Hassaan Naveed | N0898071

Nottingham trent university

Machine Learning For Data Analysis: Written Report

Comp20121

Contents

[1. Introduction 2](#_Toc71215126)

[CRISP-DM Methodology 2](#_Toc71215127)

[Project Task 2](#_Toc71215128)

[Insights 2](#_Toc71215129)

[2. Data Understanding, Pre-Processing, Exploratory Data Analysis 3](#_Toc71215130)

[Stack Overflow Developer Survey 3](#_Toc71215131)

[Data Attributes 3](#_Toc71215132)

[Characteristics of the Data Set 3](#_Toc71215133)

[Data Cleansing 4](#_Toc71215134)

[Exploratory Data Analysis 5](#_Toc71215135)

[3. Cluster Analysis 11](#_Toc71215136)

[Data Transformation and Normalisation 11](#_Toc71215137)

[Implementation of Cluster Analysis 12](#_Toc71215138)

[Characteristics of Clusters 13](#_Toc71215139)

[4. Machine Learning for Classification 15](#_Toc71215140)

[Workflow of Machine Learning 15](#_Toc71215141)

[Classification Methods 15](#_Toc71215142)

[Parameter Setting 15](#_Toc71215143)

[Implementation of Machine Learning Models 16](#_Toc71215144)

[Implementation of Ensemble Learning 19](#_Toc71215145)

[5. Evaluation 21](#_Toc71215146)

[Accuracy Score 21](#_Toc71215147)

[Confusion Matrix 22](#_Toc71215148)

[ROC Curve 23](#_Toc71215149)

[6. Discussion and Conclusion 24](#_Toc71215150)

[Summary 24](#_Toc71215151)

[Insights 24](#_Toc71215152)

[Understanding 24](#_Toc71215153)

[References 25](#_Toc71215154)

# 1. Introduction

## CRISP-DM Methodology

The **Cross-Industry Standard Process for Data Mining** (CRISP-DM) was developed in 1996. It is a non-propritary model that describes the processes of approaches used in data analysis. Major contributors of CRISP-DM were DaimlerChrysler, SPSS, NCR, and Teradata when it became a European Union backed project in 1997 (European Commission, 1997). IBM released an extension known as ASUM-DM in 2015, becoming the primary major corporation using the model.

It provides a structured approach to planning a data mining project, by fitting data mining into the general problem-solving strategy of business and research units. It consists of four levels of abstraction: phases, generic tasks, specialised tasks, and process instances. The first, topmost level is organised into a series of six phases which can be worked through sequentially and backtracked upon in an iterative, adaptive life cycle. The phases are: Business Research, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. Each of these phases includes several second-level generic tasks, which are intended to cover cover all possible data mining solutions. The third level, specialised tasks describe how the generic tasks should be carried out in various scenarios and situations. The fourth, process instance layer, is “a record of actions, decisions, and results of an actual data mining engagement”, which is organised in accordance to the tasks defined in higher levels (Wirth & Hipp, 2000).

CRISP-DM can be very advantageous while working in large, long term projects due to the notion that it is a process that requires thorough planning, reporting, and documentation to be successful (Wirth & Hipp, 2000). The idea of following a structured process can allow the stakeholders of the project to have more confidence in the development and success of said project. While having a very structured approach, CRISP-DM is also adaptable and flexible, allowing projects to work with agile concepts of unknown requirements and iterative design, allowing users of the model to gain deeper understanings of the data and the problem.

CRISP-DM is the most widely used data-mining model due to its advantages, which solved many problems existing in the data mining industry (Forbes, 2015).

## Project Task

The project that is being undertaken is an analysis of the “2020 Stack Overflow Annual Developer Survey” dataset. This is a 60-column dataset derived from a survey, covering a myriad topics such as geographic location, technologies used, work and education, and personal lives of programmers who use the Stack Overflow website. “The survey had over 65,000 responses from over 180 countries and independent territories”.

The CRISP-DM methodology will be applied to this project throughout its development. I will be following the six phases outlined in the model; understanding the data through exploratory data analysis, preparing the data through preprocessing, cleansing, and transformations, selecting and applying modelling techniques, and evaluating each one before coming to a final decision for the deployment.

## Insights

I hope to gain various insights throughout the project on the defining characteristics of high and low income developers, building a machine learning model for predicting the income of a developer and classifying them into a group based on the survey data provided, as well as gaining knowledge of core machine learning concepts.

# 2. Data Understanding, Pre-Processing, Exploratory Data Analysis

## Stack Overflow Developer Survey

The dataset being used will be the data collected from the “Stack Overflow Developers Survey 2020”. The data was collected by Stack Overflow, a forum based website for professional and enthusiast developers. There were approximately 65,000 responses in 2020 from over 180 countries and independent territories. The survey examines all aspects of a programmers’ experience, covering evey aspect of their lives, from professional to personal.

The survey was conducted between February the 5th and February the 28th 2020. We must consider external factors that could have influenced the survey, such as the fact that it was completed at the start of the COVID-9 pandemic. This is before any significant impact to the global economy occurred, however this is when the virus began spreading to regions of the world outside of East Asia. (World Health Organisation, 2020). We can also consider under-representation of minority communities, for example, 68.3% of the surveys’ respondants were from “White or of European Descent”. Additionally, 91.5% of respondants identified their gender “Male”, 99% of respondants identified as Cisgender, and 92.1% of respondants identified as heterosexual (Stack Overflow, 2020).

## Data Attributes

The dataset provided contains 61 questions, therefore it was imperative to the project that I selected the columns that were most important to the target variable, otherwise the machine learning models may produce skewed responses. According to a Stack Overflow blog post analysing the 2019 Developers Survey, the five most important factors for influencing a developers salary are location, education, years of professional coding experience, what kind of coding work developers do, and the technologies they use professionally (Stack Overflow, 2020).

Due to this, I decided to choose the features that reflected these topics the best, alongside the target variable.

|  |  |
| --- | --- |
| Feature | Description |
| ConvertedComp (Target) | Salary, converted to USD |
| YearsCodePro | Years spent coding professionally |
| Country | Country worked in |
| EdLevel | Highest education level achieved |
| DevType | Type of developer work done |
| LanguageWorkedWith | Programming languages used |

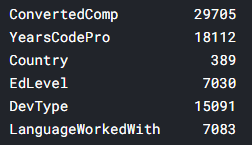
## Characteristics of the Data Set

Upon an initial exploration of the dataset, it can be found that the dataset contains exactly 64,461 entries within 61 columns. The majority of the data types are stored as a String, with four data types as floating point numbers and one as integers.

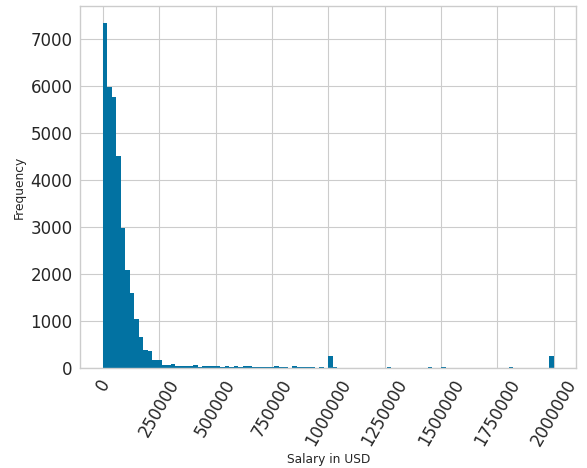
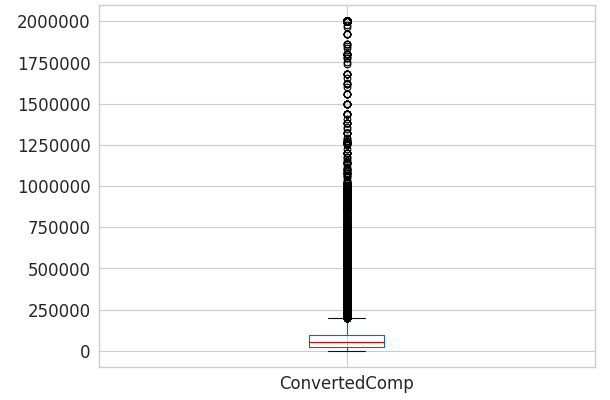
It can also be seen that not all questions have been answered by all respondants, for example ConvertedComp has only 34,756 entries. This suggests we must perform some data cleansing to identidy and remove any columns where the target variable hasn’t been answered, and fill in any other NaN values.

## Data Cleansing

When calculating the total number of missing values for all of our selected features, we can see that ConvertedComp has 29,705 missing values. All of these rows will have to be dropped, leaving the dataset with 34,756 columns to work with.

Additionally, all the other features also have missing values, with 389 missing values for Country at the minimum and 18,112 missing values for YearsCodePro at the maximum. We can assume this is because not all respondants have programmed in a professional capacity. Other missing values could be caused by respondants not wanting to answer certain questions, or accidentally leaving the field blank.

For these fields, I will calculate the modal values of each feature and replace all missing values with this modal value. I do this instead of dropping these rows as this leaves the dataset with more training data for machine learning. I have decided to use the modal value rather than the mean or median as most of the features are categorical data, while mean and median will only work with numerical.

Next, I will have to identify and remove any outliers in the numerical data. After plotting the frequencies of ConvertedComp as a box plot and histogram we can see that the majority of responses after 250,000 are outliers. It is not unheard of for programmers to be earning above this amount, with many likely in upper managerial roles. However many of these values may have been inputted accidentally, or simply fabricated, such as the slight uptick of entries at 1,000,000 and 2,000,000.

We also want to remove any values that are conspicuously low. The country with the lowest minimum annual salary in the world is Mexico, with the minimum annual salary in 2019 at 2,510 USD (OECD, 2021). After this research I decided to drop any rows in ConvertedComp where the entries were below 2500 or above 250,000.

The YearsCodePro feature is a range from 0 to 50, and therefore should not contain any noticable outliers, therefore I decided to not explore it for any erroneous data.

I decided to perform Data Cleansing before Exlporatory Data Analysis so the data I would be exploring would be completely valid, and therefore invalid data could not cause any incorrect presumptions about the relationships between the data.

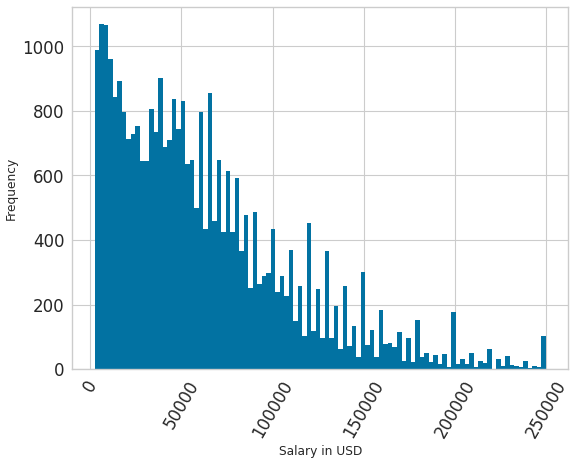
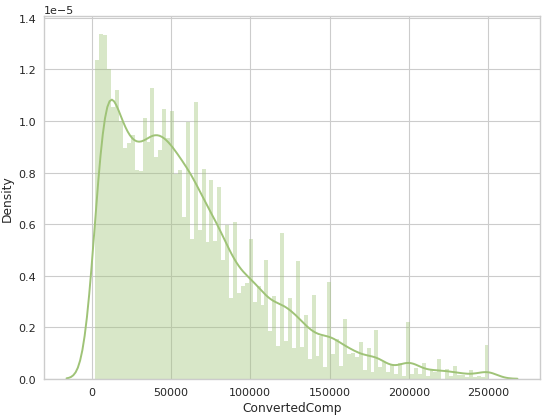
## Exploratory Data Analysis

### Initial Overview

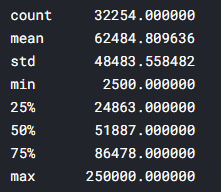
At a cursory look at the dataframe after conducting data cleansing, we can see that two of the features, ConvertedComp and YearsCodePro, are numerical, while the rest are categorical features.

There are over 180 possible responses for the Country feature, so I shall need to explore methods of plotting this in a concise and elegant manner, such as through only selecting the top countries to plot. We can also see that the responses for EdLevel are quite verbose, so I could find a way of encoding the responses in a more concise manner. Finally, the DevType and LanguagesWorkedWith features appear to have more than one response per row possible, so I will need to separate them out for plotting on graphs and performing machine learning.

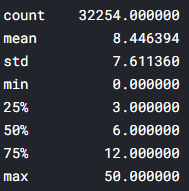
### Salary

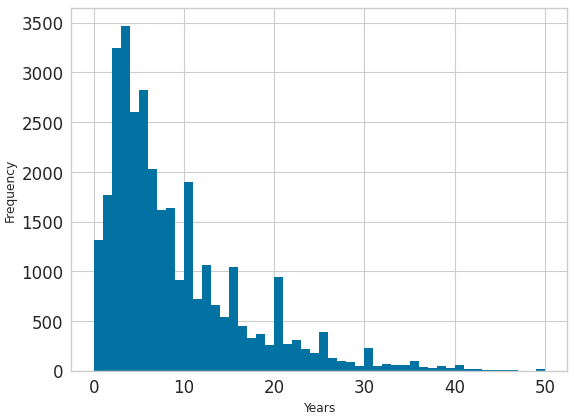
ConvertedComp is the target feature that I will be exploring, representing the annual salary in USD of each respondant. I will be comparing each feature to ConvertedComp in order to identify and measure their relationship.

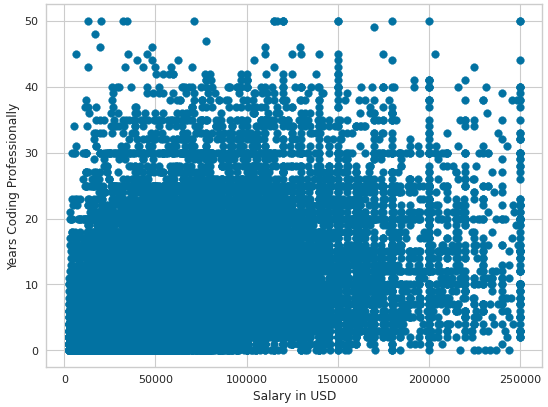
Plotting the ConvertedComp feature on a Histogram and Distribution Graph we can see that most developers earn below 50,000 USD, with a positive skew causing less programmers earning more. We can identify two peaks on the graph, one at approximately 20 – 30k USD and another at approximately 40 – 50k. These can be classified as the average salary for entry-level programmer jobs and senior-level programming jobs respectively.

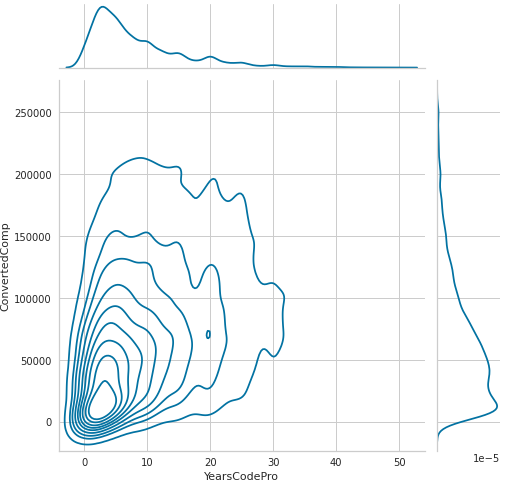
When looking at descriptive statistics, we can see that the average salary is approximately 60,000 USD, with the first quartile at 24,000 and the third quartile at 86,000, and the median at approximately 50,000. This reflects the data shown in the histograms, with a skew to the lower end of the range.

### Years Coding Professionally

YearsCodePro is a numerical feature ranging from 0 to 50, with responses of less than 1 year encoded as 0 and over 50 years as 50. It represents the number of years a respondant has spent coding professionally. An obvious requirement to this is that the respondant has worked in a professional capacity in a developer role, and so many rows had to have missing values dealt with.

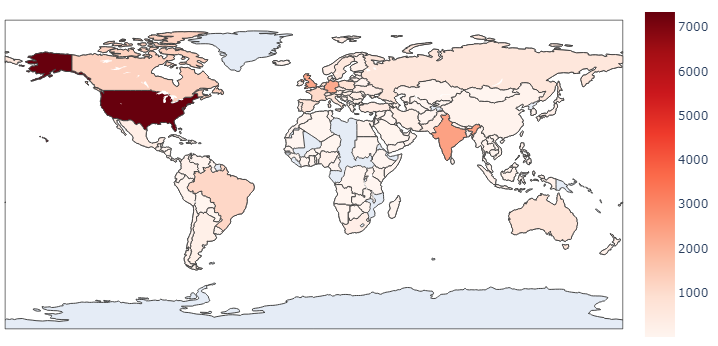
Looking at the descriptive statistics, we can see that the majority of developers have worked for 8 years professionally, with the first quartile at 3 years and the third quartile at 7 years. This suggests another positive skew, with less people having worked for longer years. This can be seen reflected in the histogram, with the majority of responses having worked below 10 years professionally.

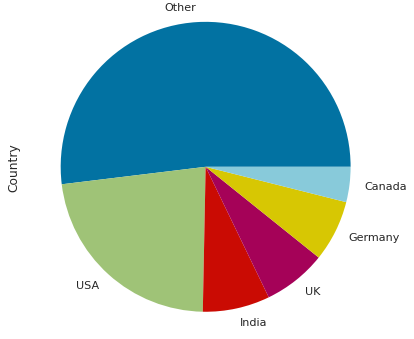
Comparing the Years Coding Professionally feature to Salary on a scattergram, we can see a positive relationship between those who have worked for longer and those who earn more. A larger amount of plots are concentrated towards the lower amount of years and the lower salary. The positive relationship is not very strong however, as it can be seen that the plots are highly distributed, and people who have worked between 40 to 50 years are not more likely to earn over 200,0000 USD as those who have worked 10 to 30 years.

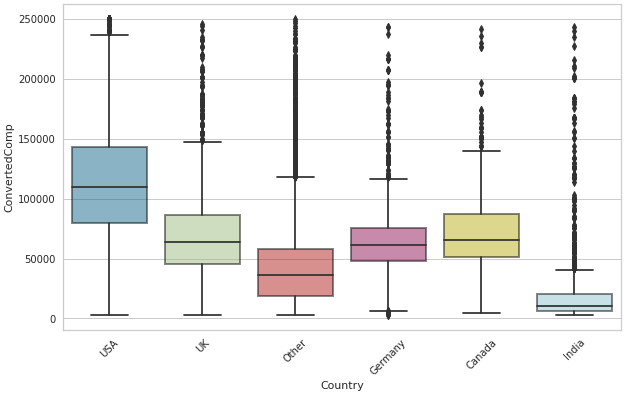
When comparing the values on a joint plot we can see a similar pattern. There is a much higher frequency of people who earn less than 50,000 USD and have worked less than 10 years. The distributions of both features decreases as the number increases, and a weak positive correlation can be seen.

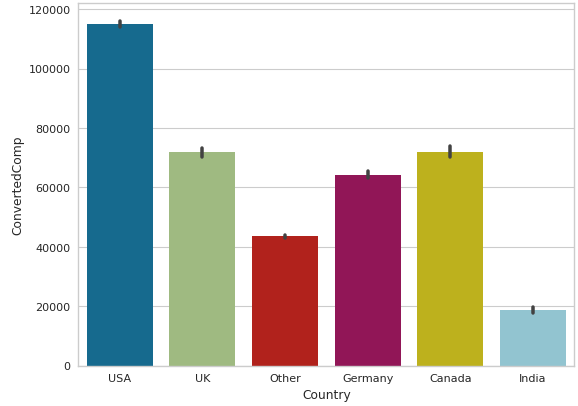
### Countries

Country is a categorical feature describing the country where developers are resided at the time of answering the survey. We can expect to learn which countries most developers are from, and which countries earn the most, and if the two are related.

After visualising the frequencies of the countries, we can conclude that the majority of developers are from the USA, which can be expected, as the majority of large tech companies operate from the USA. Many are from India, the UK, and Germany. There are also a considerable amount from Canada, Brazil, Russia. Australia, and parts of Europe.

After encoding the responses in a new datafram with the top five countries, and all other countries as “Other” for ease of plotting, we can see this reflected in a pie chart. The USA contains almost a quarter of all developers, with another quarted being shared between India, the UK, Germany, and Canada.

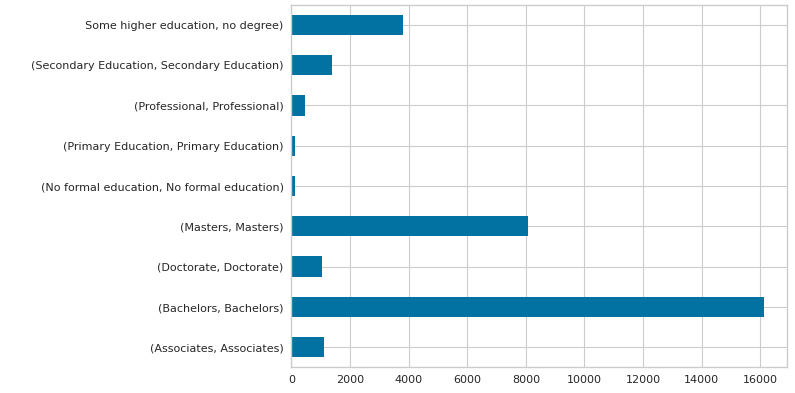
When comparing the countries to the salary in USD, we can see that the USA earns the most out of any country, with and average salary of almost 120,000 USD. This could be expected as the USA contains some of the largest tech companies such as Apple, Google, and Microsoft, as well as having the second highest annual average wage in USD as of 2019 (OECD, 2021).

The next most earning country is Canada, followed very closely by the United Kingdom and Germany. Both of these countries have a large presence in the tech industry, and have a high annual average wage.

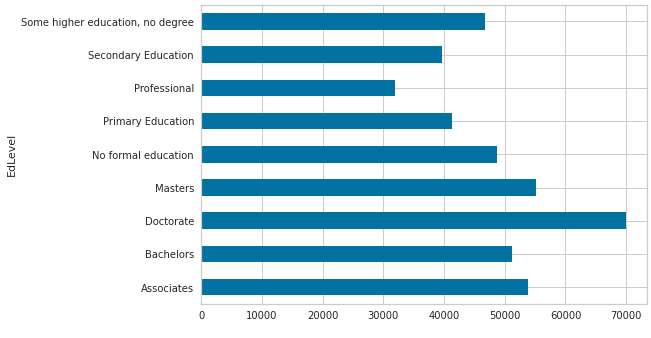
Other countries are the next largest earner, with India, occupying the second most programmers after the USA, earning the least. This could show that developers from the USA are more likely to be categorised as High Income while developers from India are more likely to be categorised as low income.

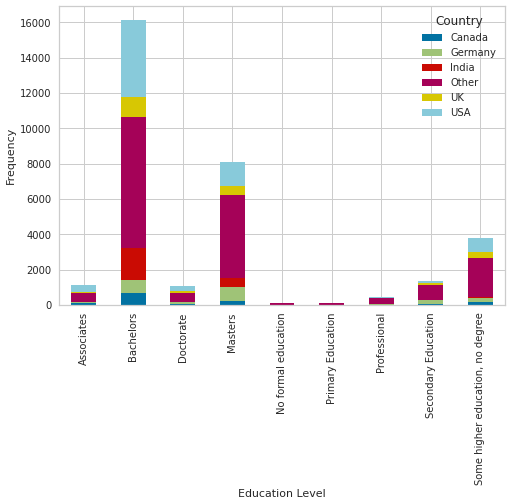
### Education Level

EdLevel is a categorical feature that describes the highest level of education earnt by each respondant. We can expect those who have a higher level of education to earn a higher amount.



From a barchart of the frequencies of each education level we can see that the majority of respondants have acquired a bachelors degree, and approximately half as many have a masters degree. A considerable amount completed some higher education but achieved no degree while a relatively similar number have secondary education, an associates degree, or a doctorate.

After plotting the education level against the average, median salary, we can see that while a doctorate encompasses a small proportion of all respondants, those with a dotorate tend to earn the most. While there are double the amount of respondants with a bachelors as with a masters, they earn a very similar amount. Those with no formal educaton also appear to earn a considerable amount, despite being the lowest frequency of respondants. It can be assumed that those with no formal education have an impressive self-taught portfolio. On average it appears that those who have earnt Higher Education earn more.

When comparing education level to country it can be seen that approximately half of respondants who have completed some form of higher education degree are from one of the top five most earning countries. The USA is the single country with the most degrees, encompassing almost a quarter of all bachelors degrees. India also has a relatively large proportion of bachelors degrees, despite being the lowest earning country in the top five. This suggests that the country has more of an impact than education level.

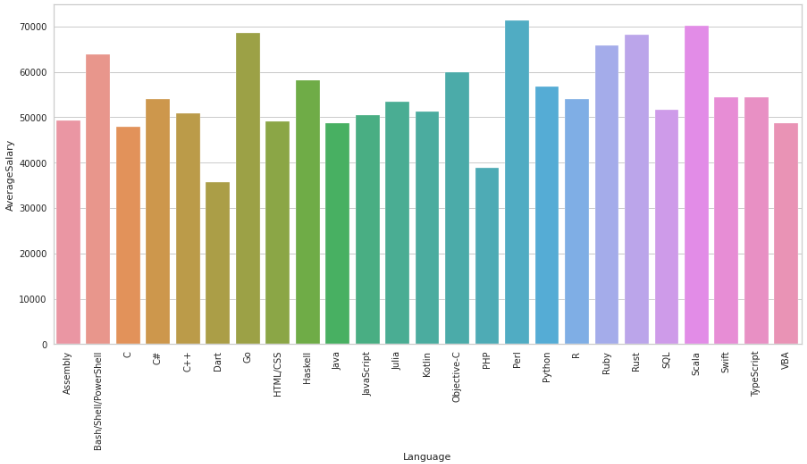
### Developer Type

DevType is a categorical feature that represents the type of programming work developers’ do. The feature has multiple responses per respondant, seperated out by a ‘;’, therefore in order to plot the feature in a graph correctly I need to separate out each possible response into its own data frame.

After plotting each developer role against the average median salary, we can see that a Site Reliability Engineer earns the most, with Senior Executive or VP close behind. at over 80,000 USD. This can be expected, as More senior roles will often earn more. Academic Researcher earns the least on average, at appproximately 40,000 USD. This shows that the type of work a developer does can have some impact on salary. The majority of developer roles, on average, earn between 40,000 and 100,000 USD, the majority appearing to fall inbetween the 40,000 and 60,000 USD bracket. We can see that some roles have more of an impact on salary than others, for example most Developer type roles have little variation between themselves, but earn considerably less than a DevOps Specialist for example.

### Languages Worked With

LanguageWorkedWIth is a categorical feature describing the programming languages developers work with most often. This feature also has multiple responses per respondant, seperated out by a ‘;’, therefore, like with the DevType feature, they must be seperated out again.

After plotting against the average salary, we can see high variations between different languages. For example python appears to be the highest earning, at approximately 70,000 USD, with Scala and GO close behind. The least earning languages appear to be PHP and Dart, at under 40,000 USD. This shows that the languages used do have an impact on salary, likely caused by industry demand.

# 3. Cluster Analysis

## Data Transformation and Normalisation

Before beginning the cluster analysis, I will be applying data transformation and normalisation to my selected fearures.

I have a variety of different features, such as numerical, ordinal, and nominal. I will need to convert the numerical features into categorical data first, then I will be able to encode all of the features using the correct encoder. Some of my categorical features also have multiple responses, so I will need to create new features through these. Finally, I will need to create taget features using the target variable, for splitting into clusters and performing machine learning algorithms for classification.

### Binning Numerical Features

The YearsCodePro feature is numerical data that ranges from 0 to 50. It is ordered, so it will be possible to encode this using an ordinal encoder. Firstly, I will need to create bins to distribute it into. These bins will be “Short”, “Medium”, “Long”, Very Long“, and “Extremely Long” relating to how long an individual has coded professionally. Considering the exploratory data analysis on this feature, 0 to 4 years will be short, 5 to 14 will be medium, 15 to 29 will be long, 30 to 39 will be very long, and 40 to 50 will be extremely long. I created a new feature with these called “Years\_bin”.

### Encoding Ordinal Features

I now had two ordinal features that could be encoded using an Ordinal Encoder: Years\_bin and EdLevel. I created an encoder for the Years and set the categories to the bin labels, then encoded them into ordered, numerical data from 0 to 4.

Next I created another ordinal encoder for the EdLevel feature, setting the categories to the levels of education, ordered by least to most educated. These were then encoded into ordered, numerical features ranging from 0 to 8.

### Encoding Nominal Features

I had three nominal, categorical features to encode using OneHotEncoder: Country, DevType, and LanguageWorkedWith. I was able to easily encode Country using an inbuilt Pandas function, creating a new column for each country.

As DevType and LanguageWorkedWith had multiple repsonses in a single entry, I had to manuallly iterate through them in order to split them by their delimiter, and create new features. While doing this I manually OneHotEncoded each column.

### Splitting Data Set

Next, I neede to split the data set by a target variable for cluster analysis and classification. I calculated the median of ConvertedComp, anything less than or equal to this would be classified as 0 meaning “Low income”, and above would be 1 or “High income”. I created a new feature called “Income” to store this data in. Here I also created the training and testing datasets that would be used for machine learning algorithms later.

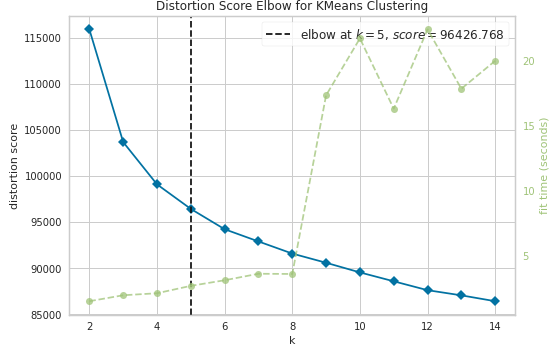
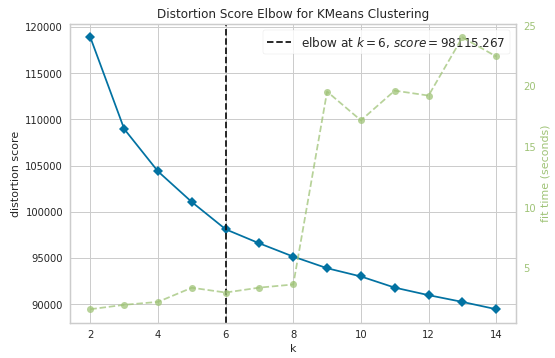
Finally I grouped the dataset by the Income feature, and created two new datasets: lowIncome and highIncome.

## Implementation of Cluster Analysis

### Kmeans Algorithm

I will be using the K-Means Clustering Algorithm to cluster the datasets. First, I will need to find the best number of clusters to use for each dataset. There are two methods I will be using: the Elbow Method in order to find the optimal number of clusters, and the Sillhoette Method to confirm.

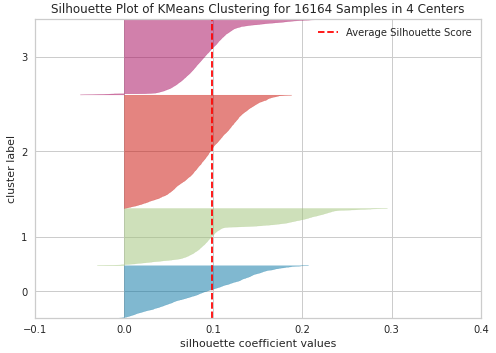
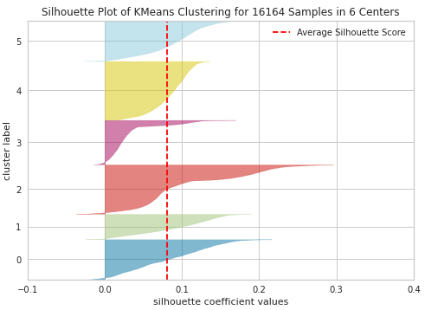
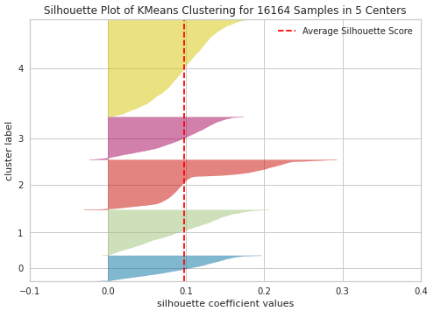
### Elbow Method

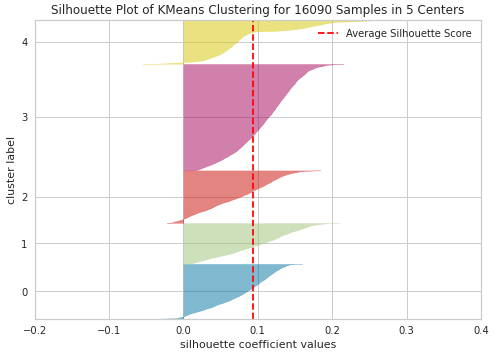
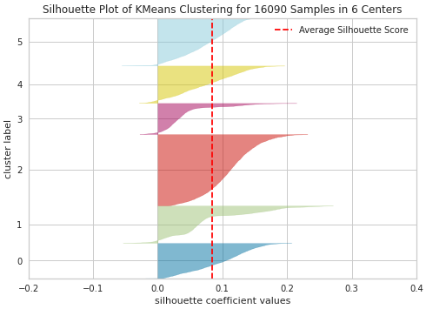
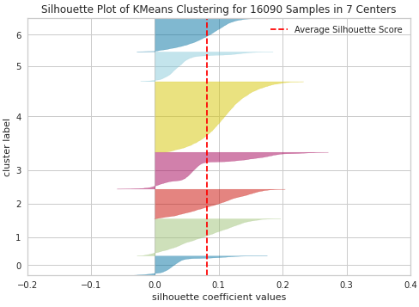
I implemented the Elbow method for both datasets, testing k for values between 2 and 15.

*(Low Income Left, High Income Right)*

Looking at the graohs produce, we can see that the optimal number of clusters for the low income dataset is 5, while the optimal number of clusters for the high income dataset is 6.

### Sillhouette Method

I then implemented the Sillhoette method to confirm the findings of the Elbow Method.

For the low income dataset, it can be confirmed that 5 clusters appears to be the optimal.

For the high income dataset, it appears that 6 clusters is the optimal number.

After having the number of clusters confirmed, I implemented the clustering algorrithm on each dataset, with the number of clusters I found; 5 fo rlow income and 6 for high income.

## Characteristics of Clusters

### Years Coding Professionally

Looking at the crosstabs for years coding professionally, we can see that in the low income dataset, there is a much higher proportion of people who have coded for a short amount of time, while the number of people who have coded for a long, very long, or extremely long time is much smaller. Whereas in the high income dataset, there is a much higher frequency of people who have coded for a medium or longer lenth of time. This can be expected as those who have been coding longer preofessionally have been in the industry longer, and have had more time to progresss.

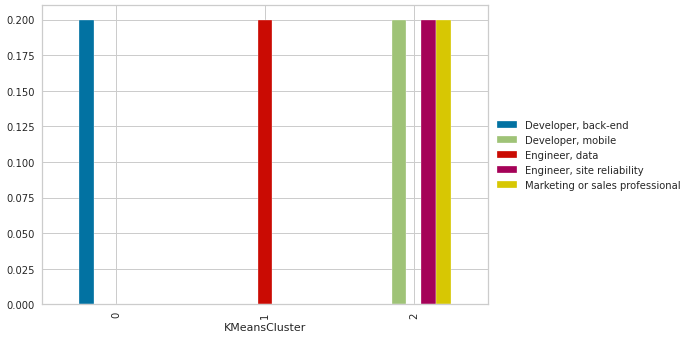
### Country

For the country, we can see that in the low income dataset there is a much higher proportion of people from India or other countries, with India being the highest out of the top five countries in almost every cluster. Conversly, in the high income dataset, we can see that there is a far greater proportion of people from the USA, and a slightly more significant number of people from the UK, Germany and Canada. India is the lowest in every cluster.

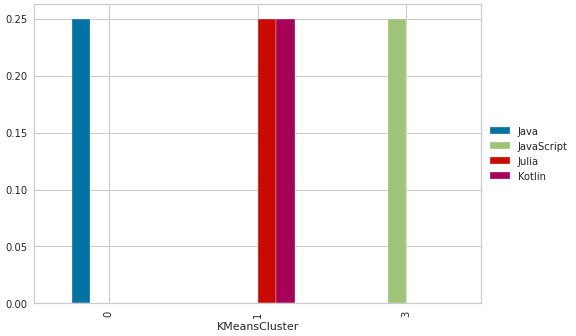
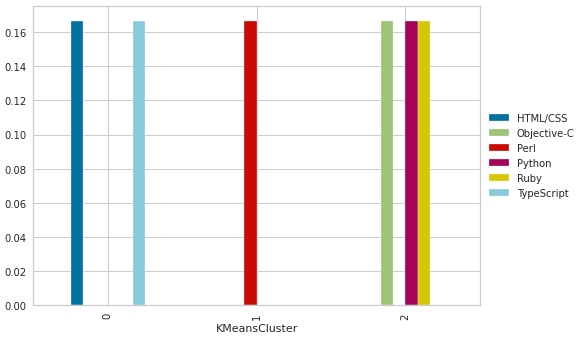
### Education Level

Looking at education level, we can see that there are similar proportions of people with similar education levels in both clusters. However there are slightly more people with higher levels of education in the high income cluster, as there are higher amounts of bachelors and masters degrees. There are also more people with doctorates in the high income dataset.

### Developer Type

When looking at the developer type features, we can see that the low income dataset has certain roles associated with it, while the high income dataset has other roles associated with it. It appears that embedded applications, front end, full stack, and games developers are more associated with low income, while back-end and mobile developers, data and site reliability engineers, and marketing roles are more associated with high income developers.

### Languages Worked With

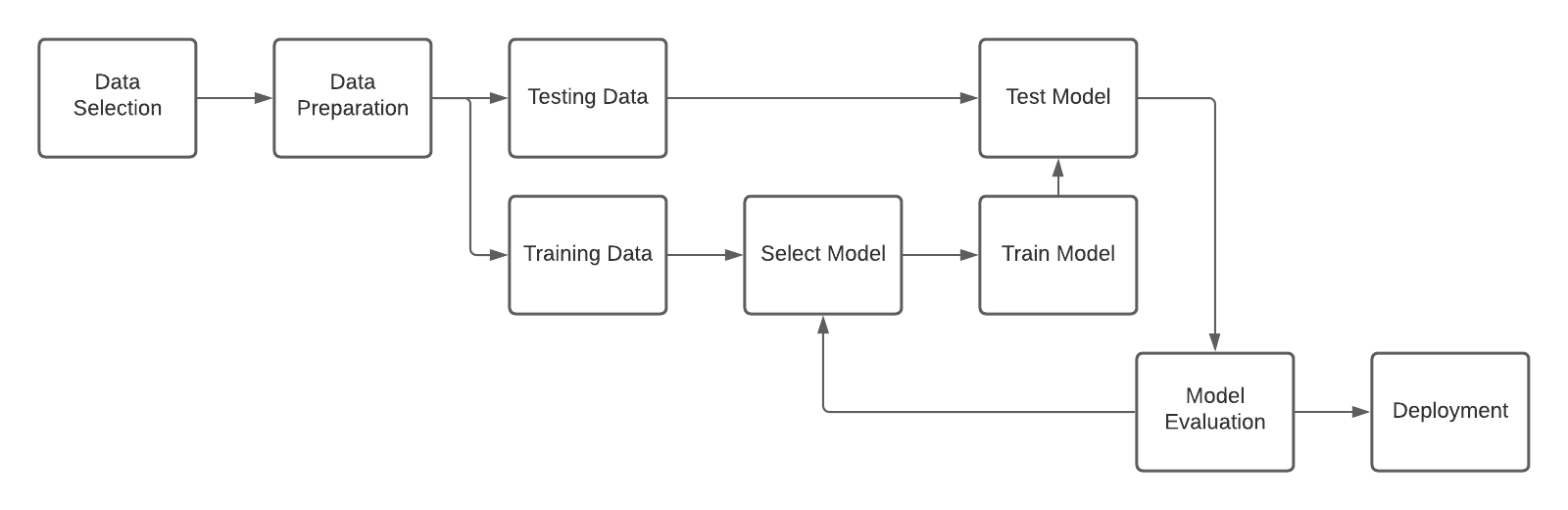
We can see a similar pattern when looking at languages worked with; some languages appear to be more strongly associated with low income developers, for example Java, Javascript, Julia and Kotlin, while other languages appear to be more associated with high income developers, such as HTML/CSS, Objective-C, Perl, Python, Ruby, and Typescript.

### Conclusion

We can see from the cluster analysis that the majority of the features selected appear to have a large impact on the salary of a developer, for example years coding professionally, country, developer type, and languages worked with. Other features have less of an impact but still have some, such as education level.

# 4. Machine Learning for Classification

## Workflow of Machine Learning

Throughout this project, I shall be following the CRISP-DM methodology. The machine learning workflow follows the same principles of this methodology.

I will be splitting the dataset into two; a training aand a testing dataset. The training dataset will be used to train the machine learning model on, so it will be able to correctly classify a developers’ predicted income based on their derails. The testing dataset will be used to test the model on an entirely new set of data. A number of different machine learning models will be used on the same dataset, and their accuracy will be compared for evaluation. Finally, the most suitable model will be chosen.

## Classification Methods

I will be using a wide number of classification algorithms, of differing levels in complexity, in order to successfully determine the most accurate model to use. I will be testing and evaluatiing each model, comparing their performance using a variety of performance metrics, such as accuracy scores and confusion matrices.

I will be using four classification algorithms on their own, and three methods of ensemble learning, in order to determine which is most accurate.

|  |  |
| --- | --- |
| Classification Algorithm | Algorithm Type |
| K-Nearest Neighbor | Single |
| Decision Tree | Single |
| Logistic Regression | Single |
| Artifical Neural Network | Single |
| Random Forest | Ensemble |
| Voting | Ensemble |
| AdaBoost | Ensemble |

## Parameter Setting

In addition to this, I will be running each model twice. On the first run I shall be evaluating the base model performance, and then I will be automatically tweaking the hyperparameters of each model using eaither GridSearch ot RandomizedSearch to find the most accurate combination of hyperparameters, to compare to the base model.

## Implementation of Machine Learning Models

### K-Nearest Neighbor

K-Nearest Neighbor is a supervised learning algorithm that is used for classification and regression. It works on the assumption that similar data is close to each other, employing distance-based functions to measure the simalarities or differences between two objects (Jiang, et al., 2007).

The algorithm requires the data be scaled to a range, so to begin with I will be applying a StandardScaler to the data.

#### Base Model

Next I will be implementing the base model of the algorithm, without tweaking any hyperparameters. We can see that both the training and testing accuracies are relatively high, above 80%.

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.867 | 0.813 |

#### Hyperparameter Tuned Model

After this I will be tweaking the hyperparameters of the model using GridSearchCV, by setting up a grid of the hyperparameters and their possible options, and testing each one, selecting the model with the highest accuracy.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Possible options | Description | Selected Value |
| Weights | Uniform, distance | Determines how each point is weighted. | Distance |
| Metric | Euclidean, manhattan, minkowski | Determines the distance formula to use | Manhattan |
| N\_neighbours | Range from 1 to 21 | Determins number of data points to create clusters | 20 |

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.997 | 0.826 |

We can see that the new model has a much higher train accuracy of 99.7% and a higher test accuracy of 82.6%. This is a 1.3% increase of the base models’ testing accuracy.

### Decision Tree

Decision Tree is a supervised, predictive modelling algorithm for classification. It works by classifying a set of data into branch-like segments in a tree data structure, consisting of root and leaf nodes (Song & Lu, 2015).

#### Base Model

Next I will be implementing the base model of the algorithm, without tweaking any hyperparameters. The training accuracy is very high at 99.7% and the testing accuracy is fairly lower, at 78.4%.

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.997 | 0.784 |

#### Hyperparameter Tuned Model

Next I will be using RandomizedSearch to automatically tune the hyperparameters, based on a grid.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Possible options | Description | Selected Value |
| Criterion | Gini, Entropy | Determines the quality of a split | Gini |
| Splitter | Best, random | Determines the strategy to choose a split at each node | Random |
| Max\_depth | Range from 1 to 21 | Determines maximum depth of a tree | 17 |

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.904 | 0.812 |

We can see that the new model has a slightly lower accuracy on the training data, of 90.4%, but has a higher accuracy on the testing data, of 81.2%, an increase of 2.6%.

### Logistic Regression

Logistic Regression is a classification algorithm that uses the mathmatical sigmoid function to classify data, producing a binary variable.

#### Base Model

I will be implementing the base model of the algorithm, without tuning the hyperparameters. The training and testing accuracies are very similar, and both are quite high, at above 85%.

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.862 | 0.855 |

#### Hyperparameter Tuned Model

Next I will be using GridSearch to automatically tune theh hyperparameters.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Possible options | Description | Selected Value |
| Solver | Newton-cg, lbfgs, liblinear, sag, saga | Algorithm to use in the optimization problem | Lbfgs |
| Penalty | L2 | Used to specify the norm used in the penalization. Due to the constraints of the Solvers, only l2 is applicable | L2 |
| C | 100, 10, 1.0, 0.1, 0.01 | Inverse of regularisation strength, smaller values specify stronger regularisation | 10 |

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.863 | 0.855 |

The training and testing accuracies of the hyperparameter tuned model are very similar to each other, and are both significantly accurate at 86.3% and 85.5%. Additionally, there is negligable difference between the base model and the hyperparameter tuned model, with no difference in testing accuracy and a 0.1% increase in training accuracy.

### Artifical Neural Network

An artificial neural network is a model inspired by biological neural networks, consisting of a collection of layers composed of nodes, or perceptrons, that are connected by weights (Wang, 2003).

The algorithm requires the data be scaled to a range, so to begin with I will be applying a StandardScaler to the data.

#### Base Model

Next I will be implementing the base model of the algorithm, without tuning the hyperparameters. The training accuracy is very high at 97.8% and the testing accuracy is fairly high at 80.7%.

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.978 | 0.807 |

#### Hyperparameter Tuned Model

I will be using RandomizedSearch to find the optimal hyperparameters for this model, as it has many different parameters to tune.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Possible options | Description | Selected Value |
| Hidden\_layer\_sizes | (3), (3, 2), (4) | The number of neurons in the hidden layer, and the size of the hidden layer | (3) |
| Activation | Tanh, relu | Activation function for the hidden layer | Relu |
| Solver | Sgd, adam | The solver for the weight optimization | Sgd |
| Alpha | 0.0001, 0.005 | L2 penalty parameter | 0.005 |
| Learning\_rate | Constant, adaptive | The intial learning rate used | Adaptive |

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.863 | 0.853 |

After tuning the hyperparameters, the training accuracy and testing accuracy are very similar, at 86.3% and 85.3%; a difference of 1%. Additionally, the training accuracy is much lower than the base model but the testing accuracy is much higher, at 85.3%, a difference of 4.6%.

## Implementation of Ensemble Learning

### Random Forest

Random forest is am ensemble learning method for classification, that works by combining groups of decision trees together (Pal, 2007).

#### Base Model

I will be implementing the base model of the Random Forest algorithm. The training accuracy was very high at 99.7%, and the testing accuracy is slightly lower at 84.1%.

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.997 | 0.841 |

#### Hyperparameter Tuned Model

I will be using RandomizedSearch to find the optimal hyperparameters for the model.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Possible options | Description | Selected Value |
| Max\_features | Auto, sqrt | The number of features | Sqrt |
| Max\_depth | Range from 1 to 21 | The maximum depth of the tree | 15 |
| N\_estimators | 10, 50, 100, 500, 1000, 1500, 2000 | The number of trees in the forest | 500 |

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.921 | 0.841 |

After tuning the hyperparameters the model still has a high training and testing accuracy, at 92.1% and 84.1%. However the training accuracy is slightly lower than the base model, and the testing accuracy is the same.

### Voting

The Voting method is a classification algorithm which combines several different classification algorithms together, and works by taking a weighted vote of their predictions to produce a prediction (Diettrich, 2000).

I will be using the top three performing models so far, Logistic Regression, K-Nearest Neighbor, and Artificial Neural Network.

#### Base Model

I will now be implementing the base model for the Voting method. The training and testing accuracies are quite similar, at 86.7% and 85.5%.

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.867 | 0.855 |

#### Hyperparamter Tuned Model

I will use GridSearch to find the optimal hyperparameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Possible options | Description | Selected Value |
| Voting | Hard, soft | The voting method used | Hard |

|  |  |
| --- | --- |
| Train Accuracy | Test Accuracy |
| 0.867 | 0.855 |

The training and testing accuracies are the exact same as the base model.

### AdaBoost

AdaBoost is an adaptive boosting algorithm that is used in conjunction with other learning algorithms to improve performance, it works by training weak learners sequentially by more weighting on unclassified data (Schapire, 2013).

I will be using the AdaBoost classifier combined with a decision tree and logistic regression.

|  |  |  |
| --- | --- | --- |
| Model | Train Accuracy | Test Accuracy |
| Decision Tree | 0.997 | 0.801 |
| Logistic Regression | 0.861 | 0.854 |

After implementing the model we can see that for the decision tree, the training accuracy is very high at 99.7% wheras the testing accuracy is significantly lower at 80.1%. For Logistic Regression, while the training accuracy is lower than the decision tree at 86.1%, the testing accuracy is much higher at 85.1%. There is also less difference between the training and testing accuracies.

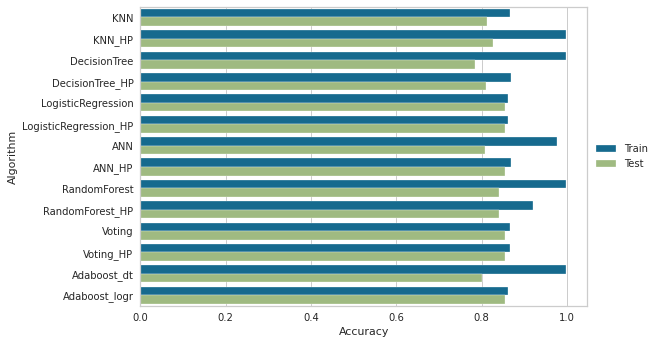
# 5. Evaluation

After implementing all of the machine learning models, I will now compare their performance using a number of performance metrics, namely accuracy scores and confusion matrices.

## Accuracy Score

|  |  |  |
| --- | --- | --- |
| Model | Train Accuracy | Test Accuracy |
| K-Nearest Neighbour | 0.867 | 0.813 |
| K-Nearest Neighbour HP | 0.997 | 0.826 |
| Decision Tree | 0.997 | 0.784 |
| Decision Tree HP | 0.870 | 0.810 |
| Logistic Regression | 0.862 | 0.855 |
| Logistic Regression HP | 0.863 | 0.855 |
| Artificial Neural Network | 0.978 | 0.807 |
| Artificial Neural Network HP | 0.868 | 0.854 |
| Random Forest | 0.997 | 0.841 |
| Random Forest HP | 0.921 | 0.841 |
| Voting | 0.867 | 0.855 |
| Voting HP | 0.867 | 0.855 |
| AdaBoost Decision Tree | 0.997 | 0.801 |
| AdaBoost Logistic Regression | 0.861 | 0.854 |

From these accuracy scores we can identify that all of the models had a relatively high training and testing accuracy, with only one falling beneath 80%. The higherst training accuracy appears to be 99.7%, while the highest testing accuracy is 85.5%. All of the models performed relatively similarly on testing data, with the highest range in testing accuracies being 7.1%. Meanwhile the training data had a much higher range, of 13.1%. The lowest trainiing accuracy, which is 86.3%, is still higher than the higherst testing accuracy. None of the models performed at 100% accuracy, and none of the testing data achieved over 90% accuracy.

We can see from this graph that, throughout all of the models, the trainign accuracy was always higher than the testing accuracy. We can also see that all of the models had relatively similar performances. We can also see that throughout almost all of the models, the hyperparameter tuned models had a higher testing accuracy. Therefor tuning the hyperparameters has had a noticable impact on imrpoving the models’ performance on unseen data.

## Confusion Matrix

|  |  |  |
| --- | --- | --- |
| KNN | KNN HP | Decision Tree |
|  |  |  |
| Decision Tree HP | **Logistic Regression** | **Logistic Regression HP** |
|  |  |  |
| ANN | **ANN HP** | **Random Forest** |
|  |  |  |
| Random Forest HP | **Voting** | **Voting HP** |
|  |  |  |
| AdaBoost Decision Tree | **AdaBoost Logistic Regression** |  |
|  |  |  |

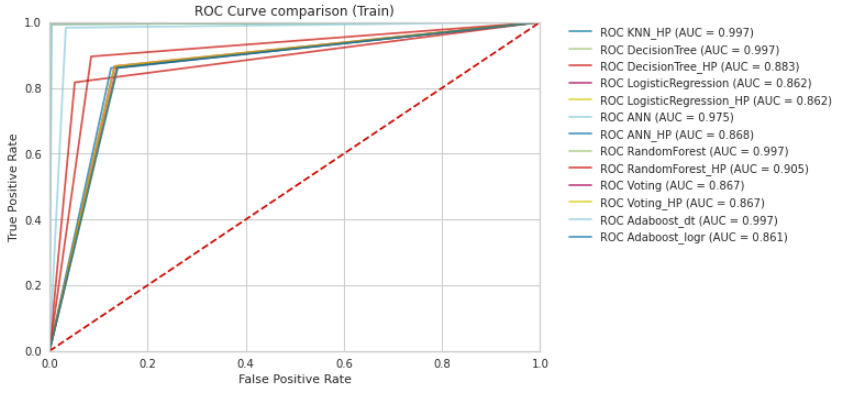
Confusion Matrices allow us to visualise exavtly what number of accurate prediction a model as made, and what number of inaccurate predictions a model has made.

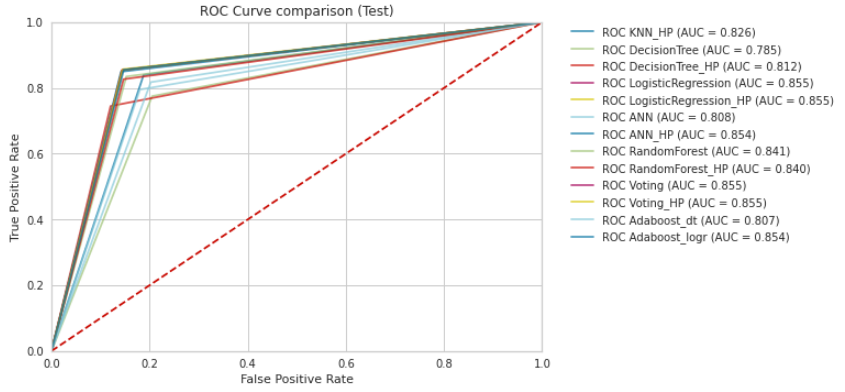
From the confusion matrices we can see that most of the models were able to accurately predict almost all of the incomes of the developers, with the slight exception of the decision tree and the hyperparameter tuned decision tree, the random forest and hyperparameter tuned random forest, and the adaboost decision tree. These all have marginally lower accurate predictions, but have still been overall successful with their predictions.

None of the models have failed, as they have all had high accuracies and have been able to correctly classify an overwhelming majority of the data. All of the models have a relatively similar success rate between classifying high income developers and low income developers.

## ROC Curve

An ROC curve is a performance meric that illustrates how well a classification model is able to correctly label data. It is plotted using two values: True Positive Rate and False Positive Rate, which correspond to how much data has been correctly and incorrectly classified. This produces the AUC value, the closer this value is to 1, the better the performance of the model.

I have plotted all of the models’ training data scores and testing data scores on ROC Curves, in order to compare. We can see that, overall, the training dataset has a much higher AUC number across all models, with some models nearing 1, such as K-Nearest Neighbour with hyperparameter tuning, decision tree, and AdaBoost with decision tree. The lowest AUC value is 0.861. Meanwhile, the testing data has a lower AUC value on average, with the highest being 0.855. All of the models have an AUC value of over 0.5, which shows that they all have a higher likelihood of producing a true positive clasification rather than a false positive.



# 6. Discussion and Conclusion

## Summary

The goal of this task was to apply machine learning and data analysis on the “Stack Overflow Developer Survey 2020”, to follow the CRISP-DM methodology in order to perform exploratory data analysis and cluster analysis on the data, to select features and preprocess them, and to build a number of machine learning models in order to accurately classify and predict the common features that would make a developer have a high or a low income.

I was able to successfully perform these tasks. I was able to identify and select the most important features relating to a developers’ salary through research, and perform exploratory data analysis on these features, using a number of graphs to identify possible relationships. I then performed data preprocessing and feature creation, correctly being able to normalise and encode the data into a format that I would be able to use for cluster analysis an**No index entries found.**d model implementation. I was then able to successfully perform cluster analysis on the data, and gained new insights into the various relationships between the features I had selected to the target variable and their links within each other. Finally, I was able to implement a numebr of classification algorithms, as well as using ensemble learning to combine them, in order to produce a wide range of performance metrics to compare each algorithms’ performance on the data set.

Overall I believe that my implementation of this project has been a success, and I have been able to correctly identify the most accurate machine learning classifcation models on the features I had selected from the dataset.

## Insights

Throughout the project I have gained numerous insights into the factors that affect a developers salary, such as country of residence, education level, years spent coding professionally, languages worked with, and development type. I have been able to visualise the impacts of these variables and come to conclusions about how they affect thesalary of a developer and how significant their imapct is. Through this understanding I have been able to correctly implement machine learning for classification algorithms that are able to predict and classifywhether a developer is high or low income, based on these factors. These models could be applied to real world scenarios.

## Understanding

Throughout my undertaking of this project, and the Machine Learning for Data Analytics module, I have been able to learn and solidify a myriad of core skills, technologies, and methodologies that are vital for the process of machine learning.

For example I have learnt to follow the CRISP-DM methodology in order to produce machine learning models, I have learnt a variety of python libraries for machine learning and plotting such as pandas, sklearn, matplotlib, and seaborn. I have learnt how to conduct a proper and thorough exploratory data analysis and cluster analysis using a variety of different graphs and plotting methods. I have gained insights into various machine learning models and techniques, such as hyperparameter tuning of models such as K-Nearest Neighbour and Logistic Regression. I have learnt to combine these models together in ensemble learning. Finally, I have learnt to critically analyze the performance of these models using a variety of metrics such as accuracy scores, confusion matrices, and ROC curves. I have further developed my understanding of these techniques through analyzing other kaggle notebooks exploring new datasets, and gained insights into new techniques that have not been taught. Finally, my understanding of the world of machine learning has been solidified, and I can continue to apply these skills in future work.

# References

Diettrich, T. G., 2000. Ensemble Methods in Machine Learning. *International Workshop on Multiple Classifier Systems,* 1857(1), pp. 1-15.

European Commission, 1997. *CORDIS EU Research Results.* [Online]   
Available at: https://cordis.europa.eu/project/id/25959  
[Accessed 05 05 2021].

Forbes, 2015. *What IT Needs To Know About The Data Mining Process.* [Online]   
Available at: https://www.forbes.com/sites/metabrown/2015/07/29/what-it-needs-to-know-about-the-data-mining-process/#2065f3a3515f  
[Accessed 05 05 2021].

Jiang, L., Cai, Z., Wang, D. & Jiang, S., 2007. Survey of Improving K-Nearest-Neighbor for Classification. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery,* 1(1), pp. 679-683.

OECD, 2021. *OECD Stats.* [Online]   
Available at: https://stats.oecd.org/Index.aspx?DataSetCode=RMW  
[Accessed 05 04 2021].

Pal, M., 2007. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing ,* 26(1), pp. 27-222.

Schapire, R. E., 2013. Explaining AdaBoost. *Empirical Inference,* 5(1), pp. 37-52.

Song, Y. & Lu, Y., 2015. Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry,* 27(2), pp. 130-135.

Stack Overflow, 2020. *Stack Overflow Blog.* [Online]   
Available at: https://stackoverflow.blog/2020/02/12/stack-overflow-salary-reports-are-out-now-how-does-your-company-compare/  
[Accessed 04 05 2021].

Stack Overflow, 2020. *Stack Overflow Insights.* [Online]   
Available at: https://insights.stackoverflow.com/survey/2020  
[Accessed 05 04 2021].

Wang, S., 2003. Artificial Neural Network. *Interdisciplinary Computing in Java Programming,* 743(1), pp. 81-100.

Wirth, R. & Hipp, J., 2000. *CRISP-DM: Towards a standard process model for data mining.* London, Springer-Verlag.

World Health Organisation, 2020. *World Health Organisation.* [Online]   
Available at: https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200205-sitrep-16-ncov.pdf?sfvrsn=88fe8ad6\_2  
[Accessed 05 04 2021].

# Appendix

## Survey

|  |  |
| --- | --- |
| Column | QuestionText |
| 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? |
| Age | What is your age (in years)? If you prefer not to answer, you may leave this question blank. |
| Age1stCode | At what age did you write your first line of code or program? (e.g., webpage, Hello World, Scratch project) |
| CompFreq | Is that compensation weekly, monthly, or yearly? |
| CompTotal | What is your current total compensation (salary, bonuses, and perks, before taxes and deductions), in `CurrencySymbol`? 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. |
| ConvertedComp | Salary converted to annual USD salaries using the exchange rate on 2020-02-19, assuming 12 working months and 50 working weeks. |
| Country | Where do you live? |
| CurrencyDesc | Which currency do you use day-to-day? If your answer is complicated, please pick the one you're most comfortable estimating in. |
| CurrencySymbol | Which currency do you use day-to-day? If your answer is complicated, please pick the one you're most comfortable estimating in. |
| DatabaseDesireNextYear | Which 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.) |
| DatabaseWorkedWith | Which 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.) |
| DevType | Which of the following describe you? Please select all that apply. |
| EdLevel | Which of the following best describes the highest level of formal education that youâ€™ve completed? |
| Employment | Which of the following best describes your current employment status? |
| Ethnicity | Which of the following describe you, if any? Please check all that apply. If you prefer not to answer, you may leave this question blank. |
| Gender | Which of the following describe you, if any? Please check all that apply. If you prefer not to answer, you may leave this question blank. |
| 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? |
| JobSat | How satisfied are you with your current job? (If you work multiple jobs, answer for the one you spend the most hours on.) |
| JobSeek | Which of the following best describes your current job-seeking status? |
| LanguageDesireNextYear | Which 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.) |
| LanguageWorkedWith | Which 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.) |
| MiscTechDesireNextYear | Which 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 framework and want to continue to do so, please check both boxes in that row.) |
| MiscTechWorkedWith | Which 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 framework and want to continue to do so, please check both boxes in that row.) |
| NEWCollabToolsDesireNextYear | Which collaboration 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 worked with the tool and want to continue to do so, please check both boxes in that row.) |
| NEWCollabToolsWorkedWith | Which collaboration 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 worked with the tool and want to continue to do so, please check both boxes in that row.) |
| NEWDevOps | Does your company have a dedicated DevOps person? |
| NEWDevOpsImpt | How important is the practice of DevOps to scaling software development? |
| NEWEdImpt | How important is a formal education, such as a university degree in computer science, to your career? |
| NEWJobHunt | In general, what drives you to look for a new job? Select all that apply. |
| NEWJobHuntResearch | When job searching, how do you learn more about a company? Select all that apply. |
| NEWLearn | How frequently do you learn a new language or framework? |
| NEWOffTopic | Do you think Stack Overflow should relax restrictions on what is considered â€œoff-topicâ€? |
| NEWOnboardGood | Do you think your company has a good onboarding process? (By onboarding, we mean the structured process of getting you settled in to your new role at a company) |
| NEWOtherComms | Are you a member of any other online developer communities? |
| NEWOvertime | How often do you work overtime or beyond the formal time expectation of your job? |
| NEWPurchaseResearch | When buying a new tool or software, how do you discover and research available solutions? Select all that apply. |
| NEWPurpleLink | You search for a coding solution online and the first result link is purple because you already visited it. How do you feel? |
| NEWSOSites | Which of the following Stack Overflow sites have you visited? Select all that apply. |
| NEWStuck | What do you do when you get stuck on a problem? Select all that apply. |
| OpSys | What is the primary operating system in which you work? |
| OrgSize | Approximately how many people are employed by the company or organization you currently work for? |
| PlatformDesireNextYear | Which platforms 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 platform and want to continue to do so, please check both boxes in that row.) |
| PlatformWorkedWith | Which platforms 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 platform and want to continue to do so, please check both boxes in that row.) |
| PurchaseWhat | What level of influence do you, personally, have over new technology purchases at your organization? |
| Sexuality | Which of the following describe you, if any? Please check all that apply. If you prefer not to answer, you may leave this question blank. |
| SOAccount | Do you have a Stack Overflow account? |
| SOComm | Do you consider yourself a member of the Stack Overflow community? |
| 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. |
| SOVisitFreq | How frequently would you say you visit Stack Overflow? |
| SurveyEase | How easy or difficult was this survey to complete? |
| SurveyLength | How do you feel about the length of the survey this year? |
| Trans | Are you transgender? |
| UndergradMajor | What was your primary field of study? |
| WebframeDesireNextYear | Which 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.) |
| WebframeWorkedWith | Which 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.) |
| WelcomeChange | Compared to last year, how welcome do you feel on Stack Overflow? |
| WorkWeekHrs | On average, how many hours per week do you work? Please enter a whole number in the box. |
| YearsCode | Including any education, how many years have you been coding in total? |
| YearsCodePro | NOT including education, how many years have you coded professionally (as a part of your work)? |