Python and Data Manipulation Learning Summary

Getting Started with Python and Setting Up the Environment

Introduction to Python Basics

I started with a concise 1-hour crash course that introduced me to the essential programming constructs in Python. I learned how to declare variables and work with different data types like integers, floats, strings, lists, and dictionaries. Control structures such as `if-else` statements and loops helped me understand how to control the flow of a program. Functions were another key concept—I learned how to define them using `def`, pass arguments, and return values. Input/output operations using `input()` and `print()` were also covered, along with basic error handling through try-except blocks. This course gave me a solid starting point in Python and made me confident in writing simple programs and understanding the language's syntax.

Next, I went through the Python course on Kaggle, which was interactive and provided a certificate upon completion. I practiced using lists and dictionaries, wrote loops and conditionals, and worked with functions. I also explored boolean logic and how to use external libraries like `random` and `math`. Importantly, I learned to read and manipulate data from files, such as CSVs. This hands-on course deepened my understanding of Python and was particularly useful for data analysis. The certificate was a nice bonus!

To expand my understanding further, I watched a more detailed tutorial. This introduced me to object-oriented programming, where I learned about classes and objects. I also explored file input/output operations, and how to organize code using modules and packages. Exception handling and debugging techniques were also explained in depth. I came away with a much more complete understanding of Python. This course helped bridge the gap between writing basic scripts and building larger, real-world applications.

Google Colab

I explored Google Colab, and I found it as an excellent tool for developing and sharing data science and machine learning projects. It's especially convenient since everything is in the cloud. It integrates seamlessly with Google Drive and allows me to write and run Python code without any local setup. Some of its features that stood out to me the most includes code and text cells for combining documentation and execution, easy access to files stored in Google Drive and one-click installation of Python libraries

I also looked into how to set up Jupyter Notebook on my own machine. I learned how to use pip installer and how to launch the notebook via command prompt.

Diving Into Data Manipulation and Foundational Math

Mastering NumPy

I began learning NumPy, the foundational library for numerical computing in Python. I discovered how to create and manipulate arrays and vectors, including indexing, slicing, and reshaping them. I explored matrix operations such as dot products and broadcasting, and practiced using mathematical functions like `np.mean()`, `np.std()`, and `np.sum()`. I also learned how to generate random data using the `np.random` module. NumPy has allowed me to perform complex mathematical operations efficiently and is critical for handling large datasets.

Getting Comfortable with Pandas

Next, I explored Pandas. I learned about Series and DataFrames, and how powerful data structures they are for handling tabular data. I practiced reading and writing data in various formats including CSV and Excel. I also learned techniques for cleaning data (e.g., handling missing values and duplicates), data summarization, and combining datasets using merging and concatenation. Learning Pandas made it easy to clean, transform, and analyze large datasets quickly.

Creating Visualizations with Matplotlib

I then moved on to Matplotlib, Python's core data visualization library. I created different types of plots, such as line charts, bar charts, scatter plots, and histograms. I learned to customize these plots by adding labels, titles, legends, and changing colors and styles. I also experimented with subplots and saving the plots as image files.

Refreshing My Math Skills for Machine Learning

I reviewed a comprehensive GitHub resource that helped reinforce my math foundation for machine learning. Topics included: - **Linear Algebra**: I revisited vectors, matrices, and their operations, including eigenvalues and eigenvectors. - **Calculus**: I brushed up on derivatives, gradients, and partial derivatives which are fundamental for optimization. - **Probability and Statistics**: I refreshed my understanding of probability distributions, descriptive statistics like mean and standard deviation, conditional probability, and Bayes' Theorem.

Understanding the Types of Learning in ML

I explored the differences between supervised and unsupervised learning:

Aspect	Supervised Learning	Unsupervised Learning
What it is	Learning from labeled data	Learning from unlabeled data
Objective	Make predictions on new data	Discover hidden patterns or groupings
Examples	Classification, Regression	Clustering, Dimensionality Reduction

Aspect	Supervised Learning	Unsupervised Learning
Algorithms	Linear Regression, SVM, Decision Trees	K-Means, PCA, Hierarchical Clustering

Conclusion

Through these resources and exercises, I built a strong foundation in Python programming, data manipulation, and visualization. Tools like Google Colab and Jupyter Notebook have made development and experimentation easier. Understanding the basic math and concepts of machine learning has also equipped me with the knowledge needed to begin developing actual models. I’m now ready to move ahead and start building data-driven projects with confidence.