set the foundation stone toward **machine learning development.**

The true purpose of machine learning is to derive meaning from the data given effectively, and to even open doors to predict the outcomes. This is done so by providing or feeding a historical data of the occurrence and then machine learning algorithms provide a model based on the example termed as training data. From there onwards, one uses a programming model to supply the data and let the computer model train itself to find patterns, and to even make predictions. Some data is held out from the training data to be used as evaluation data, which tests how accurate the machine learning data is when it is shown new data. Over time, programmers can use this to improve and rectify errors with little to no effort. This is why machine learning can be quite effective to enhance the accuracy of various prediction models that we use on a daily basis.

Any computer program that shows characteristics, such as self-improvement, learning through inference, or even basic human tasks, such as image recognition and language processing, is considered to be a form of AI. The field of artificial intelligence includes within it the sub-fields of machine learning and deep learning. Deep Learning is a more specialized version of machine learning that utilizes more complex methods solve strenuous tasks. The discerning factor of both subfields, is simply that machine learning is probabilistic (output can be explained, thereby ruling out the black box nature of AI), as for deep learning focuses more on deterministic approach.

The process of self-learning by collecting new data on the problem has allowed machine learning algorithms to pave the way for a brighter future.

## The Classifications of Machine Learning

Machine learning is classified into mainly 4 types based on its approach towards the data provided and the nature of its output.

**1. Supervised machine learning**

In supervised machine learning, algorithms are trained on *labelled* data sets that include tags describing each piece of data. In other words, the algorithms are fed data that includes an "answer key" describing how the data should be interpreted. For example, an algorithm may be fed images of flowers that include tags for each flower type so that it will be able to identify the flower better again when fed a new photograph.

Supervised machine learning is often used to create machine learning models used for prediction and classification purposes.

**2. Unsupervised machine learning**

Unsupervised machine learning uses *unlabelled* data sets to train algorithms. In this process, the algorithm is fed data that doesn't include tags, which requires it to uncover patterns on its own without any outside guidance. For instance, an algorithm may be fed a large amount of unlabelled user data culled from a social media site in order to identify behavioural trends on the platform.

Unsupervised machine learning is often used by researchers and data scientists to identify patterns within large, unlabelled data sets quickly and efficiently.

**3. Semi-supervised machine learning**

Semi-supervised machine learning uses both unlabelled and labelled data sets to train algorithms. Generally, during semi-supervised machine learning, algorithms are first fed a small amount of labelled data to help direct their development and then fed much larger quantities of unlabelled data to complete the model. For example, an algorithm may be fed a smaller quantity of labelled speech data and then trained on a much larger set of unlabelled speech data in order to create a machine learning model capable of speech recognition.

Semi-supervised machine learning is often employed to train algorithms for classification and prediction purposes in the event that large volumes of labelled data is unavailable.

### 4. Reinforcement learning

Reinforcement learning uses trial and error to train algorithms and create models. During the training process, algorithms operate in specific environments and then are provided with feedback following each outcome. Much like how a child learns, the algorithm slowly begins to acquire an understanding of its environment and begins to optimize actions to achieve particular outcomes. For instance, an algorithm may be optimized by playing successive games of chess, which allow it to learn from its past success and failures playing each game.

Reinforcement learning is often used to create algorithms that must effectively make sequences of decisions or actions to achieve their aims, such as playing a game or summarizing an entire text.

## Applications Of Machine Learning

Over several decades, machine learning has blossomed into myriad fields in science & technology, further paving a path toward a brighter future to come. The importance of machine learning at its core lies in its ability to learn the provided data, and to make predictions to unveil patterns, that was once not possible through rudimentary methods. From powering personalized recommendations on streaming platforms to enabling breakthroughs in predicting weather conditions, the importance of machine learning is vast and diverse.

1. **Virtual Personal Assistance:** This feature helps us in many ways, such as searching content using voice instruction, calling a number using voice, searching contact in your mobile, playing music, opening an email, Scheduling an appointment, etc. Google Assistant, Alexa, Cortana, Siri, etc., are a few common applications of machine learning. These virtual personal assistants record our voice instructions, send them over to the server on a cloud, decode it using ML algorithms and act accordingly.

2. **Healthcare and biomedicine:** Machine Learning is widely used in the healthcare industry. It helps healthcare researchers to analyse data points and suggest outcomes. Natural language processing helped to give accurate insights for better results of patients. Further, machine learning has improved the treatment methods by analysing external data on patients' conditions in terms of X-ray, Ultrasound, CT-scan, etc. NLP, medical imaging, and genetic information are key areas of machine learning that improve the diagnosis, detection, and prediction system in the healthcare sector.
3. **Scientific research and discovery:** Machine learning allows scientists to research, and analyse a plethora of data that were previously deemed inaccessible. It delivers the most efficient yet sought out data in data-driven scientific fields. This has helped scientists combine myriad of features into a condensed versions for them to summarize. We use machine learning to discover and understand a variety of natural phenomenon by applying machine learning algorithms to data college from various sources.
4. **Decision support**: Organizations also use machine learning to help them make better decisions. Experts of various fields noted that a decision support system (DSS) can also help cut costs and enhance performance by ensuring workers make the best decisions. To support decision-making, ML algorithms are trained on historical and other relevant data sets, enabling them to then analyse new information and run through multiple possible scenarios at a scale and speed impossible for humans to match. The algorithms then offer up recommendations on the best course of action to take.
5. **Machine learning in cybersecurity:** ML can help recognize patterns and predict threats in massive data sets, all at machine speed. By automating the analysis, cyber teams can rapidly detect threats and isolate situations that need deeper human analysis. It detects threat by constantly monitoring the behaviour of the network for anomalies. Machine learning engines process massive amounts of data in near real time to discover critical incidents. These techniques allow for the detection of insider threats, unknown malware, and policy violations.

# Machine Learning Models

For several years, predicting atmospheric conditions were only made possible using physics-based models. However, this model has its own challenges mainly the complexity, time consumption and computing intensity etc that ultimately yielded uncertainties in the final outputs. With the advancement of machine learning, it is now possible predict weather conditions with astounding precision and speed. This is made possible by identifying several patterns in historical data (such as storms, rainfall, temperature) in high complex dynamic systems with remarkable accuracy. Feeding models with data from various sources such as weather stations, satellite imagery, environmental sensors etc allows the model to understand the relationship between different scenarios and ultimately yield better results. The salient feature of incorporating machine learning into weather forecasting is certainly the ability to integrate diverse sources of data rather than simply relying of explicit equations. Deep learning models can be trained in such a manner, that it can maximize the computing power to produce and identify visual patterns from the given data set.

## Regression Models

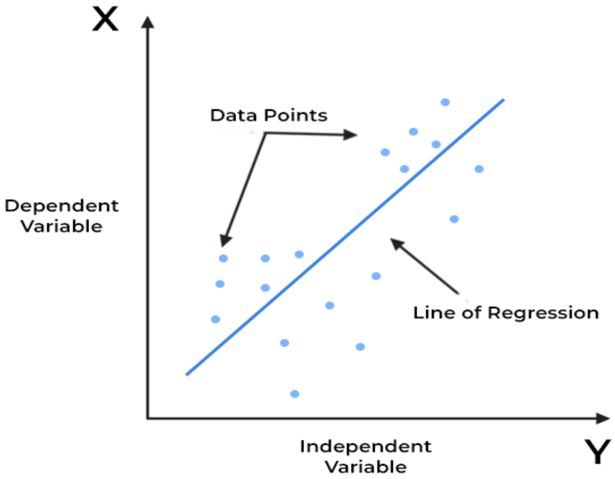
A regression model is a model that provides a function to describe the relationship between one or more independent variable and a dependent variable. It is also a statistical approach of sorting out which of the provided parameters have an impact. A regression model consists of a response variable and a predictor variable. The dependent variable is the response variable whereas the independent variable is the predictor variable. Ultimately, this relation is represented as a mathematical equation, that provides us with a straight line describing how a response variable Y changes with respect to predictor variable X. With the help of regression models, researchers can analyse the patterns and variables of a complex system such as the Atmosphere, to predict weather using merely the historical data provided. Some of the regression models are:

1. Linear regression model
2. Logistic regression
3. Neural networks
4. Clustering
5. Random forest regression

In this project, we use a type of linear regression called ridge regression.

## Linear Regression Model

Linear regression is a statistical model that predicts the value of unknown data using related and known data. It mathematically models the unknown variable and the known variable as a linear equation. In linear regression, the variable being predicted is called the dependent variable, and the variable used to predict it is called the independent variable.



## Ridge Regression Model

Ridge regression model is a statistical regularization technique that corrects for overfitting on training data in machine learning models. This is useful when we develop a model with large number of parameters like in weather, in this case.

Ridge regression helps to make predictions, whilst noting the connections between any two similar factors. Henceforth, no single factor would have too much influence over the other, and prediction still remains more reliable even when some factors have similar parameters.

Since weather conditions consist of a myriad of variables, we use regression model instead of simple linear regression model since it handles multicollinearity (two or more variables being highly related) more effectively by introducing a balance, between fitting the model well and keeping the coefficients small. This prevents the model to not rely heavily on 1 predictor variable. This model can also help further improve accuracy and resilience to model uncertainties.

---

It is a specialized technique that is used to analyse multiple regression data which is multi-collinear in nature. Ridge regression is a fundamental regulation technique, and this method could be compared to normal regression methods where the regularization, i.e., the way the model coefficients are determined, will be different.

Advantages of Ridge regression:

- It protects the model from overfitting
- It does not need unbiased estimators
- There is only enough bias to make the estimates reasonably reliable approximations to the true population values
- It is very effective when it comes to improving the least-squares estimate in situations where there is multi collinearity
- Model complexity reduction

Disadvantages:

- It includes all the predictors in the final model
- It is not capable of preforming feature selection
- It shrinks coefficients towards zero
- It trades variance for bias

# Rolling Averages Computation

Rolling average is a calculation that let us analyse data points by creating a set of series of averages based on different subsets of data set which we provide. Rolling average or moving average calculates the average of data over some periods.

### Calculating Rolling Averages using Pandas

There are various ways for calculating it in Pandas Data frame.

Some of them are as follows:

1. Simple Moving Average (SMA)
2. Exponential Moving Average (EMA)
3. Cumulative Moving Average (CMA)

**1. Simple Moving Average (SMA)**

Syntax:

To calculate SMA in Pandas data frame we will be using Pandas data frame.rolling() action that helps us to make calculations on a rolling window.

And as we continue on the rolling window we will use .mean() function to calculate the mean of each window.

An SMA tells us the unweighted mean of the previous K data points. The more the value of K, the smoother is the curve, but increasing k decreases accuracy. If the data points are $P_1, P_2, \ldots, P_n$ then we calculate the SMA

$$SMA_k = \frac{P_{n-k+1} + P_{n-k+2} + \ldots + P_n}{k}$$

$$= \frac{1}{k} \sum_{i=n-k+1}^{n} P_i$$

As we discussed before we use .rolling() method in SMA. And this method provides rolling windows over the data, and we can use the mean function over these windows to calculate moving averages. The size of the window is passed as a parameter in the function .rolling(window).

**2. Exponential Moving Average (EMA)**

Exponential moving average (EMA) tells us the weighted mean of the previous k data points. EMA places a greater weight and significance on the most read data points. The formula to calculate EMA at the time period t is:

$$EMA_t = \begin{cases} x_0 & t = 0 \\ \alpha x_t + (1 - \alpha)\, EMA_{t-1} & t > 0 \end{cases}$$

Where $x_t$ is the value of observation at time t & $\alpha$ is the smoothing factor.

In Python, EMA is calculated using .ewm() method. We can pass span or window as a parameter to window .ewm(span =) method.

**3. Cumulative Moving Average (CMA)**

To calculate CMA in Pandas Data frame we will be using data frame.expanding() function.

# CMA is the mean of all the previous values up to the current value. CMA of datapoints $x_1, x_2, \ldots$ at time t can be calculated as

$$CMA_t = \frac{x_1 + x_2 + x_3 + \ldots + x_t}{t}$$

While calculating CMA we don’t have any fixed size of the window. The size of the window keeps on increasing as time passes.

# Time Series Cross-Validation (Back-testing)

In the realm of predictive modelling, ensuring the reliability and accuracy of our models is paramount, particularly when working with time series data. Time series data is characterized by its sequential structure, where observations are recorded over successive time intervals. This sequential nature introduces dependencies among observations, necessitating specialized validation techniques to assess predictive models effectively.

Time series cross-validation is a crucial technique for evaluating how well our predictive models perform on time-ordered data. Unlike standard cross-validation, which assumes each data point is independent, time series cross-validation maintains the order of observations, reflecting real-world scenarios more accurately.

Here we use a daily weather temperature dataset where the temporal order of the data matters. Thus, using any of the conventional methods would not be ideal since the model would be trained on data from the future and get tested on the data from the past as well. Hence, we can use time series cross-validation to split the dataset where the order in terms of time is preserved.

# Data Processing and Code Implementation

## Feature Engineering

Feature engineering is the process of transforming raw data into relevant information for use by machine learning models. In other words, feature engineering is the process of creating predictive model features. A feature—also called a dimension—is an input variable used to generate model predictions. Because model performance largely rests on the quality of data used during training, feature engineering is a crucial preprocessing technique that requires selecting the most relevant aspects of raw training data for both the predictive task and model type under consideration.

## Data Collection & Preprocessing

The weather forecast machine learning prediction model relies on data sourced from the National Oceanic and Atmospheric Administration (NOAA), focusing on weather data specific to New York.

Since this project involves predicting weather, having a plethora of missing datapoints will lead to the degradation of the model's accuracy. Thus, to ensure the reliability of the model, it is necessary to remove datapoints that are missing and inconsistent throughout the dataset. This delineated approach to data collection and preprocessing forms the backbone of the weather forecast model, ensuring robustness and reliability of the model.

## Tools and packages used:

### Pandas:

It is an open-source library for data analysis in Python. It was developed by Wes McKinney in 2008. Over the years, it has become the standard library for data analysis using Python. It provides high-level data structures and functions designed to make working with structured or tabular data easy and intuitive

- Data Frame: It is a two-dimensional labelled data structure with columns of potentially different types. It is analogous to a spreadsheet or SQL table, and it is the primary data structure used in Pandas for data manipulation.
- Series: A Series is a one-dimensional labelled array capable of holding any data type. It is similar to a Python list or NumPy array but with additional features like labelled indexing.
- Data manipulation: Pandas offers a wide range of functions for data manipulation, including indexing, filtering, slicing, merging, joining, reshaping, grouping, and aggregating data.
- Data cleaning: Pandas provides tools for handling missing data, removing duplicates, converting data types, and other data cleaning tasks.
- Input/output tools: Pandas supports reading and writing data from/to various file formats, including CSV, Excel, SQL databases, JSON, HTML, and more.
- Time series functionality: Pandas includes robust support for working with time series data, including date/time indexing, resampling, time zone handling, and date range generation.

## Scikit-learn:

scikit-learn is an open source and user-friendly library that enables practitioners to efficiently implement machine learning algorithms and build predictive models for a wide range of tasks, from classification and regression to clustering and dimensionality reduction. It provides us with an array of various tools for pre-processing, cross-validation, and visualization algorithms, etc.

We use the built-in functions provided by this library to implement and train the weather machine learning model for this project, namely the Ridge function.

## Matplotlib:

Matplotlib is a popular open-source plotting library in Python used for creating high-quality visualizations and graphs. With Matplotlib, users can generate an array of plot types including line plots, scatter plots, bar charts, histograms, pie charts, contour plots, and even 3D plots.

## Jupyter Notebook:

Jupyter Notebook is a free, open-source web application that allows users to create and share documents that include live code, equations, and other multimedia resources. It is used by programmers, data scientists, and students to document and demonstrate coding workflows or simply experiment with code.

This application is used for writing, implementing, and experimenting the machine learning model for the project. The data visualisation and prototyping were done through this integrated development environment.

# **Source Code Walkthrough:**

First, we import the necessary python packages. Here we use three packages. Pandas, Scikit-learn and Matplotlib.

```
import pandas as pd
from sklearn.linear_model import Ridge
import matplotlib.pyplot as plt
```

Then, the dataset for the prediction model is imported to the program. We, use a dataset that was provided by the NOAA (National Oceanic and Atmospheric Administration). It is a United States scientific agency within the United States Department of Commerce that focuses on monitoring and predicting changes in the Earth's environment, including the atmosphere, oceans, and coasts. NOAA provides a wide range of services related to weather forecasting, climate monitoring, oceanography, marine life conservation, and environmental stewardship

```
weather = pd.read_csv("data/weather.csv", index_col="DATE")
```

It is important for machine learning algorithms to have a reliable dataset since the entire prediction model will be trained on this dataset provided. For this we transform the data and remove inconsistent and null data points.

```
null_pct = weather.apply(pd.isnull).sum()/weather.shape[0]
valid_columns = weather.columns[null_pct < .05]
weather = weather[valid_columns].copy()
weather.columns = weather.columns.str.lower()
weather = weather.ffill()
weather.apply(pd.isnull).sum()
weather.apply(lambda x: (x == 9999).sum())
weather.index = pd.to_datetime(weather.index)
weather.index.year.value_counts().sort_index()
```

In order to predict the model, first we create a new column named target in the dataset. First, we provide the placeholder data for the target as the next days data. For that the DataFrame.shift() method in the pandas library is used.

```
weather["target"] = weather.shift(-1)["tmax"]
weather = weather.ffill()
weather["target"]
```

Since the features that are used for the prediction are highly correlated, i.e., multicollinear it is important to take this into consideration. Thus, using a regression model such as Ridge Regression model which helps to shrink the coefficients to account for multicollinearity. Here, the alpha hyperparameter is controls this shrinkage.

```
ridge = Ridge(alpha=0.1)
```

we do not to use the target column and the NAME and STATION columns as features. Thus, those are removed from the features.

```
features = dataset.drop(columns=["target", "NAME", "STATION"]).columns
```

Now, we use the Scikit-learn library to implement and train the model using the Ridge Regression model. For this, the provided dataset is split into training dataset and testing dataset. Since the data used here is a time series data, i.e., the temporal order of the data is important. Thus, instead of randomly splitting the dataset into two, we use a method called time series cross-validation where this order is preserved.

```
rr = Ridge(alpha=.1)

predictors = weather.columns[~weather.columns.isin(["target", "name",
"station"])]

def backtest(weather, model, predictors, start=3650, step=90):
    all_predictions = []

    for i in range(start, weather.shape[0], step):
        train = weather.iloc[:i,:]
        test = weather.iloc[i:(i+step),:]
        model.fit(train[predictors], train["target"])

        preds = model.predict(test[predictors])
        preds = pd.Series(preds, index=test.index)
        combined = pd.concat([test["target"], preds], axis=1)
        combined.columns = ["actual", "prediction"]
        combined["diff"] = (combined["prediction"] - combined["actual"]).abs()

        all_predictions.append(combined)
    return pd.concat(all_predictions)


predictions = backtest(weather, rr, predictors)
```

Now, the mean_absolute_error function is used to measure the accuracy of the model.

```
mean_absolute_error(predictions["actual"], predictions["prediction"])

predictions.sort_values("diff", ascending=False)
pd.Series(rr.coef_, index=predictors)
```

Further steps were resorted to improve the accuracy of the model such as using the method of rolling averages.

```
def pct_diff(old, new):
    return (new - old) / old

def compute_rolling(weather, horizon, col):
    label = f"rolling_{horizon}_{col}"
    weather[label] = weather[col].rolling(horizon).mean()
    weather[f"{label}_pct"] = pct_diff(weather[label], weather[col])
    return weather

rolling_horizons = [3, 14]
for horizon in rolling_horizons:
    for col in ["tmax", "tmin", "prcp"]:
        weather = compute_rolling(weather, horizon, col)

def expand_mean(df):
    return df.expanding(1).mean()

for col in ["tmax", "tmin", "prcp"]:
    weather[f"month_avg_{col}"] = weather[col].groupby(weather.index.month,
group_keys=False).apply(expand_mean)
```

```
    weather[f"day_avg_{col}"] =
weather[col].groupby(weather.index.day_of_year,
group_keys=False).apply(expand_mean)

weather = weather.iloc[14:,:]
weather = weather.fillna(0)
predictors = weather.columns[~weather.columns.isin(["target", "name",
"station"])]
predictions = backtest(weather, rr, predictors)
mean_absolute_error(predictions["actual"], predictions["prediction"])
mean_squared_error(predictions["actual"], predictions["prediction"])
predictions.sort_values("diff", ascending=False)
weather.loc["1990-03-07": "1990-03-17"]
(predictions["diff"].round().value_counts().sort_index() /
predictions.shape[0]).plot()
weather = weather.iloc[14:,:]
weather = weather.fillna(0)
predictors = weather.columns[~weather.columns.isin(["target", "name",
"station"])]
predictions = backtest(weather, rr, predictors)
mean_absolute_error(predictions["actual"], predictions["prediction"])
mean_squared_error(predictions["actual"], predictions["prediction"])
predictions.sort_values("diff", ascending=False)
weather.loc["1990-03-07": "1990-03-17"]
(predictions["diff"].round().value_counts().sort_index() /
predictions.shape[0]).plot()
```

# REFERENCES

1.

# CONCLUSION