* Gaussian Naive Bayes (GaussianNB)
* Decision Trees
* Ensemble Methods (Bagging, AdaBoost, Random Forest, Gradient Boosting)
* K-Nearest Neighbors (KNeighbors)
* Stochastic Gradient Descent Classifier (SGDC)
* Support Vector Machines (SVM)
* Logistic Regression

Question 2 - Model Application

List three of the supervised learning models above that are appropriate for this problem that you will test on the census data. For each model chosen

* Describe one real-world application in industry where the model can be applied.
* What are the strengths of the model; when does it perform well?
* What are the weaknesses of the model; when does it perform poorly?
* What makes this model a good candidate for the problem, given what you know about the data?
* Answer 1) Gaussian Naive Bayes (GaussianNB)
  + One real world application: A real world application for Gaussian Naive Bayes (GaussianNB) is in email spam filtering. GaussianNB can be used to classify emails as spam or not spam based on the presence of certain keywords or phrases. It can also be used to classify the sentiment of a customer review or social media post as positive, neutral, or negative based on the presence of certain words or phrases.
  + Strengths of the model and when does it perform well.
    - GaussianNB is simple and easy to implement. It requires minimal data preprocessing and has a small number of parameters to fit, making it a fast and efficient algorithm.
    - GaussianNB is highly scalable, meaning it can handle large datasets with ease. It is also resistant to overfitting, making it suitable for datasets with a large number of features.
    - GaussianNB is highly accurate and performs well on a wide range of classification tasks, particularly when the assumptions of the model are met (e.g. when the features are normally distributed).
    - GaussianNB is a robust algorithm, meaning it is not sensitive to the presence of irrelevant or noise in the data. It is also resistant to class imbalance, meaning it can handle situations where one class is heavily oversampled compared to the other.
    - GaussianNB can be used for both binary and multi-class classification tasks, making it a versatile tool for solving a variety of problems.
  + Weakness of the model and when does it perform poorly.
    - It assumes that all features are independent of each other, which is often not the case in real-world data. This can lead to poor predictions.
    - It is sensitive to the presence of outliers in the data, which can significantly impact the accuracy of the model.
    - It is not suitable for handling large amounts of data, as the computation time increases exponentially with the number of features.
    - It is not able to learn complex relationships between features, as it only considers the mean and standard deviation of each feature.
    - It is not suitable for categorical data, as it only works with continuous numerical data.
    - It is not robust to changes in the distribution of the data, as it assumes a fixed distribution for each feature.
    - It can be easily influenced by the presence of irrelevant features in the data, leading to poor predictions.
  + Why is this model a good choice? It is easy to implement and easy to explain to non-technical stakeholders.
* Answer 2) Support Vector Machines (SVM)
  + One real world application. One application is Image classification - SVMs can be used to identify objects in images, for example to recognize hand-written digits.
  + Strengths of the model and when does it perform well.
    - Effective in high-dimensional spaces: SVMs are particularly effective in cases where the number of dimensions (features) is greater than the number of samples. This is because the SVM tries to find the hyperplane that maximally separates the classes in the high-dimensional feature space.
    - Memory efficiency: SVMs can be trained using a subset of the training data, called support vectors, which makes them memory efficient.
    - Versatility: SVMs can be used for both classification and regression tasks and can be adapted to perform well in a variety of different situations, such as when the classes are unbalanced or when there is noise in the data.
    - Robustness: SVMs are generally robust to overfitting, which makes them a good choice for situations where the training data may be limited.
    - Overall, SVMs tend to perform well in many different types of classification and regression tasks, especially when the data is clean and well-separated.
  + Weakness of the model and when does it perform poorly.
    - Sensitivity to scale: SVMs are sensitive to the scale of the input features, so it is important to scale the features before training an SVM.
    - Computational complexity: The training time for SVMs can be high, especially for large datasets. In addition, the prediction time for SVMs can also be relatively slow, especially when compared to some other algorithms.
    - Difficulty interpreting the model: SVMs are considered "black box" models, which means it can be difficult to understand exactly how the model is making predictions. This can make it challenging to explain the decisions made by the model to stakeholders.
    - Limited to linear problems: By default, SVMs only consider linear relationships between the features and the target. While it is possible to use kernels to allow SVMs to learn more complex relationships, this can also increase the complexity and computational cost of the model.
    - To summarize SVMs may not perform as well when the data is highly imbalanced or when there is a lot of noise in the data. They may also be less effective for problems that involve very complex relationships between the features and the target, or when interpretability of the model is important.
  + Why is this model a good choice?
    1. Sample size is greater than 50 samples (have enough data to train with).
    2. Data is labelled.
    3. Predicting a category (works with classification)
    4. Sample size is less than 100k SVMs works better with smaller data sets, meaning greater computational efficiency and speed.
* Answer 3) K-Nearest Neighbors (KNeighbors)
  + One real world application: Credit scoring, KNN has been used in credit scoring systems to predict the likelihood of an individual defaulting on a loan based on their financial history and other characteristics.
  + Strengths of the model and when does it perform well.
    - Easy to implement: KNN is a very simple algorithm that is easy to implement and understand, which makes it a good choice for many applications.
    - No assumptions about the data: KNN makes no assumptions about the underlying data, which makes it a good choice for datasets where the relationship between the features and the target is not clear.
    - Good performance on small datasets: KNN can achieve good results on small datasets, as long as the data is clean and the classes are well-separated.
    - Versatility: KNN can be used for both classification and regression tasks, and can be adapted to a wide variety of different situations.
    - Overall, KNN tends to perform well when the data is clean and the classes are well-separated, and when the relationships between the features and the target are relatively simple.
  + Weakness of the model and when does it perform poorly
    - High computational cost: The prediction time for KNN increases with the size of the dataset, which can make it a slow algorithm for very large datasets.
    - Sensitivity to irrelevant features: KNN can be sensitive to irrelevant or redundant features, which can negatively impact the performance of the model.
    - Sensitivity to the choice of K: The performance of KNN can be sensitive to the choice of K, which can be difficult to set in practice.
    - Poor performance on high-dimensional data: KNN can struggle to perform well on datasets with many dimensions (features) due to the curse of dimensionality, which refers to the difficulty of finding patterns in data when there are many dimensions.
    - Overall, KNN may not perform well when the data is very large or when the relationships between the features and the target are very complex. It may also struggle to perform well on high-dimensional datasets
  + Why is this model a good choice?
    - Sample size is greater than 50 samples (have enough data to train with).
    - Predicting a category (works with classification).
    - Sample size is less than 100k
    - Data is labelled.

References:

* <https://en.wikipedia.org/wiki/Support_vector_machine#Applications>
* https://scikit-learn.org/stable/tutorial/machine\_learning\_map/index.html
* <http://scikit-learn.org/stable/modules/svm.html>
* <http://scikit-learn.org/stable/modules/sgd.html>
* <https://brilliant.org/wiki/k-nearest-neighbors/#pros-and-cons>
* <https://scikit-learn.org/stable/modules/neighbors.html>
* https://scikit-learn.org/stable/modules/gaussian\_process.html