KNN REPORT

1. Describe the method that you use?

KNN (K - Nearest Neighbours) is one of many algorithms used in data mining and machine learning, it's a classifier algorithm where the learning is based "how similar" is a data (a vector) from other. It stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).
A case is classified by a majority vote of its neighbours, with the case being assigned to the class most common amongst its K nearest neighbours measured by a distance function. We can measure the distance between the new cases and available cases using the following formulas:

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

The pseudo code for KNN algorithm is as follows:

We can implement a KNN model by following the below steps:

1. Load the data
2. Initialise the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points
    1. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
    2. Sort the calculated distances in ascending order based on distance values
    3. Get top k rows from the sorted array
    4. Get the most frequent class of these rows
    5. Return the predicted class

The advantages of using KNN algorithm are:

- The algorithm is simple and easy to implement.
- There's no need to build a model, tune several parameters, or make additional assumptions.
- The algorithm is versatile. It can be used for classification, regression, and search.

The disadvantages of using KNN algorithm is:

- The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

2. Describe the dataset, and explain how you selected the two attributes?

The census dataset consists of 16 different attributes among which one of it is a class attribute which defines the target value for any given row in the dataset. The data set contains the following attributes and its corresponding datatypes.

Date	object
Age	int64
WorkClass	object
fnlwgt	int64
education	object
education-num	int64
marital-status	object
occupation	object
relationship	object
race	object
gender	object
capital-gain	int64
capital-loss	int64
hours-per-week	int64
native-country	object
class	object

Feature Selection is one of the core concepts in training the model which hugely impacts the performance of your model. The data features that you use to train your models have a huge influence on the performance you can achieve.

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

Benefits of performing feature selection before modelling your data?

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
- Improves Accuracy: Less misleading data means modelling accuracy improves.
- Reduces Training Time: fewer data points reduce algorithm complexity and algorithms train faster.

Here we do not consider the object datatypes as a feature to train the model as it would require for us to transform every object type to int or Boolean type which is understood by the model. Hence, we choose only the int datatypes.

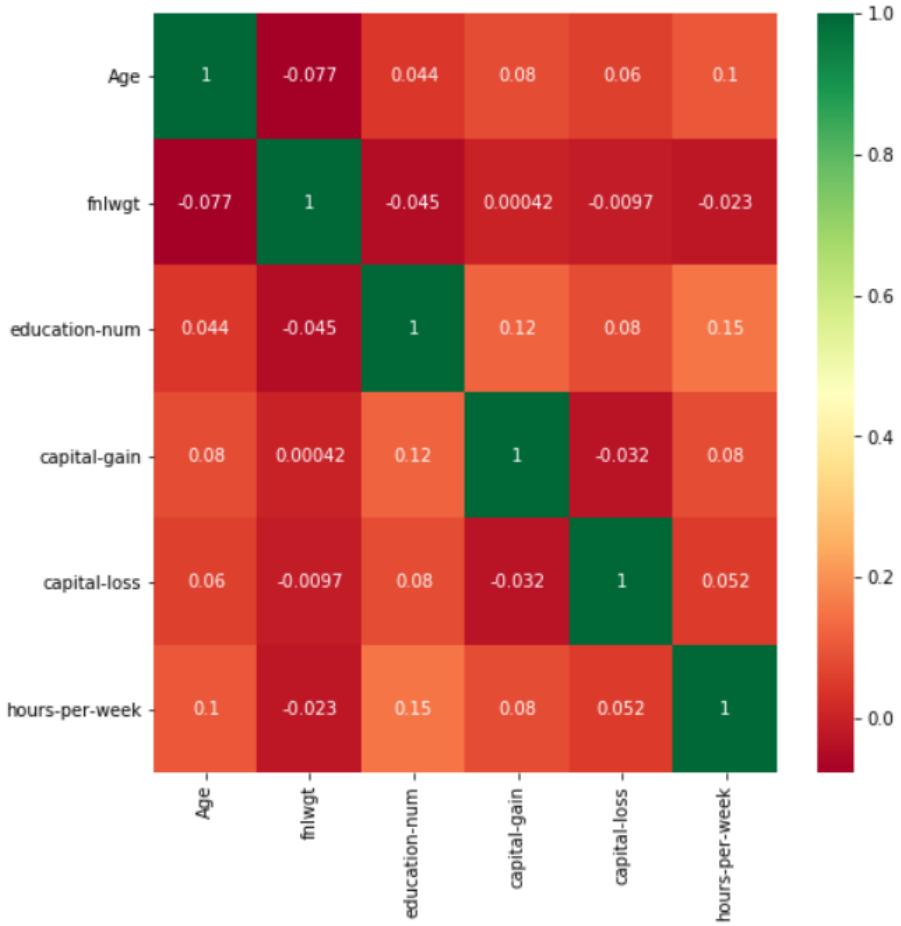
We are going to select the features which play an important role in building the model. We are selecting the features by examining the co-relation between them. The features which have the highest co-relation with each other are selected to train the model. This method of selecting features is called Correlation Matrix with Heatmap.

Correlation states how the features are related to each other or the target variable. The value of the correlation can be:

1. Positive (increase in one value of feature increases the value of the target variable)
2. Negative (increase in one value of feature decreases the value of the target variable)

Heatmap makes it easy to identify which features are most related to the target variable, we will plot heatmap of correlated features using the seaborn library.

The following heatmap shows the correlation among the attributes in the census dataset:



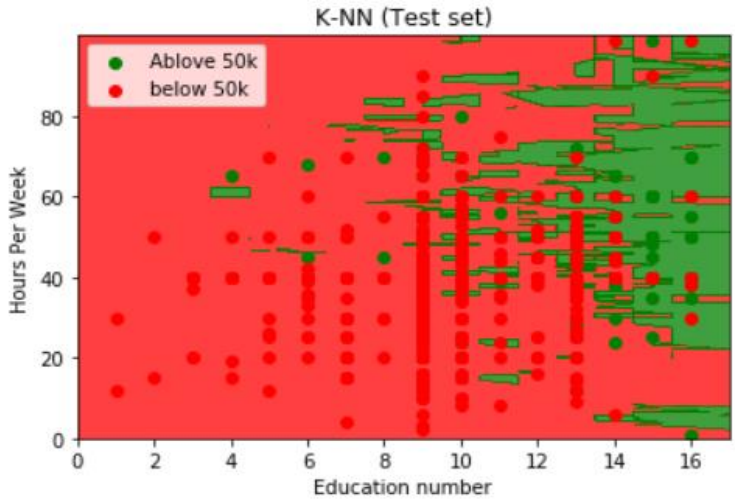
From the above correlation matrix, we can see that the education-num attribute and the hours-per-week attribute have the highest correlation when compared to other attributes in the dataset.

Hence we will choose education-num and hours-per-week attribute in training our model so that we could attain maximum accuracy and precision.

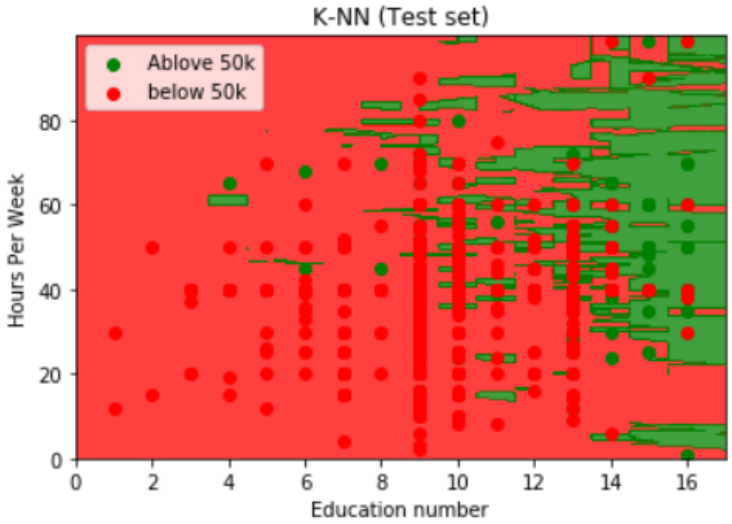
3. **Visualize the classifier in a 2D projection, for all three different number of neighbours. Interpret and compare the results.**

Following is the plot of graphs for the values predicted by the classifier against actual values

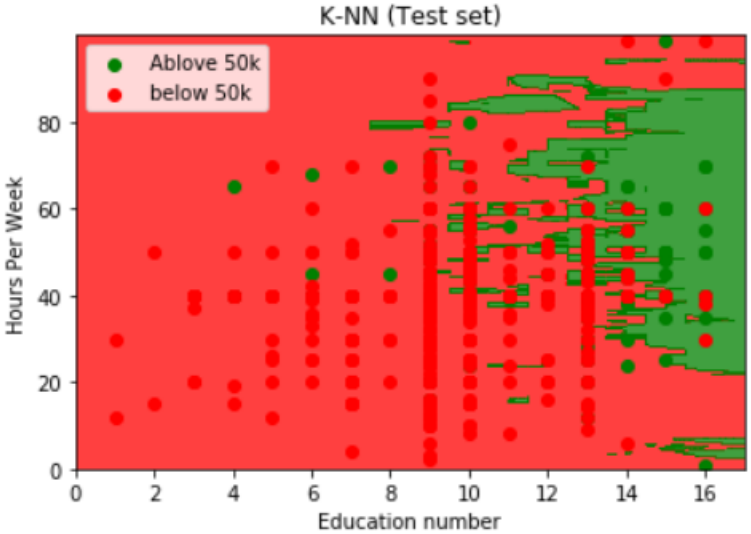
i) KNN classifier with neighbours = 5



ii) KNN classifier with neighbours = 7



iii) KNN classifier with neighbours = 32



When we build a classifier with neighbours = 5 then we get an accuracy of 75.18%, with neighbours = 7 we get an accuracy of 75.95% and with neighbours = 9 we get an accuracy of 76.61%. From this we can see that as we increase the neighbours our accuracy becomes higher and higher this is due to number of neighbours considered in predicting the target value. As we increase the number of neighbours for training the classifier we also seem increase the accuracy and when we decrease the number of neighbours we also tend to decrease the accuracy. We also choose an odd number while selecting the number of neighbours so as to resolve any conflicts of having equal number of different target classes.

Here we are using Euclidean distance to measure the distance between the test sample and its neighbours.

From the Graph we can observe that the points are very near to each other this is due to proximity of the points as there would very minute differences in values we see that almost that the point are almost attached to each other.

Hence we conclude that if we choose more number of neighbours to train our classifier we would get higher accuracy and prediction score.

REFERENCES:

[1] https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b
[2]https://stackoverflow.com/questions/23543406/getting-feature-names-in-addition-to-values-scikitlearnpandas
[3]https://stackoverflow.com/questions/18438997/why-is-pydot-unable-to-find-graphvizs-executables-in-windows-8
[4]https://datascience.stackexchange.com/questions/37428/graphviz-not-working-when-imported-inside-pydotplus-graphvizs-executables-not
[5]https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html
[6]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html#sklearn.metrics.classification_report
[7]https://stackoverflow.com/questions/19233771/sklearn-plot-confusion-matrix-with-labels
https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
[8]https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d
[9]https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
[10]https://www.saedsayad.com/k_nearest_neighbors.htm
[11]https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[12]https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html