**Data Mining Project on Finding Trending Technologies**

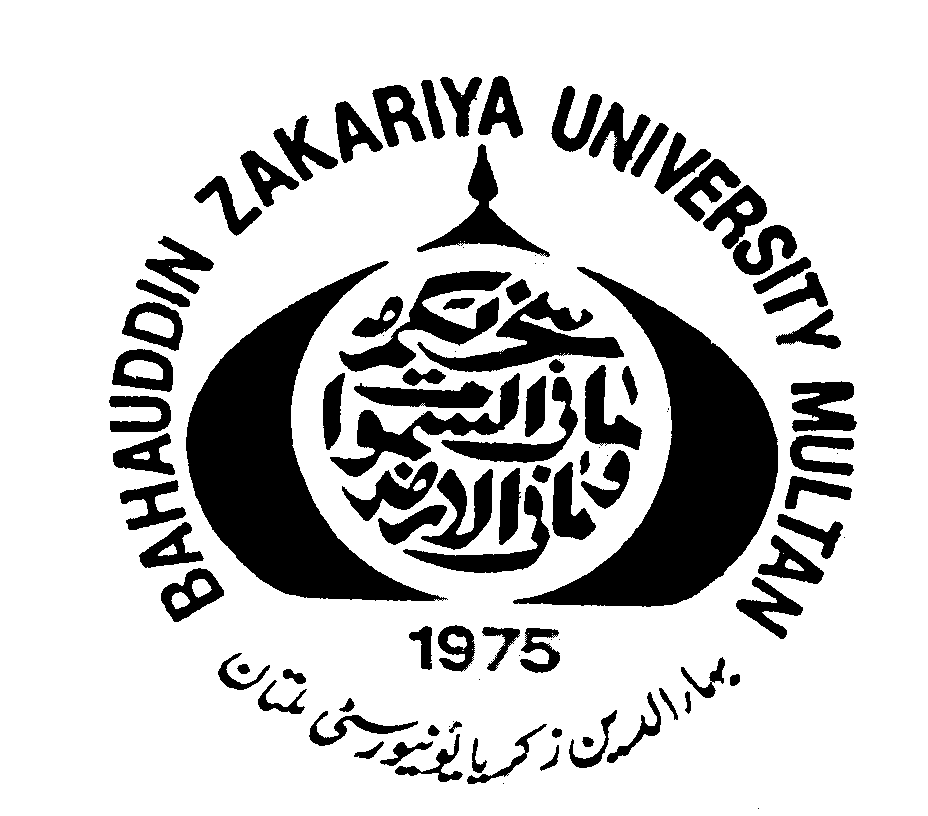
**Submitted By:**

**Muhammad Junaid**

**BSIT(M)17-17**

**Project Supervisor:**

**Dr. M. Ahsan Raza**

****

## Department Of Information Technology

**BAHAUDDIN ZAKARIYA UNIVERSITY MULTAN PAKISTAN**

|  |
| --- |
|  |

**FINAL APPROVAL**

This is to certify that we have read this report titled as **“Data Mining Project on Finding Trending Technologies”** submitted by **Muhammad Junaid, Roll No. BSIT-17-17** and it is our judgment that this report is of sufficient standard to warrant its acceptance by Bahauddin Zakariya University, Multan for the degree of **BS(Information Technology).**

# *Committee:*

1. **External Examiner \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
2. **Supervisor:**

**Dr. M. Ahsan Raza \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Lecturer

(Department of Information Technology BZU, Multan)

**3. Head of Department:**

**Dr. Maruf Pasha \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Department of Information Technology BZU, Multan)

**DEDICATION**

***To my Loving Parents***

***And***

***To my kind Teachers***

**PROJECT BRIEF**

|  |  |
| --- | --- |
| PROJECT NAME | Data Mining Project on Finding Trending Technologies |
| UNDERTAKEN BY | Muhammad Junaid |
| SUPERVISED BY | Dr. M. Ahsan Raza |
| STARTING DATE | Feb 01, 2021 |
| COMPLETION DATE | Aug 28, 2021 |
| COMPUTER USED | Intel(R) Core (TM) i5-6300U CPU @ 2.40GHz 2.50GHz Ram 8GB (DDR4 2133 MHz). |
| OPERATING SYSTEM | Windows 10 (x64) |
| SOURCE LANGUAGE(S) | Python, HTML, CSS, Bootstrap |
| TOOLS/PACKAGES | Jupyterlab Notebook, Pandas, Scikit Learn,  Plotly, Numpy, Matplotlib. |

**TABLE OF CONTENTS**

|  |
| --- |
|  |

**Chapter 1**

**Introduction**

1.1 Context …………………………………………………. 1

1.2 Motivations ……………………………………………... 1

1.3 Idea ……………………………………………………... 1

1.4 Sources …………………………………………………. 2

1.5 Purpose ………………………………………….……… 2

1.6 Resources ……………………………………….……… 4

**Chapter 2**

**The Data**

2.1 Introduction ……………………………………….……. 6

2.2 Methodologies to get Data ……………………………... 6

2.3 Data Sources …………………………………….….…... 6

2.4 The Data …………………………………………………7

**Chapter 3**

**Data Wrangling**

3.1 Introduction …………………………………….………. 13

3.2 Duplicates ………………………………………………. 13

3.3 Missing Values ……………………………......………... 14

3.3.1 Finding Missing Values …………………………... 14

3.3.2 Removing Missing Values ………………………... 15

3.4 Normalization …………………………………...………. 16

3.5 Outliers …………………………………………………... 17

3.5.1 Detection …………………………………………... 17

3.5.2 Removal and Replace ……….……………………... 18

**Chapter 4**

**Machine Learning**

4.1 Introduction ………………………………………………. 20

4.2 Supervised Learning …………………………….……….... 21

4.3 Classification ……………………………………………….22

4.3.1 Decision Tree Classification ……………….………... 23

4.4 Unsupervised Learning …………………………....….…….24

4.5 Clustering ………………………………………….………. 24

4.5.1 K-Means Classification………….………….………... 25

**Chapter 5**

**Data Visualization**

5.1 Introduction …………………………………….……….…. 28

5.3 Visualizing Graphs ………………….……………………... 29

5.3.1 Choropleth Map …………………….………………... 30

5.3.2 Word Cloud………………………………….………... 31

5.3.3 Scatter Plots………………………………….………... 32

5.3.4 Pie Charts…………………………………………….... 33

**Chapter 6**

**Dashboards**

6.1 Introduction …………………………….…………………... 34

6.2 Dash by Plotly ………………………………………….…... 34

6.3 Dash Architecture ……………………………………….…...36

6.4 Dash App ………………………………………………...…. 37

6.4.1 Dash App Layout…………………………….………... 38

6.4.2 Graphical User Interface ……………………………… 41

**Chapter 7**

**Conclusion and Future Work**

7.1 Conclusion ……………….…………………………. 46

**References**

**CHAPTER 1**

**INTRODUCTION**

**1.1 Context**

This project has been done as a part of my course for the **BS Information Technology** at Bahaudin Zakariya University Multan. I had Six month to fulfil the requirements to succeed the module. Every three weeks, a meeting was organized to show and report my progress and fix the next objectives.

**1.2 Motivations**

Being extremely interested in everything having a relation with the Data Science, the independent project was a great occasion to give me the time to learn and confirm my interest for this field. The fact that we can handle a million of data and take useful insights from it. We can use Data Mining in Finance, Medicine, almost everywhere. That is why I decided to conduct my project around the Data Mining.

**1.3 Idea**

This project was motivated by my desire to investigate the descriptive analysis field of Data Mining since it allows to approach big data which is a very hot topic. Following my previous experience where it was about airline data descriptive analysis, I applied the same idea with technologies and try to figure out which is:

**\*There was a line space between the bullets which have now removed**

* What are the top programming languages in demand?
* What are the top database skills in demand?
* What are the popular Development Platforms?
* What are the popular Web frameworks?

**1.4 Sources**

Because I truly think that sharing sources and knowledges allow to help others but also ourselves,

the sources of the project are available at the following link:

https://github.com/jonygujjar/Final-Year-Project

Feel free to give me your point of view or ideas for anything you want. I used ipython notebook which is very useful to understand the entire process of my project since you can follow each step with the corresponding code.

\*There was a space there which now have been removed

**1.5 Purpose**

To keep pace with changing technologies and remain competitive, the organization regularly analyses data to help identify future skill requirements. That is what my project is doing. This project will evaluate some of the top trending technologies that will be useful not only for the organization but also for the tech professionals by different ways. Here are some of the main processes where my project will be helpful.

* Organization Hiring Process.
* Organization System Upgradation or Modifications.
* Skill development Process.

**1.Hiring process**

Hiring Process always play an important role in the future of every organization. By the hired person, the direction any organization has been determined. In hiring process, the skill set or required skills have been listed for the job postings. In this job posting the required skills are identified by management of the organization from which the future of the organization is determined. This skill set is very important. So, in order to determined which skill set is trending now and have long lasting positive effect on organization. There is always a problem in selecting this skill set which are not only effective but also are future proof.

That is what my project is solving this problem for the organizations which provides a list of top trending technologies available for hiring process.

**2.System upgradation**

System upgradation or new system installation is also important for the proper and efficient working of an organization. Multiple technologies are available to implement in existing and new system. So, in order to decide which technology should be implemented for maximum performance and efficiency. For example, there are multiple databases are available for information system, but we have to choose the one which should be cost efficient and future proof so that organization do not need to change it after some time period. Here comes my project to solve the mystery which proposed the top trending databases and other technologies where you can choose one from a short-listed trending technologies according to your organization needs.

**3.Skill development process**

Every organization pays a lot for **skill development process**. This is not the case of only organization but also for a single person who have a desire to develop new skills. There are a lot more technologies are available in which only some are more effective and provide a trust of future proofing. To determine which skills are trending and have a secure profitable future needs complex research and even we can go to a wrong direction which in last represent a loss in term of money and time. To ensure which skill is desired and have a profitable future we need a companion that provides a accurate set of data upon which we can made a right decision. That is what my project is doing it provides a list of top trending technologies where you or your organization can invest.

**1.6 Resources**

There are multiple resources are used in my project from data wrangling process to the data visualization process which are listed below:

* **Python** (Python is an interpreted high-level general-purpose programming language). Link: [https://www.python.org](https://www.python.org/)
* **Pandas** (pandas is a software library written for the Python programming language for data manipulation and analysis). Link: <https://pandas.pydata.org>
* **NumPy** (NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays). Link: <https://numpy.org>
* **Matplotlib** (Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy). Link: [https://matplotlib.org](https://www.python.org/)
* **Dash (**Dash is a python module which is used for creating web dashboards from python program.) Link: https://dash.plotly.com/
* **Scikit-learn** (Scikit-learn is a free software machine learning library for the Python programming language.) Link: <https://scikit-learn.org/stable/.org>
* **Plotly** ( Plotly's Python graphing library makes interactive, publication-quality graphs.) Link: https://plotly.com/python/
* **Folium** (Folium is a Python library used for visualizing geospatial data.) Link: <http://python-visualization.github.io/folium/>
* **Jupyter Notebook** (Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages".)

Link: https://jupyter.org/

* **Dash Bootsrap** (dash-bootstrap-components is a library of Bootstrap components for Plotly Dash, that makes it easier to build consistently styled apps with complex, responsive layouts.) Link: https://dash-bootstrap-components.opensource.faculty.ai/

**CHAPTER 2**

**THE DATA**

**2.1 Introduction**

Data sourcing is key process of and data mining project because the whole project and outcomes is relying on this data. To get data for my project of finding trending technologies different data sources are available but I have to choose the one which is most reliable and trusted. Mostly tech people rely on web data sources, so I do the same. First, I have to choose the methodologies to get the data from web data sources.

**2.2 Methodologies to get Data.**

There are multiple methods available in data mining to get the data from data sources, but I am trying only one method to get data from web data source which are listed below:

* **API** (Data is collected through an Application Programing Interface)
* **Surveys** (Data is collected through trusted and verified surveys from well-known Organizations.)
* **Web Scraping** (Data is collect through scraping different websites.)

**2.3 Data Sources**

The authenticity and reliability of any data mining project is depending upon the authentic and reliable data source. Hence, I have to choose the data source carefully and after a lot of research I got to some authentic and most reliable data sources which are market leaders, and no one could question their authenticity. These data sources are listed as:

* **Stacks Overflow Developers Survey 2020**

Stacks Overflow is the leaders of the developer community and are used on daily basis from the tech professionals. I use stacks overflow survey as data source for my project.

**2.4 The Data**

The data from data sources is collected through different methodologies. Now the data is available for my project and the properties of this data is given below:

**The Developers Survey Data is collected from Stacks Overflow Survey so the questions of the survey along with response column is given below:**

| **Column Name** | **Question Text** |
| --- | --- |
| Respondent | Randomized respondent ID number (not in order of survey response time) |
| MainBranch | Which of the following options best describes you today? Here, by "developer" we mean "someone who writes code." |
| Hobbyist | Do you code as a hobby? |
| OpenSourcer | How often do you contribute to open source? |
| OpenSource | How do you feel about the quality of open source software (OSS)? |
| Employment | Which of the following best describes your current employment status? |
| Country | In which country do you currently reside? |
| Student | Are you currently enrolled in a formal, degree-granting college or university program? |
| EdLevel | Which of the following best describes the highest level of formal education that you have completed? |
| UndergradMajor | What was your main or most important field of study? |
| EduOther | Which of the following types of non-degree education have you used or participated in? Please select all that apply. |
| OrgSize | Approximately how many people are employed by the company or organization you work for? |
| DevType | Which of the following describe you? Please select all that apply. |
| YearsCode | Including any education, how many years have you been coding? |
| Age1stCode | At what age did you write your first line of code or program? (E.g., webpage, Hello World, Scratch project) |
| YearsCodePro | How many years have you coded professionally (as a part of your work)? |
| CareerSat | Overall, how satisfied are you with your career thus far? |
| JobSat | How satisfied are you with your current job? (If you work multiple jobs, answer for the one you spend the most hours on.) |
| MgrIdiot | How confident are you that your manager knows what they're doing? |
| MgrMoney | Do you believe that you need to be a manager to make more money? |
| MgrWant | Do you want to become a manager yourself in the future? |
| JobSeek | Which of the following best describes your current job-seeking status? |
| LastHireDate | When was the last time that you took a job with a new employer? |
| LastInt | In your most recent successful job interview (resulting in a job offer), you were asked to... (check all that apply) |
| FizzBuzz | Have you ever been asked to solve FizzBuzz in an interview? |
| JobFactors | Imagine that you are deciding between two job offers with the same compensation, benefits, and location. Of the following factors, which 3 are MOST important to you? |
| ResumeUpdate | Think back to the last time you updated your resumé, CV, or an online profile on a job site. What is the PRIMARY reason that you did so? |
| CurrencySymbol | Which currency do you use day-to-day? If your answer is complicated, please pick the one you are most comfortable estimating in. |
| CurrencyDesc | Which currency do you use day-to-day? If your answer is complicated, please pick the one you are most comfortable estimating in. |
| CompTotal | What is your current total compensation (salary, bonuses, and perks, before taxes and deductions), in Currency Symbol? Please enter a whole number in the box below, without any punctuation. If you are paid hourly, please estimate an equivalent weekly, monthly, or yearly salary. If you prefer not to answer, please leave the box empty. |
| CompFreq | Is that compensation weekly, monthly, or yearly? |
| ConvertedComp | Salary converted to annual USD salaries using the exchange rate on 2019-02-01, assuming 12 working months and 50 working weeks. |
| WorkWeekHrs | On average, how many hours per week do you work? |
| WorkPlan | How structured or planned is your work? |
| WorkChallenge | Of these options, what are your greatest challenges to productivity as a developer? Select up to 3: |
| WorkRemote | How often do you work remotely? |
| WorkLoc | Where would you prefer to work? |
| ImpSyn | For the specific work you do, and the years of experience you have, how do you rate your own level of competence? |
| CodeRev | Do you review code as part of your work? |
| CodeRevHrs | On average, how many hours per week do you spend on code review? |
| UnitTests | Does your company regularly employ unit tests in the development of their products? |
| PurchaseHow | How does your company make decisions about purchasing new technology (cloud, AI, IoT, databases)? |
| PurchaseWhat | What level of influence do you, personally, have over new technology purchases at your organization? |
| LanguageWorkedWith | Which of the following programming, scripting, and markup languages have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the language and want to continue to do so, please check both boxes in that row.) |
| LanguageDesireNextYear | Which of the following programming, scripting, and markup languages have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the language and want to continue to do so, please check both boxes in that row.) |
| DatabaseWorkedWith | Which of the following database environments have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the database and want to continue to do so, please check both boxes in that row.) |
| DatabaseDesireNextYear | Which of the following database environments have you done extensive development work in over the past year and which do you want to work in over the next year? (If you both worked with the database and want to continue to do so, please check both boxes in that row.) |
| PlatformWorkedWith | Which of the following platforms have you done extensive development work for over the past year? (If you both developed for the platform and want to continue to do so, please check both boxes in that row.) |
| PlatformDesireNextYear | Which of the following platforms have you done extensive development work for over the past year? (If you both developed for the platform and want to continue to do so, please check both boxes in that row.) |
| WebFrameWorkedWith | Which of the following web frameworks have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.) |
| WebFrameDesireNextYear | Which of the following web frameworks have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.) |
| MiscTechWorkedWith | Which of the following other frameworks, libraries, and tools have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the technology and want to continue to do so, please check both boxes in that row.) |
| MiscTechDesireNextYear | Which of the following other frameworks, libraries, and tools have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the technology and want to continue to do so, please check both boxes in that row.) |
| DevEnviron | Which development environment(s) do you use regularly? Please check all that apply. |
| OpSys | What is the primary operating system in which you work? |
| Containers | How do you use containers (Docker, Open Container Initiative (OCI), etc.)? |
| BlockchainOrg | How is your organization thinking about or implementing blockchain technology? |
| BlockchainIs | Blockchain / cryptocurrency technology is primarily: |
| BetterLife | Do you think people born today will have a better life than their parents? |
| ITperson | Are you the "IT support person" for your family? |
| OffOn | Have you tried turning it off and on again? |
| SocialMedia | What social media site do you use the most? |
| Extraversion | Do you prefer online chat or IRL conversations? |
| ScreenName | What do you call it? |
| SOVisit1st | To the best of your memory, when did you first visit Stack Overflow? |
| SOVisitFreq | How frequently would you say you visit Stack Overflow? |
| SOVisitTo | I visit Stack Overflow to... (check all that apply) |
| SOFindAnswer | On average, how many times a week do you find (and use) an answer on Stack Overflow? |
| SOTimeSaved | Think back to the last time you solved a coding problem using Stack Overflow, as well as the last time you solved a problem using a different resource. Which was faster? |
| SOHowMuchTime | About how much time did you save? If you're not sure, please use your best estimate. |
| SOAccount | Do you have a Stack Overflow account? |
| SOPartFreq | How frequently would you say you participate in Q&A on Stack Overflow? By participate we mean ask, answer, vote for, or comment on questions. |
| SOJobs | Have you ever used or visited Stack Overflow Jobs? |
| EntTeams | Have you ever used Stack Overflow for Enterprise or Stack Overflow for Teams? |
| SOComm | Do you consider yourself a member of the Stack Overflow community? |
| WelcomeChange | Compared to last year, how welcome do you feel on Stack Overflow? |
| SONewContent | Would you like to see any of the following on Stack Overflow? Check all that apply. |
| Age | What is your age (in years)? If you prefer not to answer, you may leave this question blank. |
| Gender | Which of the following do you currently identify as? Please select all that apply. If you prefer not to answer, you may leave this question blank. |
| Trans | Do you identify as transgender? |
| Sexuality | Which of the following do you currently identify as? Please select all that apply. If you prefer not to answer, you may leave this question blank. |
| Ethnicity | Which of the following do you identify as? Please check all that apply. If you prefer not to answer, you may leave this question blank. |
| Dependents | Do you have any dependents (e.g., children, elders, or others) that you care for? |
| SurveyLength | How do you feel about the length of the survey this year? |
| SurveyEase | How easy or difficult was this survey to complete? |

**Table 1.1 Survey Result column and question text**

**Source: Stacksoverflow Developers Survey. [**https://insights.stackoverflow.com/survey**]**

**Here is the sample of Dataset:**

Table

Description automatically generated**Table 1.2 Sample of Dataset**

**Number of columns and rows:**

Number of rows : 64461

Number of columns : 61

**Statistical Summary of Data:**

**Table

Description automatically generated**

**Table 1.3 Statistical Summary**

**Data Types:**

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Respondent 14281 non-null int64

1 Age 13768 non-null float64

2 ConvertedComp 13620 non-null float64

3 CompFreq 14281 non-null object

4 CompTotal 14281 non-null float64

5 Country 14281 non-null object

6 DatabaseDesireNextYear 14281 non-null object

7 DatabaseWorkedWith 14281 non-null object

8 DevType 14281 non-null object

9 EdLevel 14281 non-null object

10 Gender 14281 non-null object

11 LanguageDesireNextYear 14281 non-null object

12 LanguageWorkedWith 14281 non-null object

13 MiscTechDesireNextYear 14281 non-null object

14 MiscTechWorkedWith 14281 non-null object

15 OpSys 14281 non-null object

16 PlatformDesireNextYear 14281 non-null object

17 PlatformWorkedWith 14281 non-null object

18 UndergradMajor 14281 non-null object

19 WebframeDesireNextYear 14281 non-null object

20 WebframeWorkedWith 14281 non-null object

21 WorkWeekHrs 14281 non-null float64

22 YearsCodePro 14281 non-null object

23 NormalizedAnnualCompensation 14281 non-null float64

dtypes: float64(5), int64(1), object(18)

**Table 1.4 Data Types**

For Complete Summary of the Dataset please download the html report generated from

Pandas profiling from the link below:

Link: https://github.com/jonygujjar/Final-Year-Project/blob/5db05dc13ead1f445847e7fd6588f29ebba054f2/survey\_report.html

**CHAPTER 3**

**DATA WRANGLING**

**3.1 Data Wrangling**

Data wrangling, sometimes referred to as data munging, is the process of transforming and mapping data from one "raw" data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. The goal of data wrangling is to assure quality and useful data. Data analysts typically spend the majority of their time in the process of data wrangling compared to the actual analysis of the data.

Till now we have a dataset which need to be pre-processed and cleaned before going to analysis phase. Dataset may contain missing values, outliers, and other in-consistencies. Dataset is processed into different processes until it is declared as cleaned.

**3.2 Removing Duplicates.**

"Duplication" just means that you have repeated data in your dataset. This could be due to things like data entry errors or data collection methods. For example, if you're using a web scraper you may happen to scrape the same webpage more than once, or the same information from two different pages.

First, in data wrangling or pre-processing we need to identify the duplicates first and their numbers:

# get the number of duplicates

duplicates = df\_final[df.duplicated()]

print(len(duplicates))

0

**Code Snap 2.1. Removing Duplicates**

Above code snippets shows that there are No duplicates values in the dataset which need to be removed:

## 2.3.2 Finding Missing values:

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.

Sometimes missing values are caused by the researcher—for example, when data collection is done improperly or mistakes are made in data entry.

Now the duplicates have been removed and we may have the missing values for each column as:

df\_final.isnull().sum()

The number of missing values for each column are as :

Respondent 0

Age 19015

ConvertedComp 29705

CompFreq 24392

CompTotal 29635

Country 389

DatabaseDesireNextYear 20391

DatabaseWorkedWith 14924

DevType 15091

EdLevel 7030

Gender 13904

LanguageDesireNextYear 10348

LanguageWorkedWith 7083

MiscTechDesireNextYear 22082

MiscTechWorkedWith 24147

OpSys 8233

PlatformDesireNextYear 13856

PlatformWorkedWith 10618

UndergradMajor 13466

WebframeDesireNextYear 24437

WebframeWorkedWith 22182

WorkWeekHrs 23310

YearsCodePro 18112

**Code Snap 2.2 Finding Missing Values**

## 2.3.3 Imputing missing values

We need to replace the missing values with appropriate values i.e with max value or simply drop these rows as:

Hence, we have 0 null or missing values.

df\_final.dropna(axis=0,inplace=True)

df\_final.reset\_index(drop=True,inplace=True)

df\_final.isnull().sum()

Respondent 0

Age 0

ConvertedComp 0

CompFreq 0

CompTotal 0

Country 0

DatabaseDesireNextYear 0

DatabaseWorkedWith 0

DevType 0

EdLevel 0

Gender 0

LanguageDesireNextYear 0

LanguageWorkedWith 0

MiscTechDesireNextYear 0

MiscTechWorkedWith 0

OpSys 0

PlatformDesireNextYear 0

PlatformWorkedWith 0

UndergradMajor 0

WebframeDesireNextYear 0

WebframeWorkedWith 0

WorkWeekHrs 0

YearsCodePro 0

**Code Snap 2.3 Imputing Missing Values**

**2.3.4 Normalizing Data**

Dataset normalization is the process of structuring a dataset, usually a relational dataset, in accordance with a series of so-called normal forms in order to reduce data redundancy. There are two columns in the dataset that talk about compensation.

One is "CompFreq". This column shows how often a developer is paid (Yearly, Monthly, Weekly). The other is "CompTotal". This column talks about how much the developer is paid per Year, Month, or Week depending upon his/her "CompFreq". This makes it difficult to compare the total compensation of the developers.

In this section you will create a new column called 'NormalizedAnnualCompensation' which contains the 'Annual Compensation' irrespective of the 'CompFreq'. Once this column is ready, it makes comparison of salaries easy.

List out the various categories in the column 'CompFreq':

Df\_final['CompFreq'].value\_counts()

Yearly 6073

Monthly 4788

Weekly 331

Name: CompFreq, dtype: int64

**Code Snap 2.4 Count values for each type**

Now we must normalize this column as:

conditions = [(df\_final['CompFreq']=='Yearly'),

(df\_final['CompFreq']=='Monthly'),

(df\_final['CompFreq']=='Weekly') ]

values = [(df\_final['CompTotal']),

(df\_final['CompTotal']\*12),

(df\_final['CompTotal']\*52)]

df\_final['NormalizedAnnualCompensation'] = np.select(conditions,values)

df\_final[['NormalizedAnnualCompensation']].head()

NormalizedAnnualCompensation

0 3.824729e+243

1 1.160000e+05

2 2.500000e+04

3 3.100000e+04

4 6.600000e+04

**Code Snap 2.5 Normalizing Data**

**2.3.5 Outliers**

In statistics, an outlier is a data point that differs significantly from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses.

We have to find outliers also for a fair and accurate data analysis so we are now identifying the ootliers first and then we have to remove them.

Find out if outliers exist in the column ConvertedComp using a box plot?

A picture containing graphical user interface

Description automatically generated

**Figure 2.1 Box Plot of ConvertedComp**

Find out the Inter Quartile Range for the column `ConvertedComp`:

Q1 = df['ConvertedComp'].quantile(0.25)

Q3 = df['ConvertedComp'].quantile(0.75)

IQR = Q3 - Q1

IQR

77306.05014961446

**Code Snap 2.6 Finding IQR**

Find out the upper and lower bounds.

upper = Q3+(1.5\*IQR)

lower = Q1-(1.5\*IQR)

upper,lower

(209698.0, -82830.0)

**Code Snap 2.7 Finding Upper and Lower bounds**

Identify how many outliers are there in the ConvertedComp column.

**Code Snap 2.8 Length of outliers**

outliers = [x for x in df['ConvertedComp'] if x < lower or x > upper]

print(len(outliers))

661

Create a new data frame by removing the outliers from the ConvertedComp column and replace it with original column in the dataset.

new\_df = [x for x in df['ConvertedComp'] if x >= lower and x <= upper]

new\_df = pd.DataFrame(new\_df)

new\_df.columns = ['ConvertedComp']

new\_df = df\_final[‘ConvertedComp’]

**Code Snap 2.9 Replacing datasets**

Till now we have removed all the outliers and dataset is free from outliers.

**CHAPTER 4**

**MACHINE LEARNING**

**4.1 Introduction:**

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as predictive analytics.

**Approaches:**

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

**Supervised learning:** The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

**Unsupervised learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

**Reinforcement learning:** A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

**4.2 Supervised Machine Learning:**

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross validation process. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox.

**Supervised learning** uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.

Supervised learning can be separated into two types of problems when data mining—classification and regression:

**Classification** uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest, which are described in more detail below.

**Regression** is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.

**4.3.1 Classification:**

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

Classification belongs to the category of supervised learning where the targets also provided with the input data. There are many applications in classification in many domains such as in credit approval, medical diagnosis, target marketing etc.

**Classification Algorithms:**

There is a lot of classification algorithms available now, but it is not possible to conclude which one is superior to other. It depends on the application and nature of available data set. For example, if the classes are linearly separable, the linear classifiers like Logistic regression, Fisher’s linear discriminant can outperform sophisticated models and vice versa.

**Top 5 Classification Algorithms in Machine Learning**

* Logistic Regression.
* Naive Bayes.
* K-Nearest Neighbours.
* Decision Tree.
* Support Vector Machines.

**\*There was a space there which now have been removed**

**4.3.1 Decision-Tree Classifier**

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses *Entropy* and *Information Gain* to construct a decision tree. In ZeroR model there is no predictor, in OneR model we try to find the single best predictor, naive Bayesian includes all predictors using Bayes' rule and the independence assumptions between predictors but decision tree includes all predictors with the dependence assumptions between predictors.

**Entropy**

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

**Information Gain**

Shannon invented the concept of entropy, which measures the impurity of the input set. In physics and mathematics, entropy referred as the randomness or the impurity in the system. In information theory, it refers to the impurity in a group of examples. Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values. ID3 (Iterative Dichotomiser) decision tree algorithm uses information gain.

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X, y,random\_state=1)

mod1 = tree.DecisionTreeClassifier(criterion= 'entropy', max\_depth= 10, random\_state=1)

mod1 = mod1.fit(X\_train,y\_train)

y\_pred = mod1.predict(X\_test)

print('Accuracy Score is :',metrics.accuracy\_score(y\_test,y\_pred))

Accuracy Score is : 0.5284234108092971

**Code Snap 4.1 Decision Tree Algorithm**

**4.4 Unsupervised Learning**

Unsupervised learning, also known as [unsupervised machine learning](https://www.ibm.com/cloud/learn/machine-learning), uses machine learning algorithms to analyse and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

Unsupervised learning models are utilized for three main tasks—clustering, association, and dimensionality reduction. Below we’ll define each learning method and highlight common algorithms and approaches to conduct them effectively.

**4.5 Clustering**

Clustering isa data mining technique which groups unlabelled data based on their similarities or differences. Clustering algorithms are used to process raw, unclassified data objects into groups represented by structures or patterns in the information. Clustering algorithms can be categorized into a few types, specifically exclusive, overlapping, hierarchical, and probabilistic.

Here we are discussing mainly popular Clustering algorithms that are widely used in machine learning:

1. **K-Means algorithm:** The k-means algorithm is one of the most popular clustering algorithms. It classifies the dataset by dividing the samples into different clusters of equal variances. The number of clusters must be specified in this algorithm. It is fast with fewer computations required, with the linear complexity of **O(n).**
2. **Mean-shift algorithm:** Mean-shift algorithm tries to find the dense areas in the smooth density of data points. It is an example of a centroid-based model, that works on updating the candidates for centroid to be the center of the points within a given region.
3. **DBSCAN Algorithm:** It stands **for Density-Based Spatial Clustering of Applications with Noise**. It is an example of a density-based model similar to the mean-shift, but with some remarkable advantages. In this algorithm, the areas of high density are separated by the areas of low density. Because of this, the clusters can be found in any arbitrary shape.
4. **Expectation-Maximization Clustering using GMM:** This algorithm can be used as an alternative for the k-means algorithm or for those cases where K-means can be failed. In GMM, it is assumed that the data points are Gaussian distributed.
5. **Agglomerative Hierarchical algorithm:** The Agglomerative hierarchical algorithm performs the bottom-up hierarchical clustering. In this, each data point is treated as a single cluster at the outset and then successively merged. The cluster hierarchy can be represented as a tree-structure.
6. **Affinity Propagation:** It is different from other clustering algorithms as it does not require to specify the number of clusters. In this, each data point sends a message between the pair of data points until convergence. It has O(N2T) time complexity, which is the main drawback of this algorithm.

**4.5.1 K-Means Classification**

**K-means clustering** is a common example of an exclusive clustering method where data points are assigned into K groups, where K represents the number of clusters based on the distance from each group’s centroid. The data points closest to a given centroid will be clustered under the same category. A larger K value will be indicative of smaller groupings with more granularity whereas a smaller K value will have larger groupings and less granularity. K-means clustering is commonly used in market segmentation, document clustering, image segmentation, and image compression.

The function KMeans applies KMeans clustering to the train data with the number of classes as the number of clusters to be made and creates labels both for train and test data. The parameter output controls how do we want to use these new labels, ‘add’ will add the labels as a feature in the dataset and ‘replace’ will use the labels instead of the train and test dataset to train our classification model.

The k-means [clustering](https://www.javatpoint.com/clustering-in-machine-learning) algorithm mainly performs two tasks:

* Determines the best value for K centre points or centroids by an iterative process.
* Assigns each data point to its closest k-centre. Those data points which are near to the particular k-centre, create a cluster.

X2 = df\_mod2.drop(['ConvertedComp','income'],axis=1)

X2 = pd.get\_dummies(X2)

mod2 = KMeans(n\_clusters=2)

y2 = mod2.fit\_predict(X2)

**Code Snap 4.2 K-means Clustering**

Chart, bar chart

Description automatically generated

**Figure 4.1 Plotting Clusters**

Chart, bar chart

Description automatically generated

**Figure 4.2 Plot Clusters with Countries**

**Results From K-means Classification:**

* Most of Pakistani Respondents belongs to cluster 1 which means most of them have bachelor’s degree.
* More Indian Respondents belongs to cluster 1 which means most of them have bachelor’s degree.

**CHAPTER 5**

**DATA VISUALIZATION**

**5.1 Introduction to Data Visualization**

Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends, and outliers in large sets. The term is often used interchangeably with others, including information graphics, information visualization and statistical graphics.

Data visualization is one of the steps of the data mining process, which states that after data has been collected, processed, and modelled, it must be visualized for conclusions to be made. Data visualization is also an element of the broader data presentation architecture (DPA) discipline, which aims to identify, locate, manipulate, format, and deliver data in the most efficient way possible.

Data visualization refers to the techniques used to communicate data or information by encoding it as visual objects (e.g., points, lines, or bars) contained in graphics. The goal is to communicate information clearly and efficiently to users. It is one of the steps in data analysis or data science. According to Vitaly Friedman (2008) the "main goal of data visualization is to communicate information clearly and effectively through graphical means. It doesn't mean that data visualization needs to look boring to be functional or extremely sophisticated to look beautiful. To convey ideas effectively, both aesthetic form and functionality need to go hand in hand, providing insights into a rather sparse and complex data set by communicating its key aspects in a more intuitive way. Yet designers often fail to achieve a balance between form and function, creating gorgeous data visualizations which fail to serve their main purpose — to communicate information".

Indeed, Fernanda Vegas and Martin M. Wattenberg suggested that an ideal visualization should not only communicate clearly but stimulate viewer engagement and attention.

Data visualization is closely related to information graphics, information visualization, scientific visualization, exploratory data analysis and statistical graphics. In the new millennium, data visualization has become an active area of research, teaching and development. According to Post et al. (2002), it has united scientific and information visualization.

In the commercial environment data visualization is often referred to as dashboards. Infographics are another very common form of data visualization.

**5.2 Visualizing Graphs**

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.

In this tutorial we are going to visualize undirected Graphs in Python with the help of plotting library.

**5.2.1 Choropleth Maps**

A choropleth map is a type of thematic map in which a set of pre-defined areas is coloured or patterned in proportion to a statistical variable that represents an aggregate summary of a geographic characteristic within each area, such as population density or per-capita income.

Choropleth maps are popular thematic maps used to represent statistical data through various shading patterns or symbols on predetermined geographic areas (i.e. countries). They are good at utilizing data to easily represent variability of the desired measurement, across a region.

A choropleth map of number of developers around the world is shown below:

Map

Description automatically generated

**Figure 5.1 Choropleth Map of Respondents**

**5.2.2 Word Cloud**

Word clouds (also known as text clouds or tag clouds) work in a simple way: the more a specific word appears in a source of textual data (such as a speech, blog post, or database), the bigger and bolder it appears in the word cloud.

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is.

A word cloud of platforms used by developers is shown below:

Text

Description automatically generated with low confidence

**Figure 5.2 Word Cloud of Platforms**

**5.2.3 Scatter Plot**

A scatter plot is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. If the points are coded, one additional variable can be displayed.

A scatter plot can suggest various kinds of correlations between variables with a certain confidence interval. For example, weight and height would be on the y-axis, and height would be on the x-axis. Correlations may be positive (rising), negative (falling), or null (uncorrelated). If the dots' pattern from lower left to upper right indicates a positive correlation between the variables being studied. If the pattern of dots slopes from upper left to lower right, it indicates a negative correlation.

A 3D Scatter plot of ‘YearsCodePro’, ‘Age’ by their ‘ConvertedComp’ is shown here:

A picture containing chart

Description automatically generated

**Figure 5.3 3D Scatter Plot of ConvertedComp by Age and YearsCodePro**

**5.2.4 Pie Chart**

A pie chart (or a circle chart) is a circular statistical graphic, which is divided into slices to illustrate numerical proportion. In a pie chart, the arc length of each slice (and consequently its central angle and area), is proportional to the quantity it represents.

A pie chart of ‘EdLevel’ by percentage of developers is shown as:

Chart, pie chart

Description automatically generated

**Figure 5.4 Pie Chart of Respondents by EduLevel**

**CHAPTER 6**

**DASHBOARDS**

**6.1 Introduction**

A dashboard is a data visualization tool that tracks, analyses, and displays KPIs, metrics, and critical data points. Dashboards empower both technical and non-technical users to understand and leverage business intelligence to make more informed decisions. Users actively participate in the analytics process by compiling data and visualizing trends or occurrences and uncovering an objective view of performance metrics that can be immediately understood.

Dashboards feature visualized data via charts, tables, and gauges. Viewers use these visualizations to monitor the health of the organization against established goals and industry benchmarks.

* **Dashboards are fuelled by Business Questions.**
* **Dashboards can focus on presenting operational and analytical data.**
* **Dashboards present interactive data visualizations.**

**6.2 Dash by Plotly**

Dash gives a point-&-click interface to models written in Python, R, and Julia - vastly expanding the notion of what's possible in a traditional "dashboard." With Dash apps, data scientists and engineers put complex Python analytics in the hands of business decision makers and operators.

Dash will help you build dashboards quickly. If you’re used to analyzing data or building data visualizations using Python, then Dash will be a useful addition to your toolbox.

Three technologies constitute the core of Dash:

1. **Flask** supplies the web server functionality.
2. **React.js** renders the user interface of the web page.
3. **Plotly.js** generates the charts used in your application.

Written on top of Plotly.js and React.js, Dash is ideal for building and deploying data apps with customized user interfaces in pure Python, R, or Julia. It's particularly suited for anyone who works with data.

Through a couple of simple patterns, Dash abstracts away all of the technologies and protocols that are required to build a full-stack web app with interactive data visualization.

Dash is simple enough that you can bind a user interface to your Python, R, or Julia code in less than 10 minutes.

Dash apps are rendered in the web browser. You can deploy your apps to VMs or Kubernetes clusters and then share them through URLs. Since Dash apps are viewed in the web browser, Dash is inherently cross-platform and mobile ready.

There is a lot behind the framework. To learn more about how it is built and what motivated Dash, read our post Dash is React for Python.

Dash is an open source library released under the permissive MIT license. Plotly develops Dash and also offers a platform for writing and deploying Dash apps in an enterprise environment.

**6.3 Dash Architecture:**

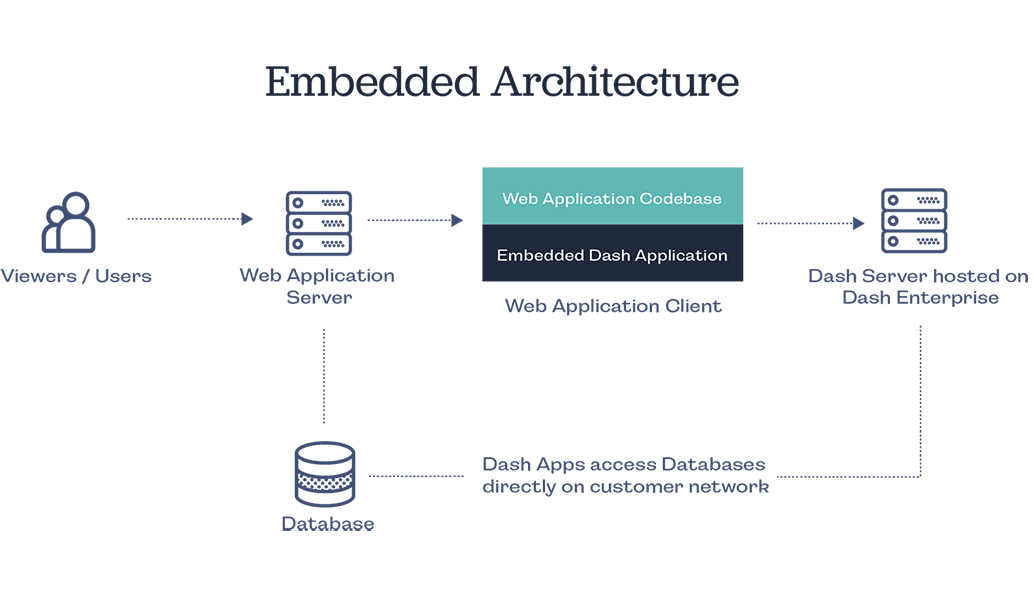
Flask and React

Dash applications are web servers running Flask and communicating JSON packets over HTTP requests. Dash’s frontend renders components using React.js, the Javascript user-interface library written and maintained by Facebook.

Flask is great. It’s widely adopted by the Python community and deployed in production environments everywhere. The underlying instance of Flask and all of its configurable properties is accessible to Dash app developers. For advanced developers, Dash apps can be extended through the rich set of Flask Plugins as well.

React is fantastic too. At Plotly, we’ve rewritten our entire web-platform and our online chart editor with React. One of the incredible things about React is how prolific and talented the community is. The open source React community has published thousands of high-quality interactive components, from Dropdowns to Sliders to Calendar Pickers to Interactive Tables.

Dash leverages the power of Flask and React, putting them to work for Python data scientists who may not be expert Web programmers.

****

**Figure 6.1 Embedded Architecture of Dash**

**Source: https://plotly.com/dash/embedding/**

Diagram

Description automatically generated

**Figure 6.2 Dash Architecture**

**Source: https://towardsdatascience.com/plotly-dash-or-react-js-plotly-js-b491b3615512**

**6.4 Dash App Creation:**

Dash is Python framework for building web applications. It built on top of Flask, Plotly.js, React and React Js. It enables you to build dashboards using pure Python. Dash is open source, and its apps run on the web browser. In this tutorial, we introduce the reader to Dash fundamentals and assume that they have prior experience with Plotly.

Dash Installation

In order to start using Dash, we have to install several packages.

1. The core dash backend.
2. Dash front-end
3. Dash HTML components
4. Dash core components
5. Plotly

**6.4.1 Dash App Layout**

A Dash application is usually composed of two parts. The first part is the layout and describes how the app will look like and the second part describes the interactivity of the application. Dash provides HTML classes that enable us to generate HTML content with Python. To use these classes, we need to import dash\_core\_components and dash\_html\_components. You can also create your own custom components using Javascript and React Js.

**HTML Components**

Just like in Flask we initialize Dash by calling the Dash class of dash. Once that is done we can create the layout for our application. We use the Div class from the dash\_html\_components to create an HTML Div. We then use the HTML components to generate HTML components such as H1, H2 etc. dash\_html\_components has all HTML tags. In order to create a graph on our layout, we use the Graph class from dash\_core\_components. Graph renders interactive data visualizations using plotly.js. The Graph class expects a figure object with the data to be plotted and the layout details. Dash also allows you to do stylings such as changing the background color and text color. You can change the background by using the style attribute and passing an object with your specific color.

In our case, we have defined a color dictionary with the background and text color we would like. Similarly, we can change the layout background using the plot\_bgcolor attribute.

In HTML the style property is specified using a semicolon, but in Dash, a dictionary is supplied. The keys in the dictionary are camelCased e.g text-align is textAlign. Instead of using classes like in HTML, className is used in Dash.

**Generating Scatter Plots**

In order to plot a scatter plot, we import the normal dash components as previously done. We also need to import Plotly graph\_objs in order to plot the scatter plot. As mentioned previously we use the Div class and Graph components from Dash in order to accomplish this. The Graph component takes a figure object which has the data and the layout description. We plot the scatter plot using graph\_objs scatter property. In order to make sure the plot is a scatter plot we pass a mode attribute and set it as markers. Otherwise, we would have lines on the graph.

**Core Components**

Next, let's look at some dash\_core\_components that you will encounter when using Dash. You can generate a drop down as shown below. You do by calling Dropdown off dash\_core\_components and passing the options as a list of dictionaries. You can set the default value using the values attribute and passing in the default option.

**Authentication**

Dash provides authentication through a separate package called dash-auth. It provides two modes of authentication HTTP Basic Auth and Plotly OAuth. In Basic Auth, you hardcode a set of usernames and passwords in your application. This method has some challenges such as users cannot log out of your application, users cannot create accounts or change passwords, and you are responsible for safely storing the usernames and passwords in your code. Plotly OAuth provides authentication through your online Plotly account, and it's not free.

**Dash Bootstrap Components**

dash-bootstrap-components is a library of Bootstrap components for Plotly Dash, that makes it easier to build consistently styled apps with complex, responsive layouts. Dash Bootstrap Components for Python can be easily installed with pip or conda.

Once installed, just link a Bootstrap stylesheet and start using the components exactly like you would use other Dash component libraries.

Bootstrap components are available as native Dash components to let you easily incorporate them into your Dash apps. Each component exposes a number of props to let you control the behaviour with Dash call-backs.

Dash Bootstrap Components is compatible with any Bootstrap v4 stylesheet of your choice. Learn how to customise the look of your app using the bundled themes or your own custom themes.

**Use Case Diagram of Dashboard:**

Current Trending Technologies

Demographics

In Future Trending

Technologies

User

**Figure 6.3 Use Case Diagram**

**6.4.2 Graphical User Interface (GUI)**

Graphical user interfaces would become the standard of user-cantered design in software application programming, providing users the capability to intuitively operate computers and other electronic devices through the direct manipulation of graphical icons such as buttons, scroll bars, windows, tabs, menus, cursors, and the mouse pointing device. Many modern graphical user interfaces feature touchscreen and voice-command interaction capabilities.

Graphical user interface design principles conform to the model–view–controller software pattern, which separates internal representations of information from the manner in which information is presented to the user, resulting in a platform where users are shown which functions are possible rather than requiring the input of command codes. Users interact with information by manipulating visual widgets, which are designed to respond in accordance with the type of data they hold and support the actions necessary to complete the user’s task.

**Authentication:**

**Graphical user interface, application

Description automatically generated**

**Figure 6.4 Authentication Page**

**Current Trending Technologies:**

**Chart, bar chart

Description automatically generated**

**Figure 6.5 Current trending tech. page**

**In Future Trending Technologies:**

**Graphical user interface, chart, application, bar chart

Description automatically generated**

**Figure 6.6 Future Trending tech. page**

**Demographics:**

**Graphical user interface, chart, application

Description automatically generated**

**Figure 6.7 Demographics page**

**CHAPTER 7**

**CONCLUSION & FUTURE WORK**

**7.1 Conclusion**

I categorize my conclusions into following in which I discuss my findings and implications one by one.

**Programming Languages Trends:**

**Chart, bar chart

Description automatically generated**

**Chart, bar chart, histogram

Description automatically generatedFigure 7.1 Comparison of Current and Future trends in Programming Languages**

**FINDINGS**

* **JavaScript is the most in demand programming language**
* **Python, HTML/CSS, and SQL are other in demand programming languages**
* **Skills in TypeScript is more desirable than Bash/Shell/PowerShell for next year.**

**IMPLICATIONS**

* **Python’s increase in demand may overtake JavaScript in the future.**
* **JavaScript, Python, HTML/CSS, and SQL are consistently in demand.**
* **Web development remains in high demand.**

**Databases Trends:**

**Chart, bar chart

Description automatically generated**

**Chart, bar chart

Description automatically generated**

**Figure 7.1 Comparison of Current and Future trends in Databases**

**IMPLICATIONS**

**• PostgreSQL and MongoDB are the most consistent in demand databases**

**• Open-source databases are preferred**

**• Demand is volatile**

**FINDINGS**

**• PostgreSQL and MongoDB have an increase in demand**

**• MySQL has a decrease in demand for next year**

**• Skills in Redis and Elastic Search are more desirable than Microsoft SQL Server and SQLite for next year**

**Overall Findings and Implications:**

**IMPLICATIONS**

**• IT professionals and companies have to learn and adapt to new demands**

**• Having skills that are high in demand leads to more job opportunities**

**• Gender gap will affect job hiring trend**

**FINDINGS**

**• Trends and demands fluctuate based on new technologies**

**• JavaScript and Python are the most popular programming languages**

**• Gender bias & gap in the IT field**

**Conclusion:**

**• The IT industry needs to diversify**

**• New technology leads to different trends and demands**

**• JavaScript and Python are the most popular programming languages**

**• PostgreSQL is the most popular databases**

**References:**

* **Python** Link: [https://www.python.org](https://www.python.org/)
* **Pandas** Link: <https://pandas.pydata.org>
* **NumPy** Link: <https://numpy.org>
* **Matplotlib** Link: [https://matplotlib.org](https://www.python.org/)
* **Dash** https://dash.plotly.com/
* **Scikit-learn** Link: <https://scikit-learn.org/stable/.org>
* **Plotly** Link: https://plotly.com/python/
* **Folium** Link: <http://python-visualization.github.io/folium/>
* **Jupyter Notebook** Link: https://jupyter.org/
* **Dash Bootsrap** Link: <https://dash-bootstrap-components.opensource.faculty.ai/>
* <https://www.ibm.com/cloud/learn/machine-learning>
* <https://dash.plotly.com/dash-core-components>
* <https://plotly.com/python/>
* <https://jupyter.org/documentation>
* <https://jupyter-plotly-dash.readthedocs.io/en/latest/>
* <https://amueller.github.io/word_cloud/>
* <https://www.geeksforgeeks.org/generating-word-cloud-python/>
* <https://www.datacamp.com/community/tutorials/learn-build-dash-python>
* https://github.com/Coding-with-Adam