MSADS
Portfolio Milestone

SYRACUSE UNIVERSITY JUNE 2022

Cal Wardell
ID – 889182165 CWARDELL@SYR.EDU

Table of Contents

Introduction 3

IST 687 Introduction to Data Science 3-10

IST 659: Database Administration & Database Management 10-13

IST 719: Information Visualization 13-17

IST 652: Scripting for Data Analysis 18-23

Conclusion 23

References 24

Introduction

The Applied Data Science program at Syracuse University is designed to give students the skills necessary to become professionals who can analyze data to solve problems.

These skills include:

- Data collection, transformation, and storage
- Knowing how to visualize the data to both see trends and patterns and report findings
- Using supervised and unsupervised machine learning to gain understanding of the data and make predictions
- Reporting the findings and tailoring them to the understanding of individuals with various technical backgrounds
- Making recommendations from the analysis to the business/organization/management/shareholders

The following projects demonstrate the skills I have learned in various classes of my program:

- IST 687 Introduction to Data Science
- IST 659 Database Administration & Database management
- IST 719 Information Visualization
- IST 652 Scripting for Data Analysis

IST 687 Introduction to Data Science

This course gave me hands-on experience with the process of collecting, processing, transforming, managing, and analyzing data. I learned the concepts of applied statistics, visualization, text mining and machine learning to evaluate the data. I programed in the language of *R* in the software *R-Studio*.

Project description

Goal: Determine which factors correlate with individual income to be greater or less than $50,000 per year.

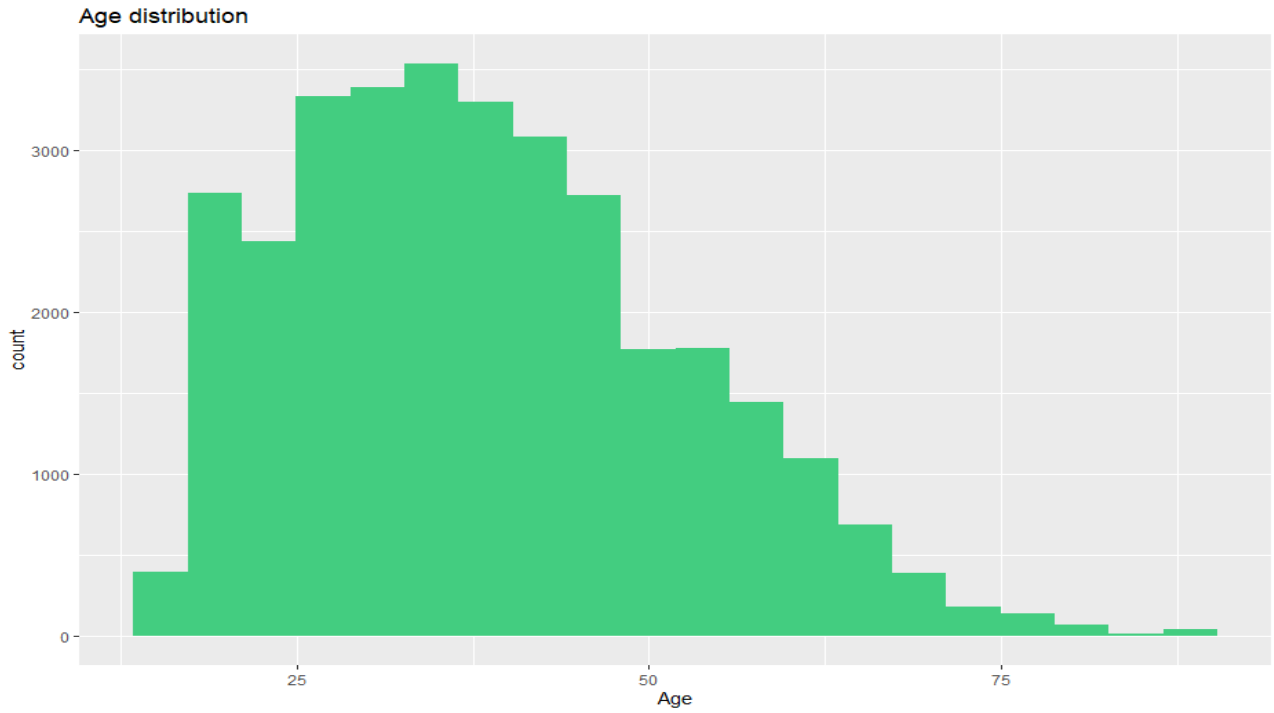
About the data: The data for the analysis is extracted from the 1994 census, retrieved from the *UCI Machine Learning Repository* website (Dua, 2019). It provides over 32,000 rows containing information of the individual's age, working class, education, marital status, occupation, relationship to head of household on the census, race, sex, hours per week worked, native country, and whether the person made greater or less than $50,000 annually. Additionally, the data contained the following three columns that the website did not clearly explain: *fnlwgt, capital-gain,* and *capital-loss.*

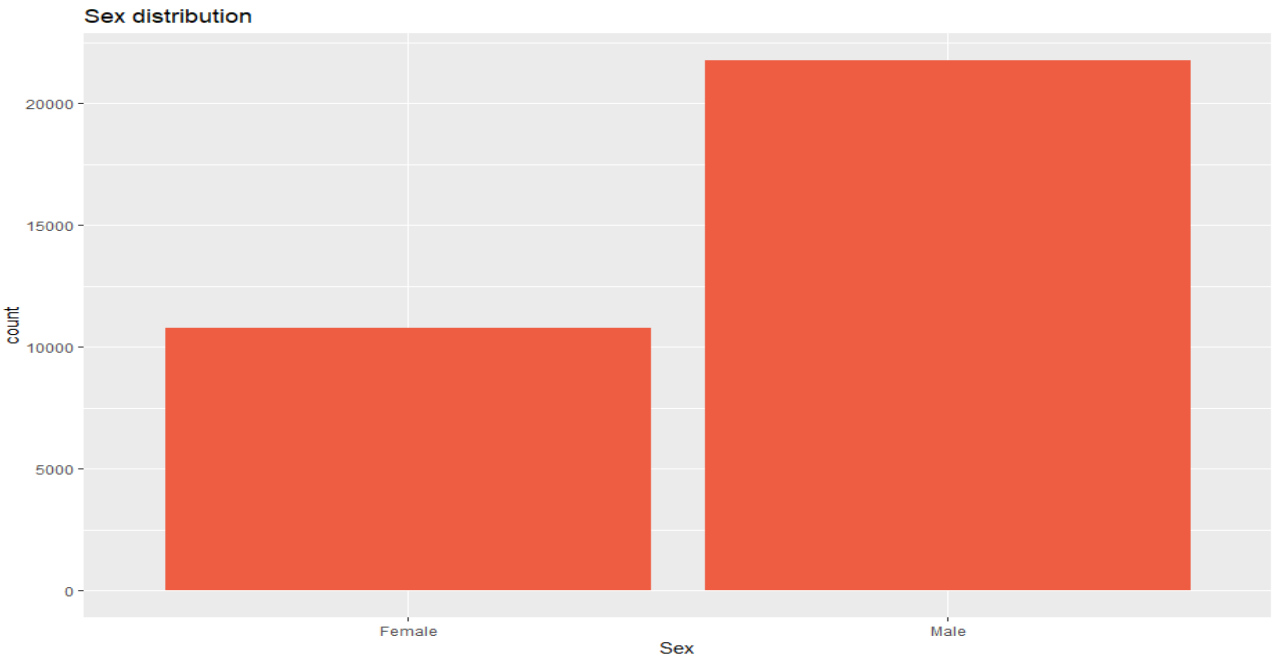
This information can be used by local and national governments, non-profit groups, and other organizations that aim to raise individual incomes to decrease the rate of poverty.

Step 1: Clean and prep the data

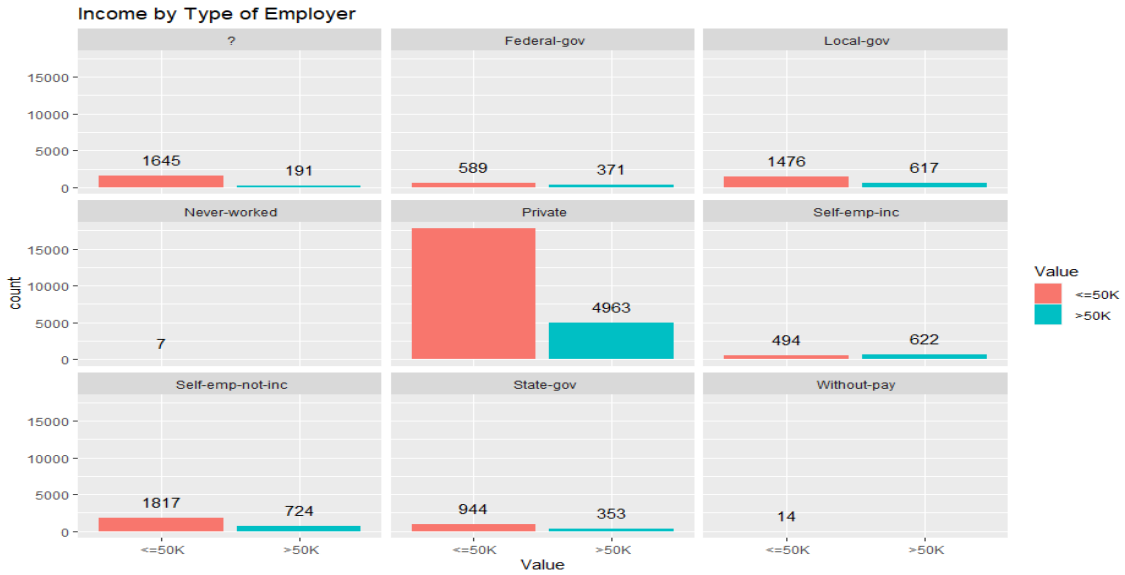
My group and I examined the data for null values and transformed the needed columns to factors to be used in the machine learning model.

Step 2: Visualize the data



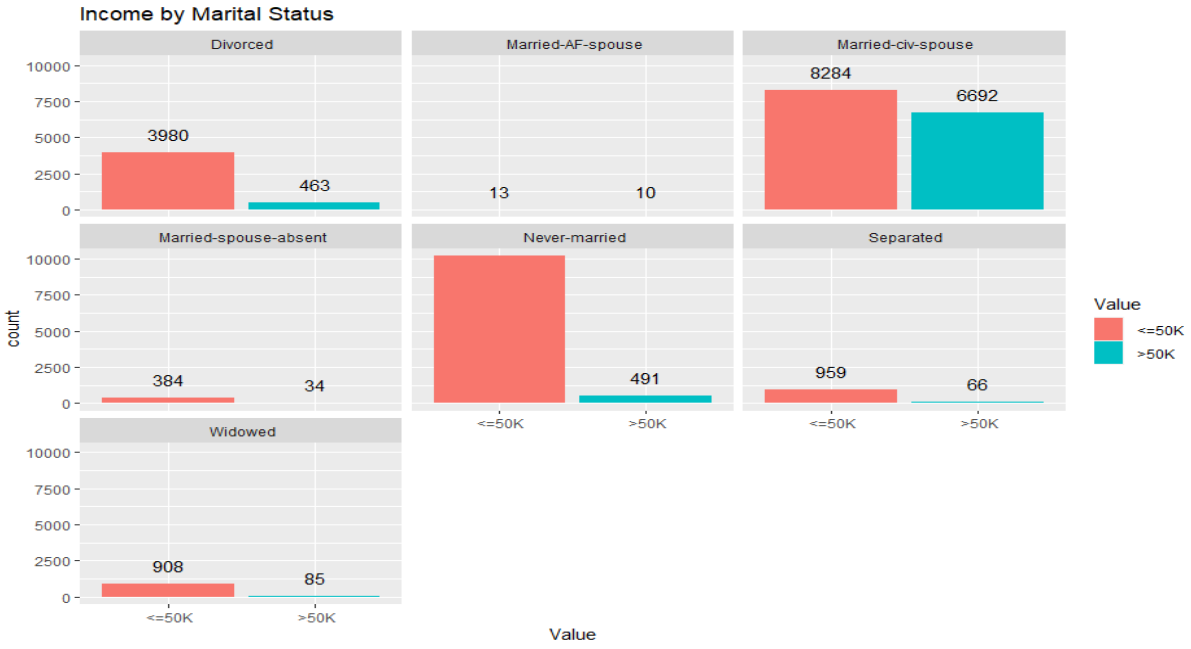


We visualized the number of individuals who made more or less than $50,000 annually by type of employment.



The data suggests that individuals who are employed by the federal government or who are self-employed with an incorporated business are more likely to be making over $50,000. However, the overall percentage of individuals in these situations is much less than the other employment situations.

Next, we visualized the number of those making more or less than $50,000 by marital status.

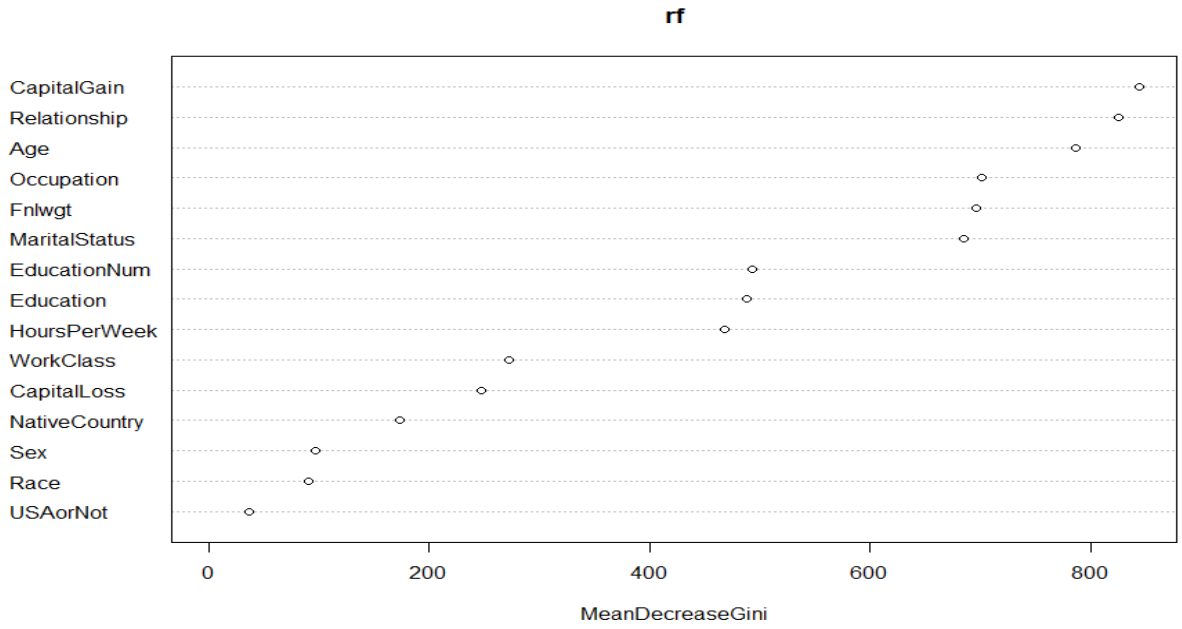


The data implies that a married individual is more likely to earn over $50,000 than any other marital status.

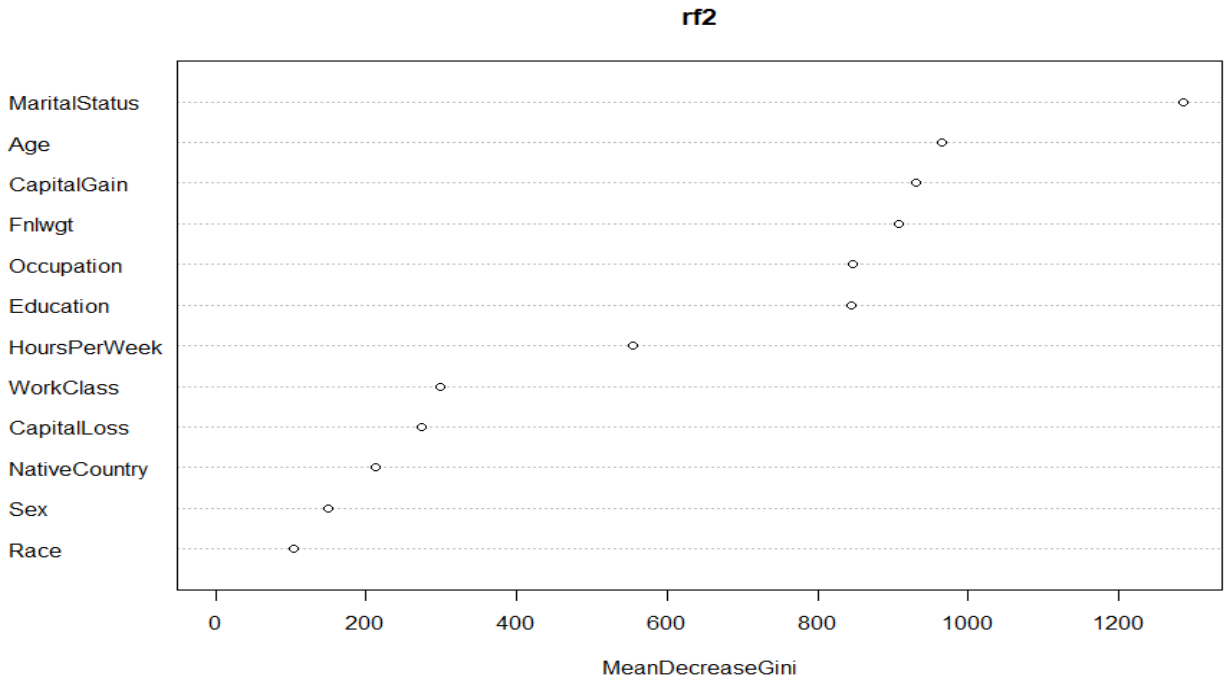
Step 3: Create a test and training data set

Because there were 42 distinct countries in the data set, we chose to look primarily at whether the individual's native country was the United States.

Step 4: Create and test the *Random Forest* model



First, we built a *Random Forest* model with all the variables. The graph above shows the significance of each variable in the model. *USAorNot* was insignificant, so we reran the model without it. Additionally, because *Education* and *EducationNum* both refer to the same characteristic, we did not include *EducationNum* in the model. Finally, we removed *Relationship* as we felt that *Marital Status* was sufficient. After running the model again, the graph below depicts the significance of the variables:



Here are the initial results of our model:

```
Call:
 randomForest(formula = Value ~ ., data = train2)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 13.8%
Confusion matrix:
        <=50K  >50K class.error
 <=50K  16006  1201  0.06979718
 >50K    1945  3640  0.34825425
```

Our model had an initial accuracy of 86.2%. It was 93.03% accurate at predicting below or equal to $50,000, but only 65.18% accurate at predicting if the individual made more than $50,000. We then tested our model; below are the results:

```
          Reference
Prediction  <=50K  >50K
     <=50K   7008   504
     >50K     790  1466

               Accuracy : 0.8675
                 95% CI : (0.8606, 0.8742)
    No Information Rate : 0.7983
    P-Value [Acc > NIR] : < 2.2e-16
```

With a p-value well below the 0.05 mark, our model was statistically significant. It was 86.75% accurate, and more accurate at predicting those who make less than or equal to $50,000 than those who make more than $50,000.

Step 5: Create and test the *XGBoost* model

We then created an *XGBoost* model on the data. We ran it with the default parameters and here are the results when we ran it against the test:

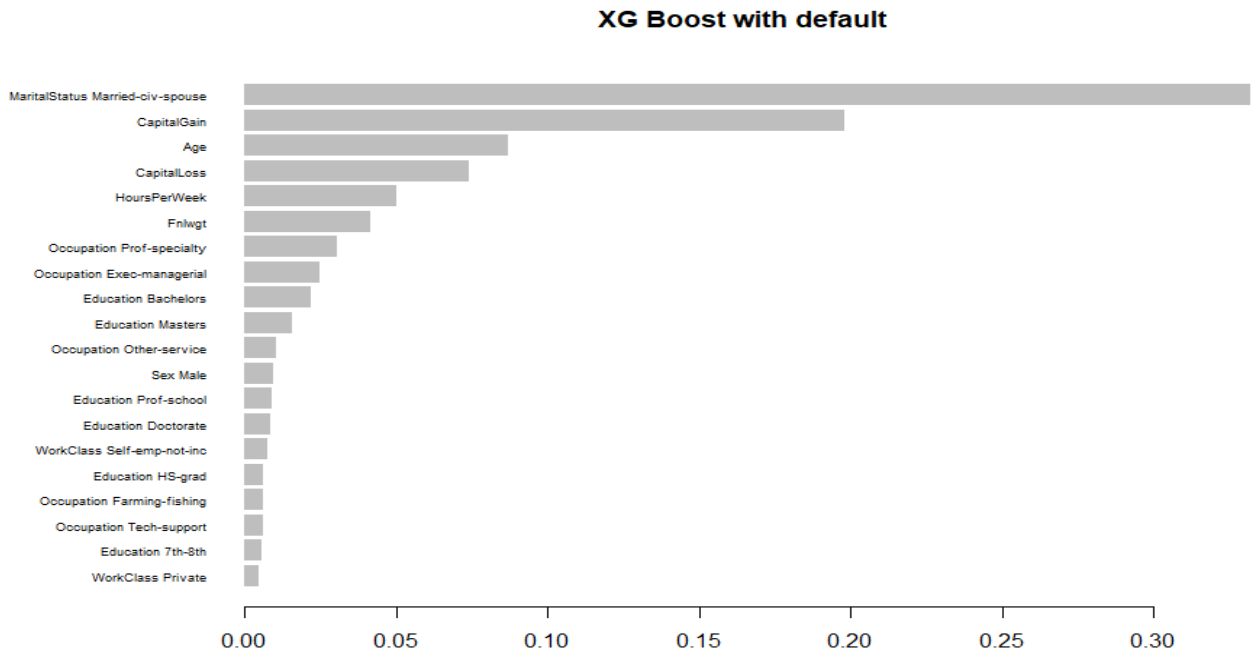
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7082  776
         1  430 1480

               Accuracy : 0.8765
                 95% CI : (0.8698, 0.883)
    No Information Rate : 0.769
    P-Value [Acc > NIR] : < 2.2e-16
```

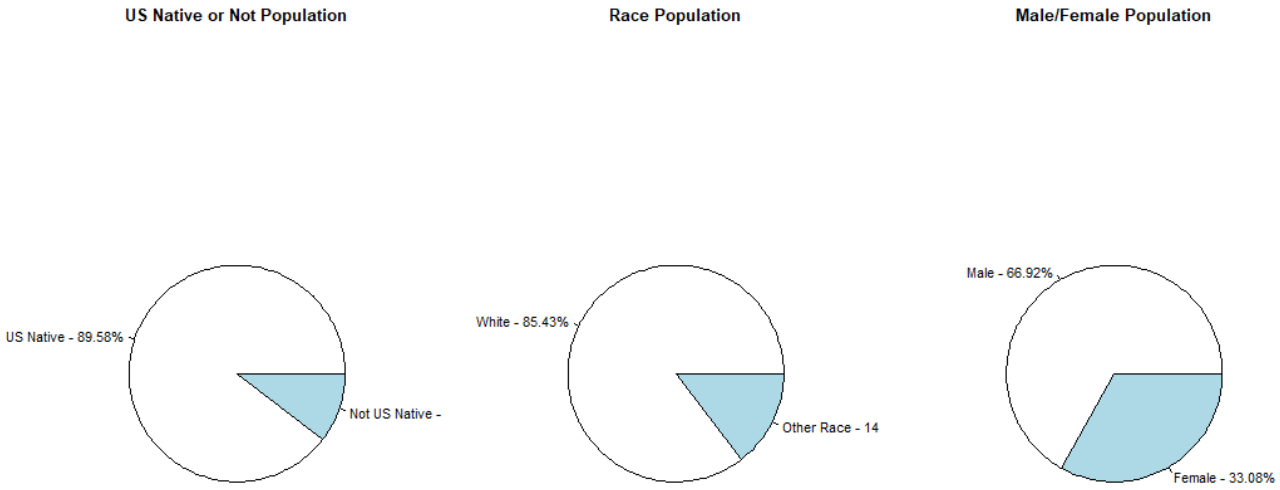
This was more accurate than our *Random Forest* model. This model was statistically significant as it had a p-value less than 0.05.

Below shows a graph depicting which variables had the greatest impact on our model. Like *Random Forest*, marital status had the greatest influence.



Acknowledgement of potential bias

To be thorough in our analysis, we assessed the demographics to see if we had potential bias. We looked at ratio for *US Native* vs *Not US Native*, *White* vs *Other Race*, and *Male* vs *Female*, which is pictured below:



Our data set was mostly *White*, *US Native*, and *Male*. Further analyses may be needed with other demographics to see if the results are consistent with other populations.

Conclusion

Our *XGBoost* model was 87% accurate at predicting if an individual would make greater or less than $50,000 annually. The most influential variable to determine whether a person makes more or less than $50,000 was marital status. The graphs suggest that married individuals had a better chance of making greater than $50,000 than those who are not married. Our recommendation is to increase programs and incentives that encourage individuals to get married, as this will increase the likelihood of earning more than $50,000 or more annually.

Skills learned from this project:

- Load, clean, and prepare data for analysis
- Create useful visuals from our dataset
- Train and test *Random Forest* and *XGBoost* models and visualize which variables are most influential in the models
- Make recommendations from findings in the analysis

IST 659: Database Administration & Database Management

This class gave me experience with database design, implementation, and management. I became competent with entity-relationship diagrams, the basics of *Structured Query Language (SQL)*, data normalization, and hierarchical, network, and relation data models. I did my work in *SQL Server Management Studio.*

Project description

Goal: Create a database of players who have appeared in the *National Basketball Association (NBA) All-Star Games* to determine which schools/leagues are producing the top players in the *NBA.*

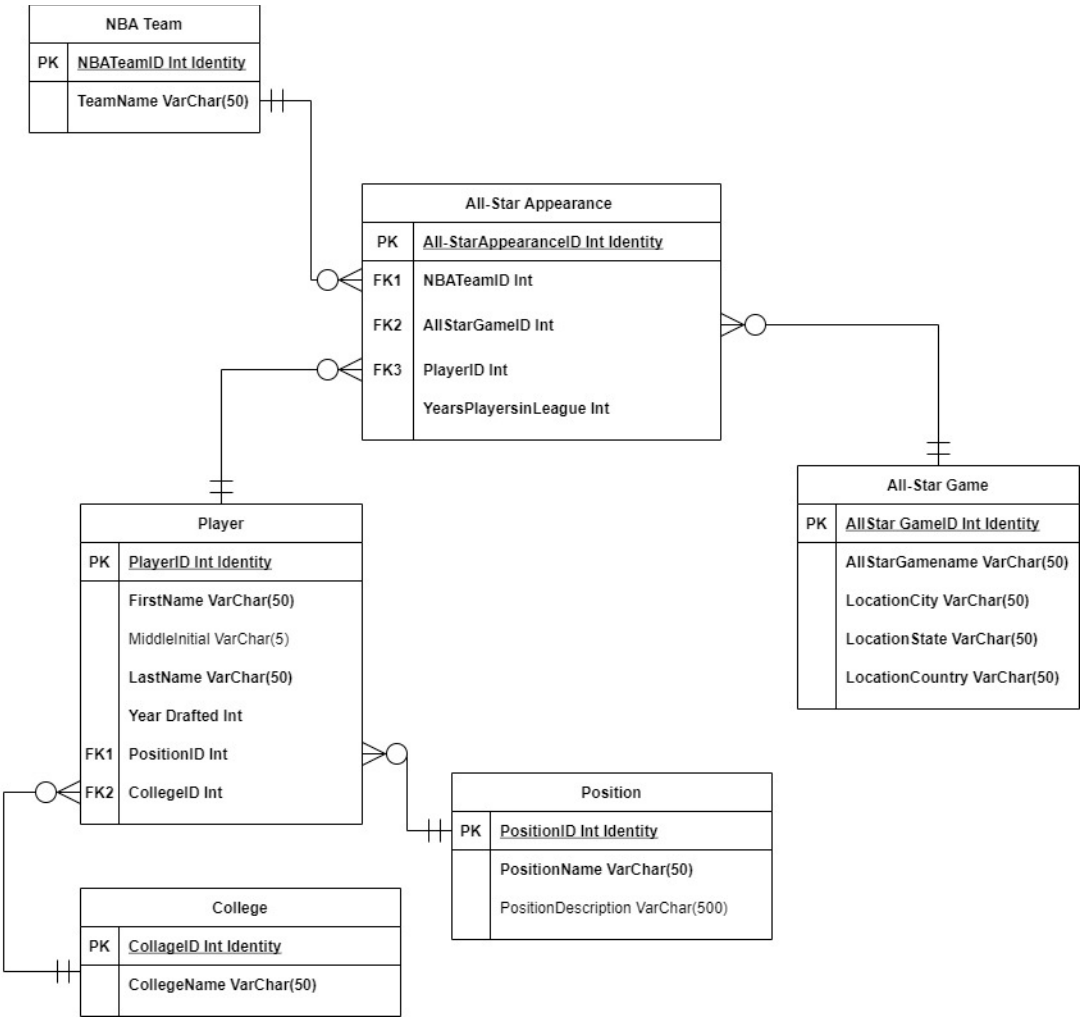
About the data: This database had information on each of the players in the 2019 and 2020 *All-Star Game*. The data I collected includes: the player's position, *NBA* team, college/high school/overseas league they played for before joining the *NBA*, and which *All-Star Games* they played. My main two questions were:

1. Which college/high school/overseas organization provided the most *NBA All-Stars*?
2. Does a college/high school/overseas organization produce more *NBA All-Stars* who play a certain position?

With this information, scouts would know where to potentially find and draft an *All-Star* to play a needed position.

Step 1: Create an *Entity Relationship Diagram (ERD)*

To begin the project, I created an *ERD* to determine how the data would be normalized.



Step 2: Make queries that build the database

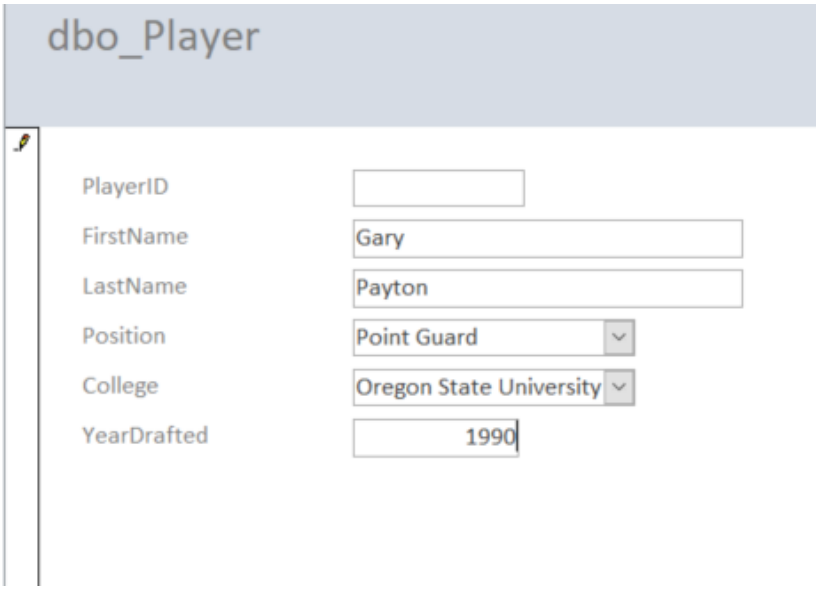
Once I had the plan for the database, I created the queries to build each table. Next, I inserted data into each of the tables. Below are some examples of my queries:

```
Create Table Player (
        --Creating Column
        PlayerID int identity Primary Key,
        FirstName VarChar(50) not null,
        LastName VarChar (50) not null,
        PositionID int not null,
        CollegeID int not null,
        YearDrafted int not null,
        --Creating Constraints
        Constraint FK1_Player Foreign Key (PositionID) References Position(PositionID),
        Constraint FK2_Player Foreign Key (CollegeID) References College(CollegeID)
        )
```

```
--Inserting values into table Player
INSERT INTO Player (FirstName,LastName, PositionID, CollegeID, YearDrafted)
VALUES                      ('James', 'Harden', 2, (SELECT College.CollegeID FROM College
WHERE CollegeName= 'Arizona State University'), 2009)
                                ,('Kawhi', 'Leonard', 3, (SELECT College.CollegeID FROM
College WHERE CollegeName= 'San Diego State University'), 2011)
```

Step 3: Create front end to my database

With the database set up, I connected *Microsoft Access* to act as the front end. I created a form for each table that can be used to add data. Here is an example of the player form:



I created a report to show the different schools/organizations in our database and the information of *All-Stars* that had come from said school/organization. See a depiction below:

Colleges and All Stars

CollegeName	FirstName	LastName	PositionName	TeamName	YearGameWasPlayed
Arizona State University					
	James	Harden	Shooting Guard		
				Houston Rockets	2020
Cholet Basket					
	Rudy	Gobert	Center		
				Utah Jazz	2020
Duke Univeristy					
	Brandon	Ingram	Power Forward		
				New Orleans Pelicans	2020
	Jayson	Tatum	Small Forward		
				Boston Celtics	2020
	Loul	Deng	Small Forward		
				Chicago Bulls	2012
Filathlitikos					
	Giannis	Antetokounmpo	Power Forward		
				Milwaukee Bucks	2020
Gonzaga Univeristy					
	Domantas	Sabonis	Power Forward		
				Indiana Pacers	2020
Louisiana State University					
	Ben	Simmons	Point Guard		
				Philadelphia 76ers	2020
Marquette University					
	Jimmy	Butler	Small Forward		
				Miami heat	2020

Conclusion

From the section depicted above (which comes from the 2019 and 2020 *All-Star Games*), we see that Duke University produced several *All-Star* players at the forward position. A team that needs players in this position could use this information to search for and draft players from Duke.

Skills learned from this project:

- Design a database using an *ERD*
- Build the database and insert data using SQL
- Create a report from the database to generate insight

IST 719: Information Visualization

IST 719 provided a solid introduction into many data visualization techniques. I completed my work in *R-Studio* and *Adobe Illustrator*. A key aspect of our learning includes data cleaning and prepping as this is an integral part of visualization.

Project description

Goal: Through visualization, answer various questions about cars being sold in Poland to obtain the best value for the price of a car.

About the data: This data set comes from *Kaggle.com* (Glotov, 2022). It contains the following information on cars being sold in Poland: make, model, generation, year, mileage, engine size in cc's, engine type, city in Poland, region of Poland, and price of the car in Polish złoty (PLN). My top two questions were:

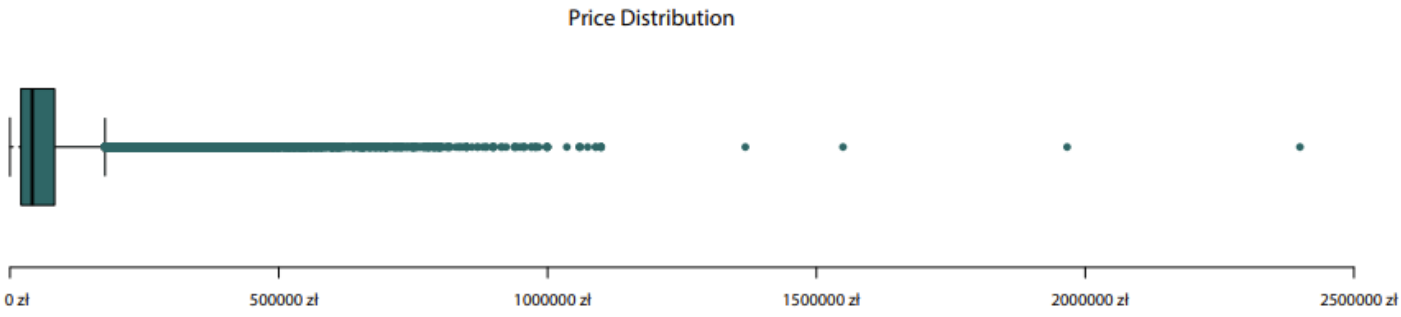
1. How does mileage affect the price of a car?
2. Which car makes have the cheapest selection of cars?

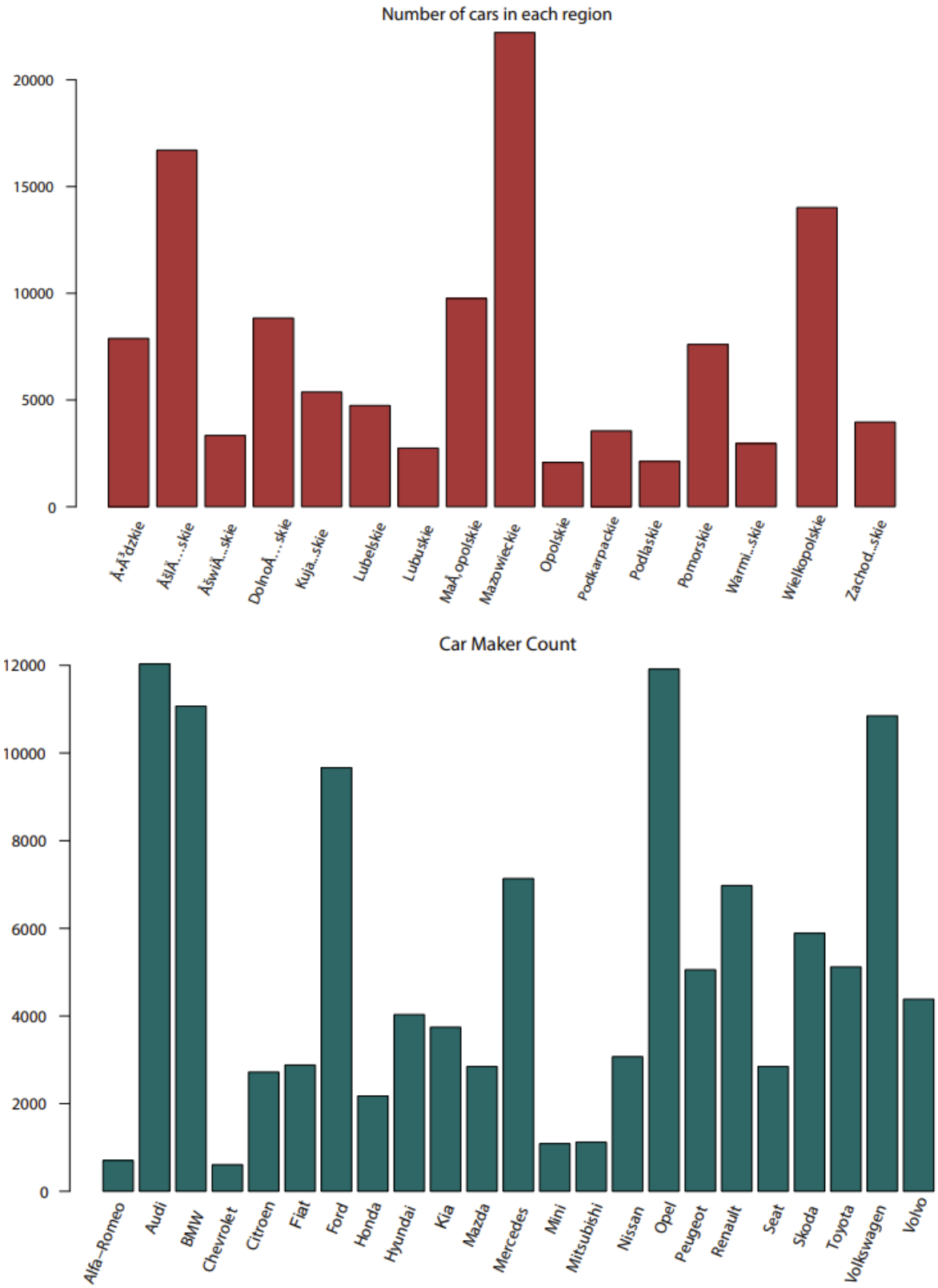
Step 1: Clean and prep the data

Upon looking at the data, I noticed some bad values; for example, ')' was a location. I filtered these bad values out. I also wanted to focus on the regions of Poland that had the greatest selection of vehicles, because more options provide a better probability of getting a good deal. So, I filtered out the regions that had a very low selection of cars.

Step 2: Create plots

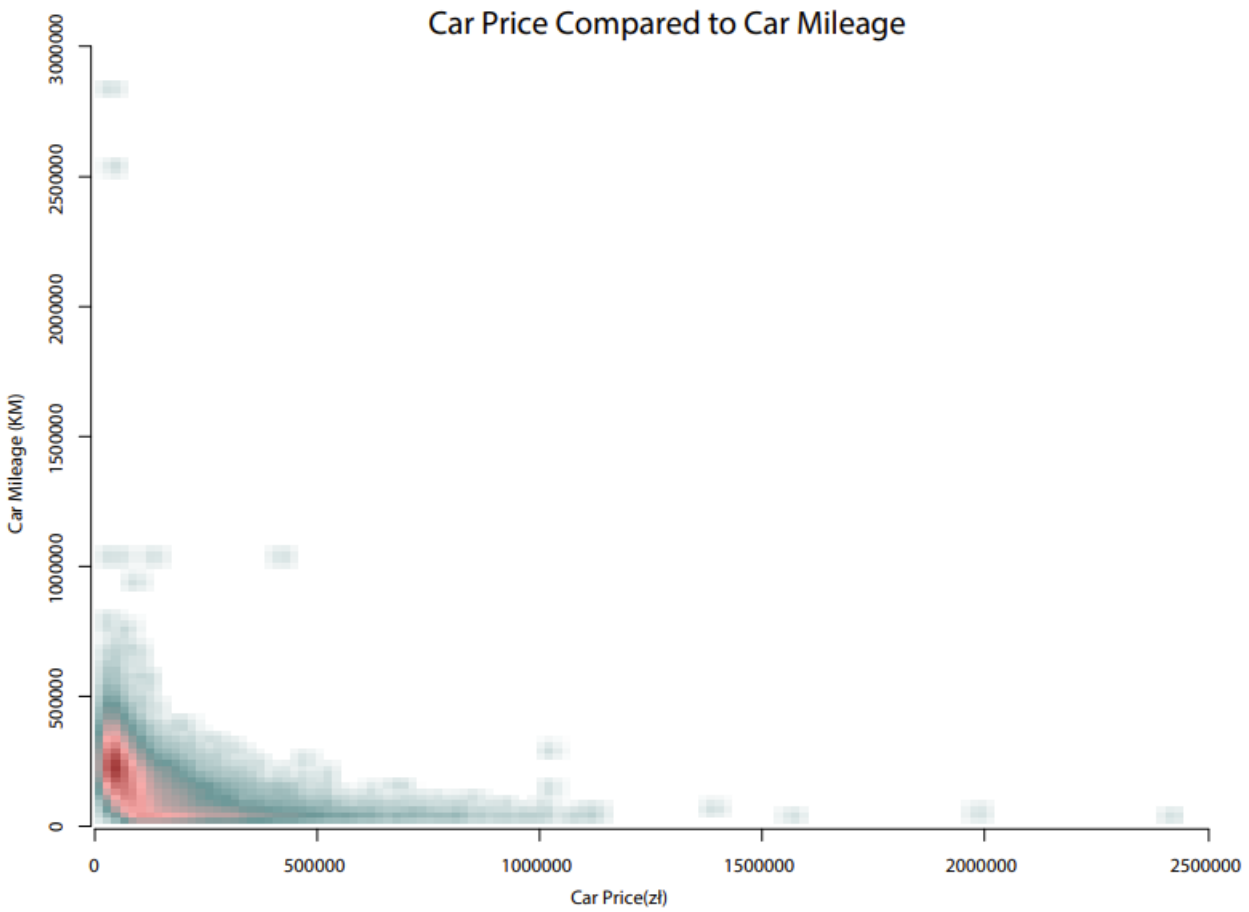
I created single-dimensional plots to better understand the data.





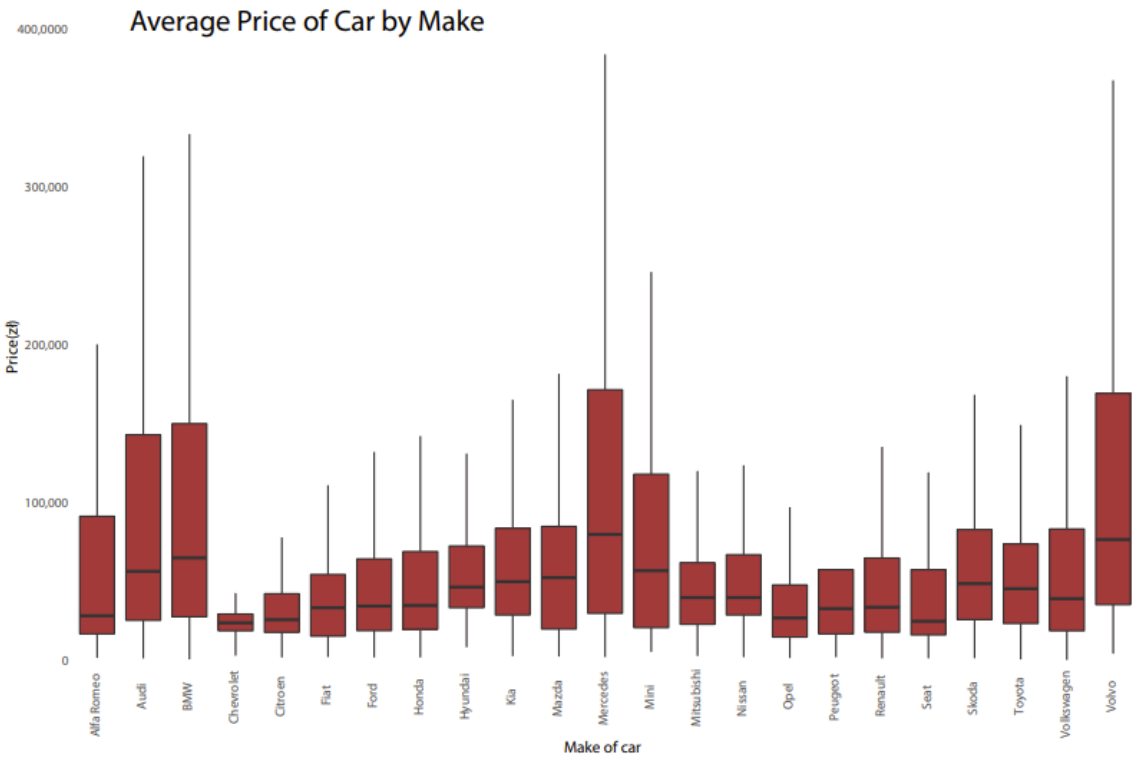
After I gained insight on which models and regions had multiple options to choose from, I created two-dimensional plots to answer my questions.

1. How does mileage affect the price of a car?



This plot implies that cheaper cars generally have more miles while more expensive cars have fewer miles. Additionally, the plot suggests that the greatest concentration of price and mileage combinations is roughly 100,000 zloty and 25,000 km.

2. Which car makes have the cheapest selection of cars?



Mercedes, *Volvo*, *Audi*, and *BMW* are the car makes with the largest distribution of pricing, making them less likely to offer a cheaper car. *Ford*, *Honda*, and *Opel* have a greater chance at providing more affordable cars because the distribution of their prices is much smaller and lower.

Conclusion

Finding a good deal on a car can be difficult, but visualizing the data offers quick and efficient insight to help our decisions. The recommendation I would make to an individual searching for a good deal on a new car would be to look at a *Ford* in the Mazowiekie region of Poland. This make has a low-price range and has many cars for sale. In addition, the Mazowiekie region has the largest selection of vehicles in Poland, providing customers with a better chance at finding a good deal. Finally, because the data suggests that low mileage correlates with higher cost, I would recommend looking for a car with relatively low mileage but not at a very high price.

Skills learned from this project:

1. Create visuals in *R* to gain insight about data
2. Use functions in base *R, ggplot,* and *smoothscatter* packages for visuals
3. Edit in *Adobe Illustrator* to enhance visuals

IST 652: Scripting for Data Analysis

During this class, I became experienced with data wrangling. I learned how to acquire, transform, and prep both structured and unstructured data for analysis. *Python* was the programming language I used.

Project description

Goal: Answer questions about the *National Football League (NFL)* by analyzing play-by-play data from all regular season games during years 2009 to 2018.

About the data: The data set came from a CSV downloaded from *Kaggle.com* (Horowitz, 2017). It is detailed play-by-play data from regular season games during years 2009 to 2018. The data has over 350,000 rows and 100 columns. Some of the columns in the data set include which team has the ball, type of play, result of the play, time left in the game, and current score at the time of the play. As this was a group project, my responsibility was to answer the following questions:

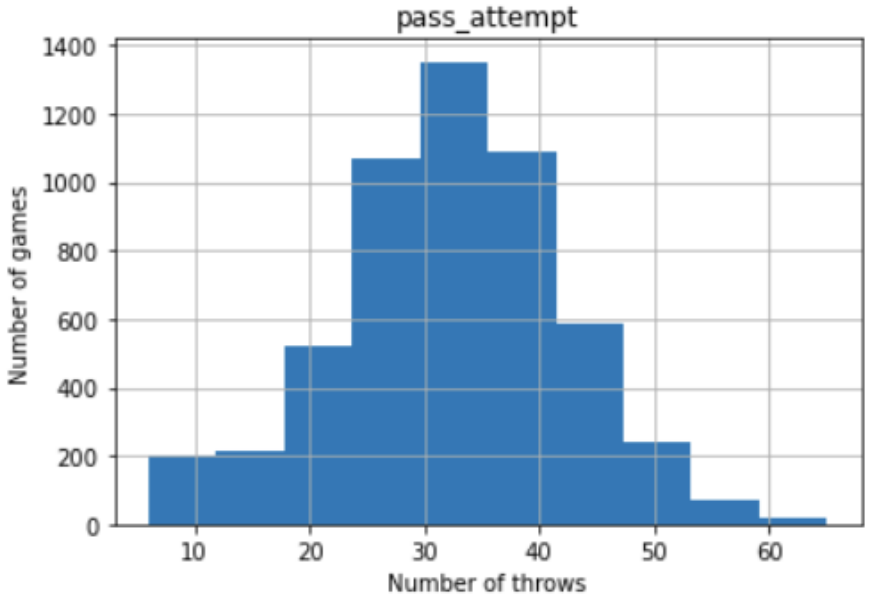
1. What is the winning percentage for quarterbacks in games they throw over 40 times? (This question is to help us know if I should recommend that teams have their quarterbacks throw the ball more.)
2. Which leads to more points: runs or throws? (Knowing if runs or throws generate more points will lead to our recommendations for teams to throw the ball or run more.)

Step 1: Clean and prep the data

First, we removed several unnecessary columns for our analysis, such as player IDs and probabilities (e.g., touchdown probability, extra point probability). Next, we corrected some team names due to changes over the years (e.g., the Chargers changed from San Diego to Los Angles and their abbreviation changed from SD to LAC). Finally, a season spans from the end of one year to the beginning of the next. In order to be able to group by season, we created a column that assigns the season based on the month and year that game is played.

Step 2: Analyze data and get results

To answer question one, I created a sub-data frame with the game ID, the teams in the game, who the quarterback was during the play, the score, and if the play was a pass or not. The data frame was filtered to plays that were pass plays only. Then I grouped by game, quarterback, and the quarterback's team and counted the number of plays to count the number of pass attempts by each quarterback. As shown in the histogram below, the quarterback throws between 20 and 40 times for most games.

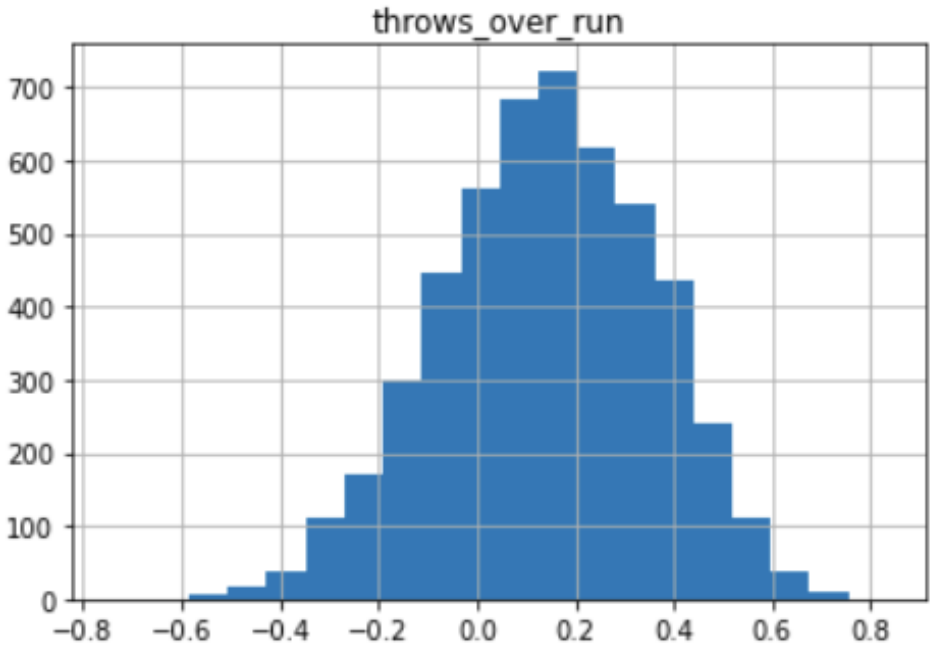


I also created another data frame that included the winner of each game and joined that to my grouped data frame by game ID. I then counted the number of games the quarterback threw over 40 times as well as the number of games they won throwing over 40. Then I created a column of the winning percentages:

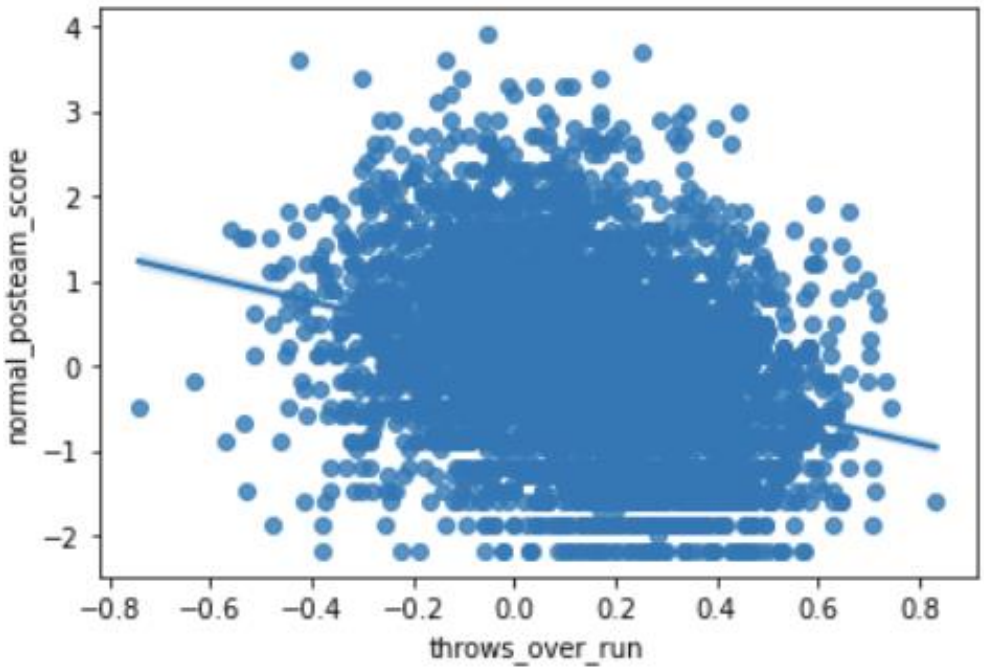
		WinPerc	WinCount	NumofGames
passer_player_name	win_with_40			
T.Brady	True	0.680851	32	47.0
P.Manning	True	0.675000	27	40.0
M.Schaub	True	0.500000	7	14.0
A.Rodgers	True	0.452381	19	42.0
D.Brees	True	0.428571	30	70.0
A.Luck	True	0.400000	14	35.0
P.Rivers	True	0.365854	15	41.0
B.Roethlisberger	True	0.363636	16	44.0
A.Dalton	True	0.346154	9	26.0
J.Flacco	True	0.340909	15	44.0
M.Ryan	True	0.339623	18	53.0
M.Hasselbeck	True	0.333333	5	15.0
R.Fitzpatrick	True	0.318182	7	22.0
M.Sanchez	True	0.307692	4	13.0
E.Manning	True	0.304348	14	46.0
M.Stafford	True	0.288136	17	59.0
K.Cousins	True	0.280000	7	25.0

As shown in this table, only two quarterbacks have a winning percentage over 50% in games they throw over 40: Tom Brady and Payton Manning.

For question two, I created a subset of data with the game ID, each team in the game, number of points scored by each team, the number of plays in that game that were runs and the number of plays in that game that were a pass. I also created a column that had a value representing the number of plays that were throws vs. runs, which is shown below. A negative value means more runs while a positive value means more throws.



With most values depicting throws over runs being positive as shown above, most teams threw the ball more than they ran it. I then plotted the normalized score vs. the throws over runs and added a regression line, as displayed below:



We have a slight decline in our regression line, meaning that running the ball will lead to more points than throwing the ball. To confirm this finding, I ran an ordinary least square regression on the data. Below are the results:

OLS Regression Results

Dep. Variable:	normal_posteam_score	**R-squared:**	0.091
Model:	OLS	**Adj. R-squared:**	0.091
Method:	Least Squares	**F-statistic:**	507.0
Date:	Tue, 26 Apr 2022	**Prob (F-statistic):**	4.62e-107
Time:	19:20:55	**Log-Likelihood:**	-6926.3
No. Observations:	5052	**AIC:**	1.386e+04
Df Residuals:	5050	**BIC:**	1.387e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.1956	0.016	12.240	0.000	0.164	0.227
throws_over_run	-1.3922	0.062	-22.516	0.000	-1.513	-1.271

Omnibus:	81.960	**Durbin-Watson:**	1.842
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	86.033
Skew:	0.306	**Prob(JB):**	2.08e-19
Kurtosis:	3.186	**Cond. No.**	4.70

From our R-Squared value, only 9% of the change in the score can be explained by the number of throws over runs. While running the ball more than throwing does influence a higher score, it is a weak influence. It is statistically significant, so it can be used with other variables to build a stronger model in predicting the number of points.

Conclusion

For question one, out of the 366 quarterbacks in our dataset, only two had winning records when throwing over forty times a game. My recommendation is to not increase the number of throws per game to increase chances of winning, as most quarterbacks have losing records when doing so. In question two, the trend of the data suggests that running the ball leads to more points than throwing. However, with an R-squared of 0.091, it is too low to confidently say that running the ball will lead to more points. I recommend further research with other variables to understand what leads to more points per game.

Skills learned from this project:

1. Organize, subset, and aggregate data to perform analyses
2. Create visuals in *Python*
3. Run and interpret linear regression model

Conclusion

During this program's classes, I learned how to gain insights from data and make recommendations to businesses and other organizations using visuals, statistical analysis, machine learning and other analytical tools. I would like to thank all my professors, classmates, and advisors for helping me through this program and preparing me for my future career in data science.

References

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Aleksandr Glotov, (2022), Car Prices Poland, Version 1, Retrieved February 2022 from https://www.kaggle.com/datasets/aleksandrglotov/car-prices-poland

Max Horowitz, (2017), Detailed NFL Play-by-Play Data 2009-2018, Version 6, Retrieved September 2021 from https://www.kaggle.com/datasets/maxhorowitz/nflplaybyplay2009to2016