Summer Fellowship Report

On

Python for Machine learning

Submitted by

T Harini

Vellore Institute of Technology, Chennai

Under the guidance of

Prof. Kannan M. Moudgalya

Chemical Engineering Department
IIT Bombay

Mentors from IIT Bombay

Ms. Nirmala Venkat

Mentors from VIT Chennai

Dr. Subbulakshmi T
Dr. Bhargavi Rentachintala

September 19, 2025

Acknowledgment

I would like to extend my gratitude to my mentor at IIT Bombay including Ms. Nirmala for her constant support throughout the internship. Without her support and suggestions at each step of the script and code development, this tutorial series would not have taken shape.

I also appreciate, and am forever grateful to the faculties at VIT Chennai, including Dr. Subbulakshmi T and Dr. Bhargavi Rentachintala for their technical expertise during the code development phase, and their valuable suggestions for the content.

Last, but not the least, I thank my fellow intern Anvita Thadavoose Manjummel for her cooperation.

Contents

Chapter 1

Introduction

The Spoken Tutorial project, developed under the FOSSEE initiative at IIT Bombay, is an online platform offering video tutorials on Free and Open Source Software (FOSS). The tutorials are structured to enable learners to study independently, at their own pace, using just a computer and internet connection, with minimal prior knowledge required.

Practical learning is emphasized throughout the tutorials, with most of the time dedicated to demonstrations of the concepts being taught. All necessary resources, including source files used in the tutorials, are provided to facilitate hands-on practice. To help learners consolidate their understanding, each tutorial includes practice problems that apply the concepts covered. Additionally, learners can engage with domain experts through tutorial-specific forums, where questions are answered and further explanations are provided.

The Spoken Tutorial project also offers certification opportunities, allowing learners to validate their skills through online assessments after completing the tutorials. This structure ensures a comprehensive learning experience, combining theory, practical exercises, and expert guidance to strengthen understanding of the subject matter.

Chapter 2

Tutorials on Python for Machine Learning

The Python for Machine Learning tutorials provided through the Spoken Tutorial project are designed to build a strong foundation in both programming and machine learning concepts using practical, hands-on exercises. The tutorials start with setting up the Python environment for machine learning, guiding learners through the installation of necessary libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn, and familiarizing them with tools like Jupyter Notebook.

Once the environment is ready, learners explore a range of machine learning algorithms. The tutorials cover classification techniques such as K-Nearest Neighbor (KNN) Classification, Logistic Regression for both binary and multiclass problems, Decision Trees, Support Vector Machines (SVM), and ensemble methods like Random Forest. Regression techniques are also included, with K-Nearest Neighbor Regression and Linear Regression demonstrated for predicting continuous outcomes.

Learners are also introduced to Artificial Neural Networks (ANNs), providing a basic understanding of deep learning concepts, and K-Means Clustering, which illustrates how similar data points can be grouped. Each algorithm is taught through step-by-step demonstrations, with datasets and source files provided to allow hands-on experimentation. Practice problems at the end of each tutorial help reinforce the concepts, while tutorial-specific forums enable learners to ask questions and receive guidance from domain experts.

Overall, the tutorials offer a comprehensive approach to learning Python for machine learning, combining theoretical knowledge with practical implementation, preparing learners to apply these techniques to real-world problems effectively.

2.1 Setup Python Environment for Machine Learning

This tutorial focuses on preparing the Python environment required for machine learning. It begins with an introduction to machine learning and the advantages of using Python, followed by an overview of essential libraries. Learners are guided through installing Miniconda on Ubuntu, creating and activating a dedicated conda environment, and installing all required libraries from a provided package file. The tutorial also covers setting up Jupyter Notebook, adding conda kernels, and introduces the basics of using Jupyter Notebook for implementing machine learning programs.

The tutorial covers the following:

- About Machine Learning.
- Why we use Python for Machine Learning.
- About Python libraries.
- Installation of Miniconda in Ubuntu OS.
- Creating a conda environment for Machine Learning.
- Activating conda environment for Machine Learning.
- Download MLpackage.txt file.
- Install all the libraries available in the txt file in Ubuntu OS.
- Installing Jupyter Notebook in the Machine Learning environment.
- Installing conda kernels in Machine Learning environment.
- About Jupyter Notebook and its basics.

2.2 K Nearest Neighbor Classification

This tutorial introduces the K-Nearest Neighbor (KNN) algorithm for classification tasks. It demonstrates how data points are assigned to the class most common among their nearest neighbors, covering distance metrics, training, and predicting labels for new data.

This tutorial covers the following:

- Introduction to Nearest Neighbors and K Nearest Neighbor
- Introduction to KNN classification
- Explanation about Iris dataset

- KNN working example using one of the iris features
- Importing the necessary libraries
- Loading the Iris dataset
- Basic Data Exploration and Analysis
- Train and Test Split of dataset
- Choosing the K value using elbow method
- KNN classification model building
- Model prediction and outcome
- Evaluation metrics using classification report

2.3 K Nearest Neighbor Regression

The K-Nearest Neighbor Regression tutorial explains how KNN can be adapted to predict continuous values. It covers finding the average of the nearest neighbors to make predictions, implementing the algorithm in Python, and analyzing the results using performance metrics such as Mean Squared Error.

This tutorial covers the following:

- Introduction to K Nearest Neighbor Regression
- Various distance metrics used in KNN
- Importing the necessary libraries
- Loading the iris dataset
- Standard scaling of the dataset
- Train and Test Split of dataset
- Choosing the K value using elbow method
- KNN regression model building
- Model prediction and outcome
- Evaluation using MSE and Adjusted R squared score

2.4 Linear Regression

This tutorial teaches the concept of linear regression for predicting continuous outcomes. Learners explore the relationship between independent and dependent variables, fit a linear model using Python, visualize regression lines, and evaluate model performance using evaluation metrics

This tutorial covers the following:

- About Linear Regression
- About Simple Linear Regression
- About Multiple Linear Regression
- About Evaluation Metrics
- Splitting the data into training and testing sets
- Implementing Simple Linear Regression model from scikit-learn
- Importing required Libraries
- Loading the dataset
- Evaluating the model's accuracy
- Implementing Multiple Linear Regression model from scikit-learn
- Evaluating the model's accuracy

2.5 Logistic Regression Binary Classification

The tutorial focuses on logistic regression for binary classification problems. It explains how probabilities are modeled using the sigmoid function, fitting the model in Python, and evaluating its performance.

This tutorial covers the following:

- Introduction to Logistic Regression
- Introduction to Binary classification
- Introduction to Multi class classification
- About Purchase prediction
- Implementing Binary classification

- Model Instantiation of Binary Classification and Model training
- Prediction for Train Data - Verification for Binary Classification
- Predictions for Test Data for Binary Classification
- Calculate the ROC-AUC score on the training data
- Calculate the cross entropy loss for the training data

2.6 Logistic Regression MultiClass Classification

This tutorial extends logistic regression to handle multiple classes. Using Python libraries, learners train and test models on multiclass datasets.

This tutorial covers the following:

- Implementing Multiclass classification
- Model Instantiation of Multiclass Classification and Model training
- Visualize this correlation using a heatmap
- Split the data into training and testing sets
- Build a multiclass classification model
- Prediction for Train Data - Verification for Multiclass Classification
- Predictions for Test Data for Multiclass Classification
- Compare the predicted with the actual test class
- Visualize the confusion matrix of the model

2.7 Decision Tree

This tutorial provides a step-by-step introduction to building and evaluating a Decision Tree classifier. Learners also explore how tree visualization provides insights into the decision-making process, making it easier to understand how the model arrives at its outcomes.

This tutorial covers the following:

- Introduction to Decision Tree
- Describing the dataset

- Importing required Libraries
- Loading the dataset
- Encoding Categorical Features
- Splitting the dataset into Training and Testing sets
- Training Decision Tree Classifier
- Evaluating the model's accuracy
- Plotting Confusion matrix
- Visualizing Decision Tree

2.8 Artificial Neural Networks

This tutorial introduces Artificial Neural Networks (ANNs) and their structure through the Multi-Layer Perceptron (MLP). It explains the architecture of neurons and layers, and applies the concepts to the Breast Cancer dataset. The tutorial demonstrates training an MLP classifier, making predictions, and evaluating performance using common metrics.

This tutorial covers the following:

- Introduction to Artificial Neural Networks
- Introduction to Multi-Layer Perceptron
- About ANN Architecture
- Explanation of Neuron Structure
- Importing necessary libraries
- Loading Breast Cancer dataset
- Basic Data Exploration and Analysis
- Train and Test split of dataset
- MLP Classification model building
- Model prediction and outcome
- Evaluation of model's performance

2.9 Support Vector Machine

This tutorial introduces Support Vector Machines (SVMs), covering both Linear and Non-Linear (RBF) approaches. Using the California Housing dataset, it explains preprocessing steps such as encoding and train-test splitting. The tutorial demonstrates building classification models with Linear SVM, followed by Non-Linear SVM, and compares their predictions and evaluation results.

This tutorial covers the following:

- About Support Vector Machine
- Introduction to Linear SVM
- Introduction to Non-Linear SVM
- Explanation of the California Housing dataset
- Importing necessary libraries
- Loading the dataset
- Label Encoding
- Train and Test Split of dataset
- Linear SVM classification model building
- Model prediction and outcome
- Evaluation for Linear SVM classification
- Non-Linear (RBF) SVM classification model building
- Model prediction and outcome
- Evaluation for Non-Linear (RBF) SVM classification

2.10 K Means Clustering

This tutorial introduces K-Means Clustering and explains its working for grouping similar data points. It covers evaluation using the Silhouette Score and uses a customers dataset to demonstrate the process. It then shows how to instantiate and fit the K-Means model, assign cluster labels, and visualize the resulting clusters to interpret patterns within the data.

This tutorial covers the following:

- Introduction to K-means Clustering

- Working of K-means Clustering
- Description about Silhouette Score
- Description about the customers dataset
- Importing required Libraries
- Loading the dataset
- Data Exploration
- Finding optimal number of clusters
- Instantiating K-means Clustering model
- Clustering the data
- Visualizing the Clusters for the Data

2.11 Random Forest

This tutorial introduces Ensemble Learning and focuses on the Random Forest algorithm. It explains how Random Forest combines multiple decision trees to improve predictive performance.

This tutorial covers the following:

- Introduction to Ensemble Learning
- Introduction to Random Forest
- Importing Libraries
- Loading the dataset
- Data Preprocessing
- Train and Test Split
- Model Instantiation of Random Forest and Model training
- Prediction for Train Data - Verification for Random Forest
- Predictions for Test Data for Random Forest
- MSE and Adjusted R square score for Random Forest

Chapter 3

Code and Script Writing

For the tutorials described in the previous chapter, I prepared the source code along with the corresponding scripts, including visual cues and narration, and recorded the tutorial videos. The scripts were carefully written in accordance with the Spoken Tutorial guidelines. Each segment of the code was explained in detail, and additional references were provided whenever prerequisite concepts could not be fully addressed within the tutorial.

The source code and the corresponding scripts can be found in the following links:

- Setup Python environment for Machine Learning
  Code: Sample.ipynb
  Script: Tutorial-1 Setup Python Environment for Machine Learning

- K Nearest Neighbor Classification
  Code: KNN classification.ipynb
  Script: Tutorial-2 KNN Classification Script

- K Nearest Neighbor Regression
  Code: KNNregression.ipynb
  Script: Tutorial-3 KNNregression

- Linear Regression
  Code: LinearRegression.ipynb
  Script: Tutorial-4 Linear Regression

- Logistic Regression Binary Classification
  Code: LR_Binary.ipynb
  Script: Tutorial-5 LR_Binary

- Logistic Regression MultiClass Classification
  Code: LR_Multiclass.ipynb
  Script: Tutorial-6 LR_Multiclass

- Decision Tree
  Code: DecisionTree.ipynb
  Script: Tutorial-7 Decision Tree

- Artificial Neural Networks
  Code: Artificial Neural Networks.ipynb
  Script: Tutorial-8 Artificial Neural Networks Script

- Support Vector Machine
  Code: SVM.ipynb
  Script: Tutorial-9 SVM

- K Means Clustering
  Code: K means clustering.ipynb
  Script: Tutorial-10 K-means Clustering Script

- Random Forest
  Code: RandomForest.ipynb
  Script: Tutorial-11 Random Forest

Chapter 4

Creating slides

In addition to developing the source code and scripts with narration and visual cues, I also prepared presentation slides to accompany each tutorial, including the introduction, system requirements, prerequisites, and key concepts to make the learning process clearer. All materials were created in line with the Spoken Tutorial guidelines, ensuring that each topic is explained thoroughly, with references provided for prerequisite knowledge when needed.

The slides for the tutorials can be found at the following links:

- Setup Python Environment for Machine Learning
- K Nearest Neighbor Classification
- K Nearest Neighbor Regression
- Linear Regression
- Logistic Regression Binary Classification
- Logistic Regression MultiClass Classification
- Decision Tree
- Artificial Neural Networks
- Support Vector Machine
- K Means Clustering
- Random Forest

Chapter 5

Phases of my Spoken Tutorial

5.1 Outline

The outline defines the structure of the Spoken Tutorial, listing all the topics to be covered in a sequential manner. It serves as a roadmap, ensuring that the content maintains a logical flow, beginning with basic concepts and gradually progressing to advanced topics.

5.2 Script

The script contains detailed, step-by-step instructions, explanations, and examples that form the core of the tutorial. It is written in a simple and precise manner, enabling learners, especially beginners, to follow the process without difficulty. Key points for demonstration are also included to maintain clarity. The script follows the Spoken Tutorial guidelines and is written in a clear and concise manner for easy understanding.

5.3 Slides

Slides provide supporting visual content for the tutorial. They include an introduction, prerequisites, system requirements, and key concepts, making the learning process more structured and engaging. The slides are created using LaTeX to ensure a neat layout and professional presentation. The color scheme and formatting remain consistent across all tutorials, in line with the Spoken Tutorial guidelines.

5.4 Novice Check

At this stage, the script and slides undergo a novice check to ensure that the material can be easily understood by learners with no prior experience in the subject. Feedback is used to simplify explanations, add missing details, and eliminate ambiguities. The Novice check for my tutorials were performed by Ms. Rashmi Patankar from the Spoken Tutorial team. She performed the check multiple times to ensure that the content is well explained and that the source code is bug-free.

5.5 Domain Check

The domain check is conducted by subject experts to verify technical accuracy and correctness of the explanations. The domain check was performed by Dr. Subbulakshmi T from VIT Chennai and Ms. Nirmala Venkat from IIT Bombay.

5.6 Admin Check

The admin check ensures compliance with Spoken Tutorial guidelines. This includes reviewing the format, structure, consistency of language, and slide presentation standards. Approval at this stage confirms that the material is ready for recording. The admin check was done by Ms. Nirmala from the Spoken Tutorial team

5.7 Recording

In the final phase, the tutorial is recorded using the approved script and slides. The recording combines screen demonstrations with clear narration, ensuring that learners can follow along smoothly. The recording is produced in high quality, adhering strictly to the approved script and maintaining focus on the relevant content without deviations.

Chapter 6

Conclusion

The tutorials on Python for Machine Learning provided a structured and practical approach to understanding essential algorithms and techniques. By preparing the source code, scripts, slides, and recorded videos, the concepts were explained in a way that supports both self-paced and applied learning. This process not only strengthened my knowledge of machine learning algorithms but also deepened my confidence in applying them to real-world datasets. Since the slides were prepared using LaTeX, I also gained valuable experience in working with LaTeX for professional documentation. Additionally, working with the team member improved my teamwork and coordination skills, making the overall experience technically enriching and personally rewarding.

I would again like to thank the Spoken Tutorial team and my mentors at VIT Chennai for providing me this opportunity

References

- https://spoken-tutorial.org/
- https://scikit-learn.org/stable/