**1. Consider the data adult.csv**

**A picture containing graphical user interface

Description automatically generated**

**2. Formulate a Hypothesis**

H0: There is no association between the two categorical variables (gender and income)

H1: There is some association between the two variables gender and income.

#In other words H0 is trying to say that, Gender does not impact an individual's income. Income level cannot determine if an individual is male or female.

Since we have 2 categorical variables, we will be using Chi-square test of independence to test our hypothesis towards the end. We basically wish to see if gender has an impact on the income of an individual.

Text

Description automatically generated

**3. Analyze your Data.**

1. **Exploratory Data Analysis:**

Firstly, we look at the columns and see what each of them mean in context.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| age | The age of the individual in years (numeric) |
| workclass | The type of employer for the individual (categorical: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked) |
| fnlwgt | The final weight (sampling weight) associated with the individual (numeric) |
| education | The highest level of education achieved by the individual (categorical: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool) |
| education-num | The highest level of education achieved by the individual represented as a numeric value (numeric) |
| marital-status | The marital status of the individual (categorical: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse) |
| occupation | The type of job held by the individual (categorical: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces) |
| relationship | The relationship status of the individual (categorical: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried) |
| race | The race of the individual (categorical: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black) |
| sex | The gender of the individual (categorical: Female, Male) |
| capital-gain | The capital gains of the individual (numeric) |
| capital-loss | The capital losses of the individual (numeric) |
| hours-per-week | The number of hours worked per week by the individual (numeric) |
| native-country | The country of origin of the individual (categorical) |
| income | Whether the individual earns more than $50K per year (categorical: <=50K, >50K) |

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

**Checking for imbalance in dataset.**

Chart, bar chart

Description automatically generated

Text

Description automatically generated

**Check for duplicates.**

Text

Description automatically generated with medium confidence

**Age**

Chart, histogram

Description automatically generated

**Work-Class**

Chart

Description automatically generated

**Education-level**

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated with medium confidence

**Age and Income**

Chart, box and whisker chart

Description automatically generated

**Correlation Analysis**

**Graphical user interface, application, Teams

Description automatically generated**

**A picture containing application

Description automatically generated**

**Hours-per-week**

**Chart, histogram

Description automatically generated**

**4. Preprocess your data**

**Graphical user interface, text, application, email

Description automatically generated**

**A screenshot of a computer

Description automatically generated with medium confidence**

**5. Will you need to Vectorize?**

Yes, we will need to vectorize the categorical variables in the “adult.csv” dataset to apply Naive Bayes Classifier. Naive Bayes assumes that the features (i.e., columns) in the dataset are numerical, so you will need to convert the categorical variables into numerical features. Currently, we are using a process called one-hot encoding.

One-hot encoding creates a new binary column for each category in a categorical variable. For example, the “workclass” column has eight categories: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, and Never-worked. To one-hot encode this column, you would create eight new binary columns, one for each category, and assign a value of 1 to the appropriate column for each row. For example, if an individual had a “workclass” value of “State-gov”, their one-hot encoded values would be 0, 0, 0, 0, 0, 1, 0, 0.

We are using scikit-learn’s OneHotEncoder functions to perform one-hot encoding on the categorical variables in the dataset. Once the data has been one-hot encoded, we can use it as input to the Naive Bayes algorithm.

Label encoding is another way to encode categorical variables as numerical values, but it is not recommended for use with Naive Bayes, especially in cases where the categorical variable has more than two categories.

Label encoding assigns a unique numerical value to each category in a categorical variable, starting from 0. For example, in the “workclass” column of the “adult.csv” dataset, label encoding might assign the values 0-7 to the eight categories in the column. However, this creates an inherent ordering of the categories, which may not always be appropriate or accurate. In the case of the “workclass” column, there is no meaningful ordering to the categories, so label encoding would not be appropriate.

**Naive Bayes assumes that the features are independent, which means that the encoding of the categories should not introduce any relationships or patterns in the data that do not already exist. One-hot encoding avoids this issue by creating a separate binary feature for each category, which ensures that there is no inherent ordering or relationship between the categories.**

Therefore, for categorical variables with more than two categories, one-hot encoding is typically the preferred method for encoding categorical variables in Naive Bayes.

**Reference:** [**https://medium.com/acing-ai/why-do-we-need-one-hot-encoding-7bcb456d49df**](https://medium.com/acing-ai/why-do-we-need-one-hot-encoding-7bcb456d49df)

One-hot encoding can be taken as a type of vectorization. One-hot encoding is one of the most commonly used techniques for vectorizing categorical data, where each category is represented as a binary vector with a 1 in the corresponding category and 0s elsewhere.Other techniques include label encoding, binary encoding, and count encoding, among others. These techniques differ in how they represent the categories as numerical values.

**6. Use Naive Bayes model**

**Graphical user interface, text, application, email

Description automatically generated**

**7. Perform your Train and Test Split**

Graphical user interface, text, application

Description automatically generated

**9. Now use your test data to compute the accuracy of the model**

**Graphical user interface, text, application, email

Description automatically generated**

**10. Can it be improved? Explain how.**

**Common Methods to Improve our Model Performance:**

**1. Removing Irrelevant Columns based on research, domain understanding and preliminary EDA.**

As per preliminary analysis, we can see that the attributes education and education-num refer to the same thing that is the highest level of education attained by an individual. So, to improve the performance of our model we will remove the education attribute.

Secondly, there is an attribute “fnlwgt” which refers to the final weight (sampling weight) associated with the individual (numeric in nature).

Although, the attribute “fnlwgt” might have some use while doing some statistical analysis with the dataset. It does not have much relevance when it comes to building a predictive model/classifier. So, to improve model performance we have decided to remove this attribute from our model. (this we decided based on research and domain understanding.)

Table

Description automatically generated with medium confidence

**Accuracy – Without removal**

**Text, letter

Description automatically generated**

**Accuracy – With removal**

Graphical user interface, text, application

Description automatically generated

1. **Principal Component Analysis – For dimensionality reduction**

This is a linear method that transforms the original features into a new set of uncorrelated features called principal components. The principal components capture the maximum variance in the data and are used to reduce the dimensionality of the data.

Other methods for dimensionality reduction: Linear Discriminant Analysis (LDA), t-distributed Stochastic Neighbor Embedding (t-SNE), Random Projection, etc.

1. **Scaling the data for better model performance.**

Scaling is important because many machine learning algorithms assume that the input features are on a similar scale, and failure to scale the features can result in poor model performance. Scaling the data involves transforming the features so that they have a similar scale and range. Common scaling techniques include standardization and normalization. **Although, is a marginal difference, we still see some improvement using the standard scaler.**

Graphical user interface, text, application

Description automatically generated

**Accuracy – Before Scaling Data**

**Graphical user interface, text, application, email

Description automatically generated**

**Accuracy – After Scaling Data**

Graphical user interface, text, application

Description automatically generated

1. **Looking at other measures of model evaluation.**

Another method to improve performance could be to change the way we are computing the model performance (change in perspective). Currently, we are computing accuracy and taking it as the basis for comparing and evaluating alternative models. We could also use other methods as mentioned below to compare alternative models:

* Log Loss
* ROC Curve and AUC
* FI Score
* Precision
* Recall

Graphical user interface, text, application

Description automatically generated

1. **Changing Train-Test Split Ratio –** shows a slight increase in testing and training accuracy as shown below.

**Accuracy with split ratio 80-20 (Before)**

Graphical user interface, text, application, email

Description automatically generated

**Accuracy with split ratio 70-30 (After)**

**Graphical user interface, text, application, chat or text message, email

Description automatically generated**

**We can also note that changing split ratio from 80-20 to 70-30 also slightly reduces the difference between accuracy of our training and testing data as shown in the image below:**

**Graphical user interface, text, application, email

Description automatically generated**

1. **Applying k-fold Cross validation**

k-fold cross validation is a technique used to improve model performance by estimating the performance of a machine learning model. It is a resampling technique where the original dataset is randomly split into k equally sized parts or "folds". The model is trained on k-1 folds and evaluated on the remaining fold. This process is repeated k times, each time using a different fold for evaluation, and the results are averaged to provide an estimate of the model's performance.

By using k-fold cross validation, we can obtain a more reliable estimate of the performance of our model compared to using a single train-test split. This is because each data point in the dataset is used for testing exactly once, which reduces the variance of the estimate. Additionally, k-fold cross validation allows us to make use of all available data for training and testing, which can lead to better model performance**.**

**In summary, the initial model performance we observed** (before applying K-fold Cross Validation, before removal of irrelevant columns and scaling of features)

**Text, letter

Description automatically generated**

**After K-fold Cross Validation removal of irrelevant columns, scaling features.**

Graphical user interface, text, application, email

Description automatically generated

1. **Outlier Detection & Removal (based on judgement)**

Outliers can have a significant impact on model accuracy, as they can skew the data and lead to inaccurate predictions. **We need to be careful while removing outliers. Not all outliers are influential on the model. Some of them contain important information. Sometimes, we may fall into the trap of removing important information just to improve accuracy or to prove our hypothesis.** There are several ways to detect and handle outliers in a dataset:

* Visualize the data: Visualizing the data can help identify potential outliers. Box plots, scatter plots, and histograms are common visualization tools that can be used to identify outliers.
* Use statistical methods: Statistical methods such as z-score or Tukey's method can be used to identify outliers based on the standard deviation of the data.
* Remove outliers: Once outliers have been identified, they can be removed from the dataset. However, it is important to be cautious when removing outliers, as they may contain important information. Removing too many outliers can lead to underfitting.
* Transform the data: Transforming the data can help normalize the distribution and reduce the impact of outliers. Common transformations include log transformation or power transformation.

By detecting and handling outliers in the dataset, we can improve the accuracy of the model by reducing the impact of these extreme values on the predictions. However, it is important to carefully consider the impact of outlier removal or transformation on the overall dataset and the model performance.

1. **Based on the dataset consider using alternative classifiers**

There are several other classifiers that can be applied on the adult.csv dataset, including:

* Logistic Regression
* Decision Trees
* Random Forests
* Support Vector Machines
* K-Nearest Neighbors
* Gradient Boosting
* Neural Networks

Each of these classifiers has its own advantages and disadvantages and may perform better or worse depending on the specific dataset and problem. It is often recommended to try multiple classifiers and compare their performance using metrics such as accuracy, precision, recall, and F1-score.