**DSC 550 Data Mining**

**Trisha Winquist 3/2/2024**

**Sales Prediction Using Social Media Ads**

**Problem Statement:**  
  
Predicting sales of a company needs time series data of that company and based on that data the model can predict the future sales of that company or product.

Social Media Advertising plays a vital role in today’s business. It helps marketers to build relationships with their customers and increase sales. Marketers are using social media to advertise their products and generate sales. Social networking sites such as YouTube, Facebook, and Instagram are essential in today’s competitive business for boosting the sales of the firm. Social media advertisements can steer users to deals that they would most likely search for anyway, making the reality of a sale, impulse or not, higher. It's up to the company to channel customer impulses to make more effective sales. Social media advertising allows businesses to target specific groups of people based on their demographics, interests, and behaviors. This means that businesses can create ads that are more personalized and relevant to their target audience, increasing the likelihood of engagement and conversion. Top social media sites are becoming effective marketing tools, perhaps taking the place of more conventional options like TV ads or brochures. The Internet is a key marketing tool that may be utilized to increase brand awareness, draw in clients, and establish credibility.

Sales forecasts can be used to identify benchmarks and determine incremental impacts of new initiatives, plan resources in response to expected demand, and project future budgets.

**This project** will predict whether a customer will purchase or not purchase.

Getting buy-in is important because it guarantees everyone is on the same page, increasing the likelihood of success, improving collaboration, and leading to better decision-making.

When you have buy-in from stakeholders, colleagues, or other relevant parties, it increases the likelihood that your idea will be successfully implemented. This is because people will be more invested in the idea and committed to making it work

Getting team buy-in requires collaboration and communication with others, fostering stronger relationships and teamwork. When team members and stakeholders feel included in the decision-making process, they are more likely to feel a sense of ownership and responsibility, leading to greater collaboration and cooperation.

Marketing teams are growing and social media budgets increasing. And because of social media influencers, companies need to pay attention to capture the correct audience. Showing the link between marketing and business strategy is crucial.

Using data is one way to educate the team on the value of social media advertising, really helping to show the bigger picture of your vision and help verify all the potential you just demonstrated. This means sharing key industry statistics to showcase just how effective social media marketing can be for achieving business goals.

Lean on marketing examples from other businesses in your niche too. Because let's be honest, nothing captures the attention of a stakeholder better than news of a competitor's success.

It can be all too easy for social media marketing to take a back seat in the minds of stakeholders. But by maintaining a regular dialogue with them and educating them on marketing efforts and wins and how these translate to business value, you will do well in keeping them engaged and on board with your plans

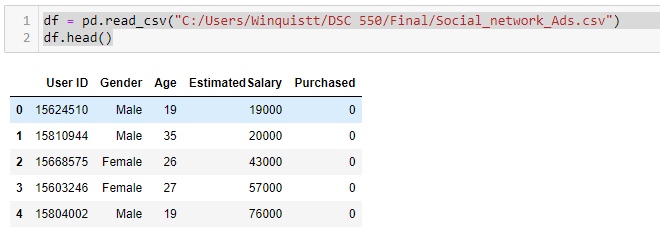
Since social media has grown over the years, it wasn’t too hard to find data on the web. I was able to find a good data set on a Kaggle competition platform. Using the fictional dataset of Gender, Age, Salary, and purchased (Target variable), the company wants to know whether a customer will buy its product or not

**An organized and detailed summary of Milestones 1- 3**

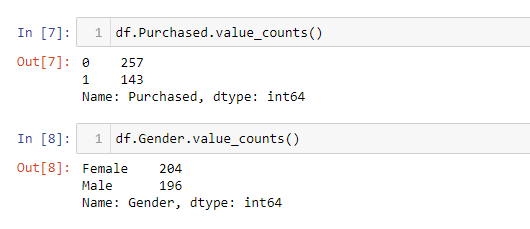
**Getting Started Milestone 1**

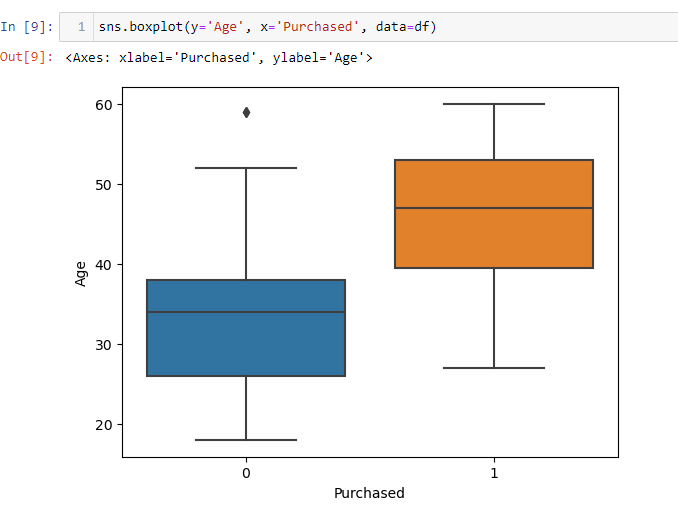
* Data wrangling, which consists of:
  + Gathering data
  + Assessing data
  + Cleaning data
* Storing, analyzing, and visualizing our wrangled data

The first step is to load the data and transform it into a structure that will be used for each model.

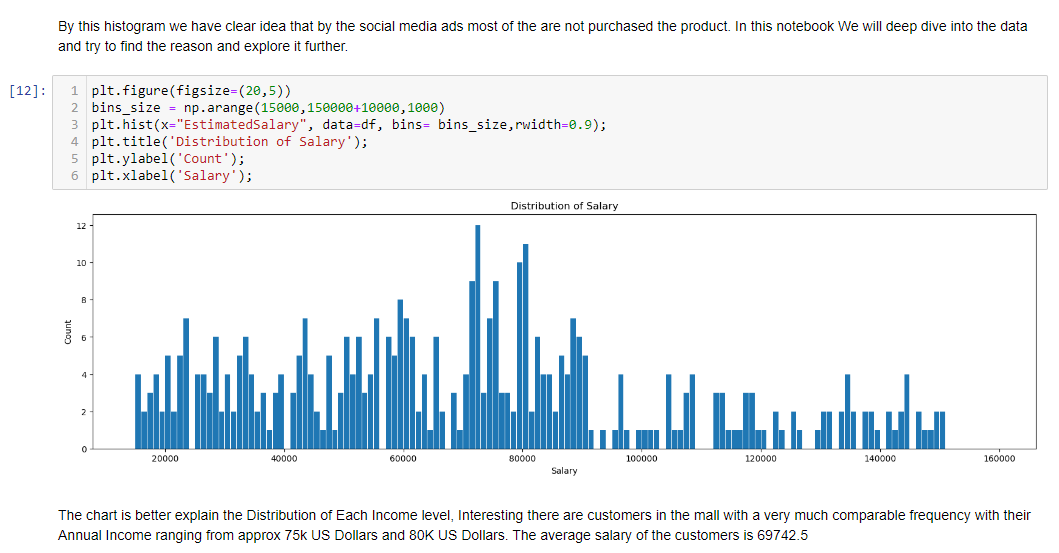


This data set was clean so there was no need to manipulate the data. The data set I am using is to classify social media ads and it indicates whether a person of a certain age and a given income buys the product.





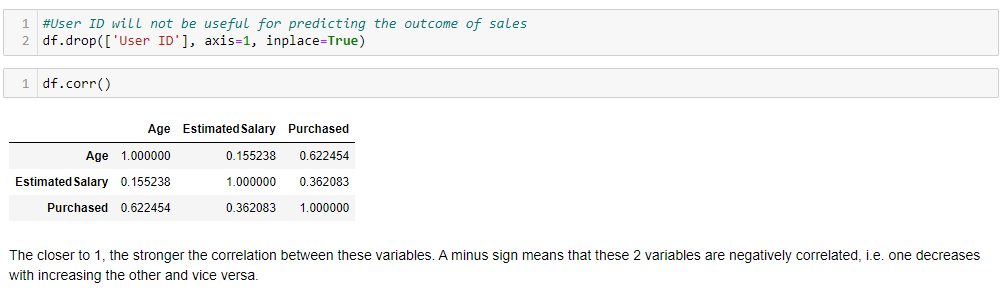
This boxplot shows the distribution of age that purchased after “looking” at the ad

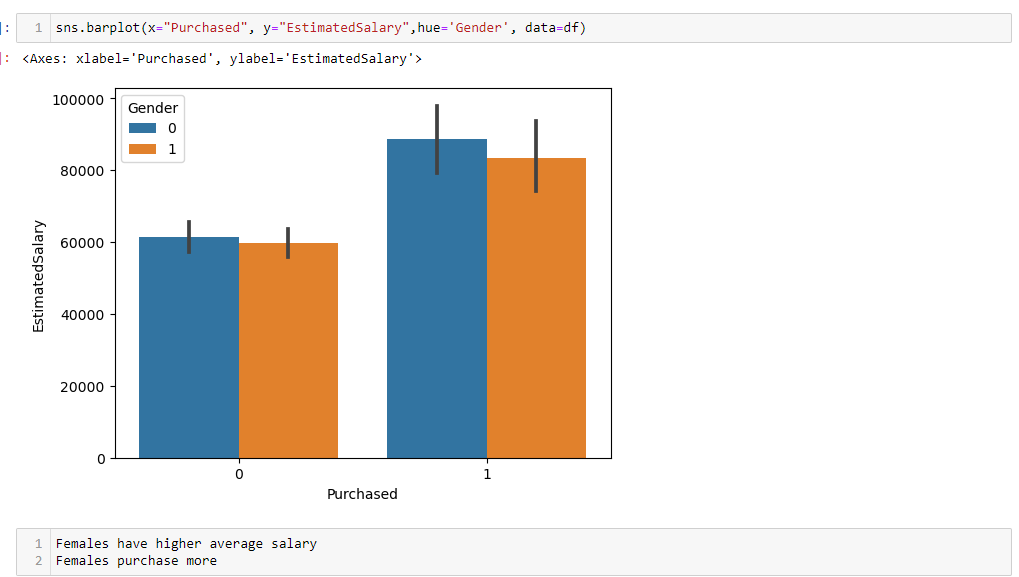


Graphs really can tell a story to your problem if you have the right graphs

**Milestone 2: In this milestone, we will start the process of data preparation.**

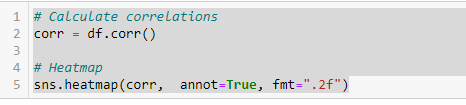
The user ID column will not be useful for predicting the outcome of sales, so we drop that column.

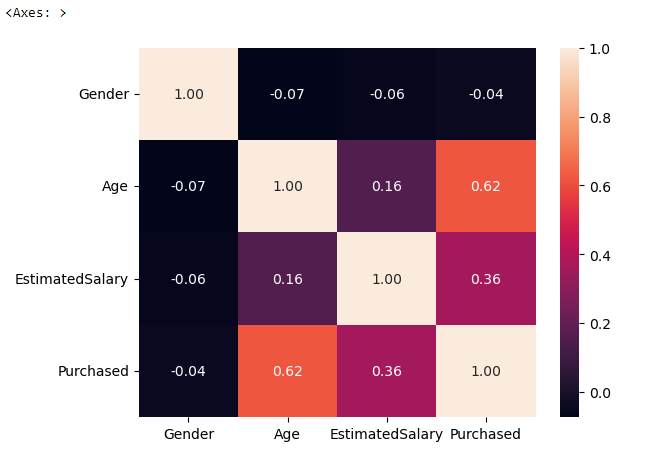




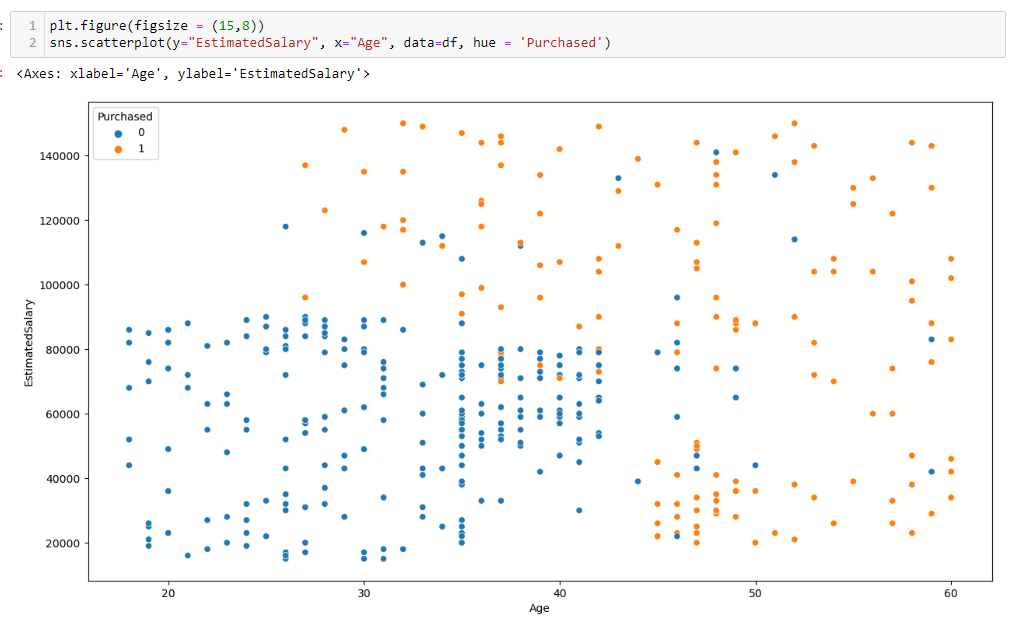
I wanted to calculate the correlation between Gender, Age, and Estimated Salary to predict if there was any correlation for purchasing.

# Calculate correlations:





The Above Graph shows the linear correlation between the different attributes of the Mall Customer Segmentation Dataset, This Heat map reflects the most correlated features with Beige Color and the least correlated features with black color. We can see that only age is linearly related to the purchase.



In this scatterplot, we can see who purchased and who did not purchase based on Estimated Salary and Age.

**Milestone 3 Model Training Preparing the Data**

In this step, I used a 75/25 split of the model to predict whether a person would purchase or not.



Using train\_test\_split () from the data science library scikit-learn, you can split your dataset into subsets that **minimize the potential for bias** in your evaluation and validation process.

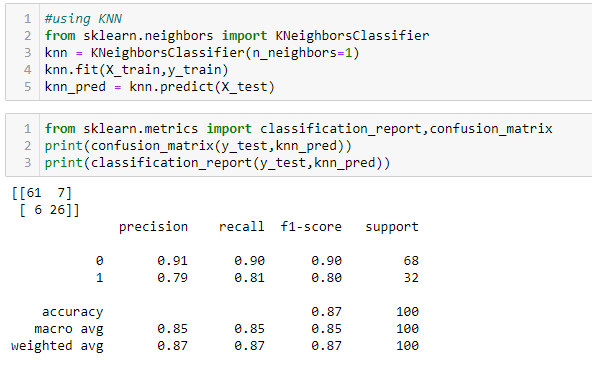
* **Purpose of Train-Test Split**:
  + The primary goal of supervised learning is to build a model that performs well on **new, unseen data**.
  + However, during model development, we only have access to the data we’ve collected so far.
  + To simulate how our model would perform on new data, we use the **train-test split** procedure.
* **How Train-Test Split Works**:
* **Data Arrangement**:
  + First, we arrange our data into a format suitable for the split.
  + In scikit-learn (a popular Python library for machine learning), this involves separating our full dataset into **“Features”** (input variables) and **“Target”** (output variable).
* **Data Splitting**
  + Next, we split the dataset into two parts:
    - **Training set**: Contains about 75% of the rows (you can adjust this ratio).
    - **Testing set**: Contains the remaining 25%.
* We randomly sample the data without replacement to create these sets.
* **Model Training**:
  + We train our machine learning model using the **training set** (denoted as **X\_train** and **y\_train**).
* **Model Testing**:
  + We evaluate the model’s performance using the **testing set** (denoted as **X\_test** and **y\_test**).
  + This step helps us understand how well the model generalizes to unseen data.

**train-test split** is a fundamental practice in machine learning to ensure our models perform well beyond the data they were trained on

**Classification** is a technique that is useful for determining the class based on one or more independent variables.

KNN Classifiers (K nearest neighbors)

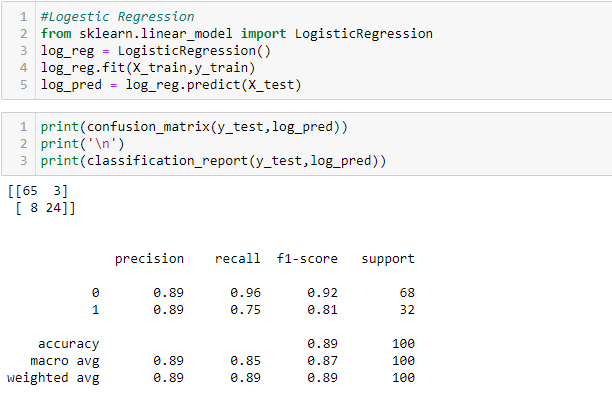
Because the KNN classifier predicts the class of a given test observation by identifying the observations that are nearest to it, the scale of the variables matters. Any variables that are on a large scale will have a much larger effect on the distance between the observations, and hence on the KNN classifier, than variables that are on a small scale

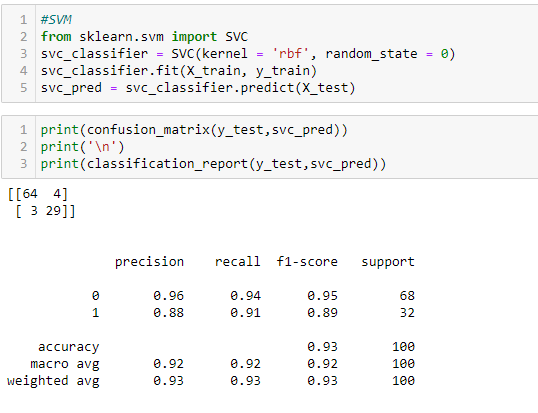


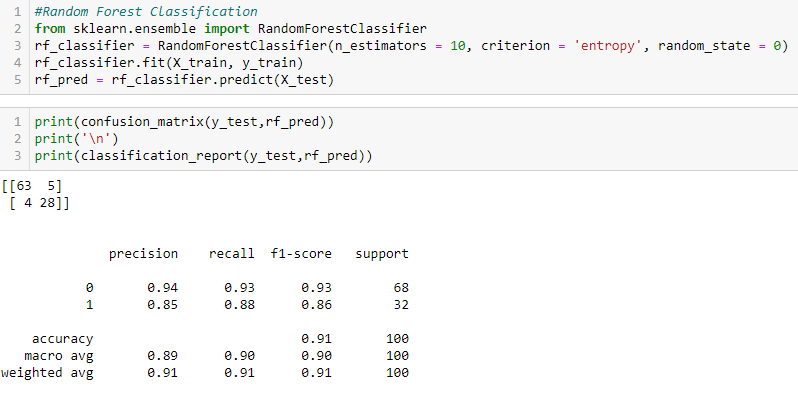
**Choosing a K Value**- The choice of k is very critical – A small value of k means that noise will have a higher influence on the result. A large value makes it computationally expensive and kind of defeats the basic philosophy behind KNN (that points that are near might have similar densities or classes)

I wanted to use different classification models and use a K Fold cross-validation to train the model. I used:

* Logistic Regression **67.5%** mean accuracy
* SVM **77.2%** mean accuracy
* Random Forest Classification **84.7%** mean accuracy





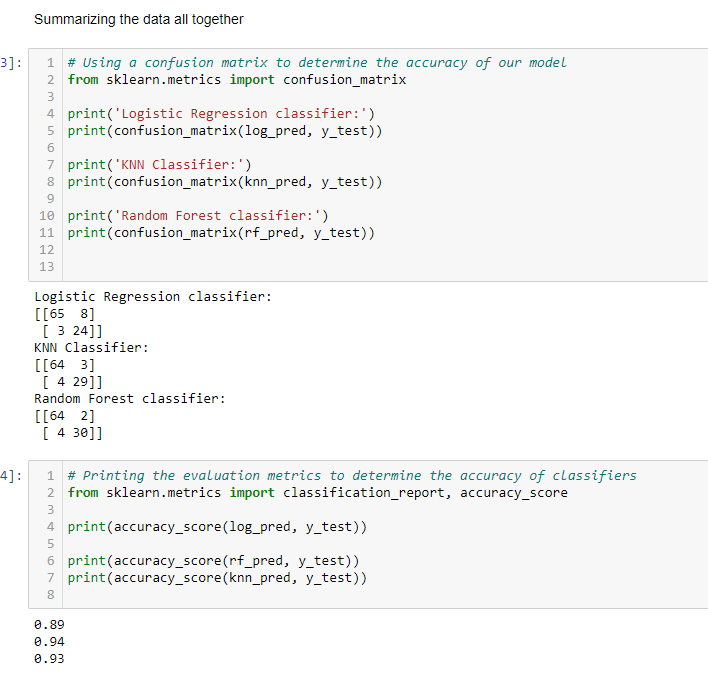


Logistic regression is **easier to implement, and interpret, and very efficient to train**. If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting. It makes no assumptions about distributions of classes in feature space. Based on the 3 prediction classifiers, my logistic regression model was a better fit.

**Running Hyperparameter**

In **machine learning**, **hyperparameters** are configuration variables that are set before the training process of a model begins. Unlike regular parameters, which are learned from the data during training, hyperparameters control the learning process itself.

1. **Model Hyperparameters**:
   * These hyperparameters are related to the architecture and structure of the model itself. They typically cannot be inferred from the training data because the objective function is often non-differentiable concerning them.
   * Examples of model hyperparameters include:
     + **Topology and size of a neural network**: How many layers and neurons the network should have.
     + **Degree of regularization**: Determines how much to penalize complex models to prevent overfitting.
     + **Number of clusters in a clustering algorithm (e.g., k-means)**.
2. **Algorithm Hyperparameters**:
   * These hyperparameters are specific to the learning algorithm and affect its behavior during training.
   * Examples of algorithm hyperparameters include:
     + **Learning rate**: Controls the step size during gradient descent optimization.
     + **Batch size**: Determines how many training examples are used in each iteration.
     + **Choice of optimizer**: Determines how the model’s parameters are updated during training.
3. **Tunability**:
   * Most performance variation in machine learning models can be attributed to just a few hyperparameters.
   * The tunability of an algorithm or hyperparameter measures how much performance can be gained by tuning it.



By the hyperparameter tuning, KNN performed best in the above model.

**Conclusion**

Based on the metrics of the model, I would predict that my model is good at predicting sales accurately. The outputs from these kinds of analysis can easily be incorporated into marketers' dashboards and activity systems. These models enable them to see the impact of their and their competitor's activity and overall brand health and sales as well as changes in tone, message, and strategy that could accelerate growth in brand equity and sales.

You can model your ML to reflect any (or all) of the traditional models and use your experience to tell which factors should be given the highest relevance. Your sales representatives can all contribute to your ML engines by powering them with their quantitative and qualitative information.

Once trained and verified in the initial testing stage, your ML can function as an independent assistant. Whenever it spots any new behavioral patterns or predicts potential risk factors, you and your team will be the first to know and, ultimately, adjust.

This project showcases the effective utilization of data analysis, feature engineering, and machine learning techniques to solve a real-world forecasting problem. The insights gained from the analysis provide valuable information for decision-making within the retail industry.