

```
/*-----*/
/*  Step 1: Define Library & Import Data          */
/*-----*/

libname mylib "/home/u62449093/project GPU";

proc import datafile="/home/u62449093/project GPU/final_gpu_specs_with_predictions.xlsx"
  out=mylib.gpu_data
  dbms=xlsx replace;
  sheet="Sheet1";
  getnames=yes;
run;

/*-----*/
/*  Step 2: Display Data Structure                */
/*-----*/

proc contents data=mylib.gpu_data; run; /* Display metadata */
proc print data=mylib.gpu_data(obs=10); run; /* Display first 10 rows */

/*-----*/
/*  Step 3: Summary Statistics & Correlation Analysis  */
/*-----*/

proc means data=mylib.gpu_data n mean median std min max;
  var gaming_score productivity_score price_INR memSize gpuClock memClock;
  class releaseYear;
  output out=summary_stats mean= median= std= min= max= / autoname;
run;
proc print data=summary_stats;
  title "Summary Statistics of GPU Dataset";
run;

/* Correlation Analysis */
proc corr data=mylib.gpu_data;
  var gaming_score productivity_score price_INR memSize gpuClock memClock unifiedShader;
  title "Correlation Analysis of GPU Features";
run;

/*-----*/
/*  Step 4: Compute Performance-to-Price & Future-Proof Score */
/*-----*/
data mylib.gpu_scored;
  set mylib.gpu_data;
  Performance_to_Price = (gaming_score + productivity_score) / price_INR;
  Future_Score_Adjusted = Future_Proof_Score * (2025 - releaseYear);
```

```
run;

/*-----*/
/* Step 5: Rank GPUs Based on Performance & Price */
/*-----*/
proc rank data=mylib.gpu_data out=ranked_gpus ties=low;
    var gaming_score productivity_score price_INR;
    ranks gaming_rank productivity_rank price_rank;
run;

proc print data=ranked_gpus(obs=10);
    title "Ranked GPUs Based on Performance & Price";
run;

/*-----*/
/* Step 6: Generate Top 10 GPU Lists */
/*-----*/
/* Top 10 GPUs for Gaming */

proc sort data=mylib.gpu_data out=gaming_top10;
    by descending gaming_score;
run;

proc print data=gaming_top10(obs=10);
    title "Top 10 GPUs for Gaming Performance";
run;

/* Top 10 GPUs for Productivity */
proc sort data=mylib.gpu_data out=productivity_top10;
    by descending productivity_score;
run;

proc print data=productivity_top10(obs=10);
    title "Top 10 GPUs for Productivity Performance";
run;

/* Top 10 Budget GPUs for Gaming */
proc sort data=mylib.gpu_scored out=gaming_budget_top10;
    by descending Performance_to_Price;
run;

proc print data=gaming_budget_top10(obs=10);
    title "Top 10 Budget GPUs for Gaming";
run;
```

```
/* Top 10 Budget GPUs for Productivity */
proc sort data=mylib.gpu_scored out=productivity_budget_top10;
    by descending Performance_to_Price;
run;

proc print data=productivity_budget_top10(obs=10);
    title "Top 10 Budget GPUs for Productivity";
run;

/* Top 10 GPUs with Best Price-to-Performance Ratio */
proc sort data=mylib.gpu_scored out=overall_budget_top10;
    by descending Performance_to_Price;
run;

proc print data=overall_budget_top10(obs=10);
    title "Top 10 GPUs with Best Price-to-Performance Ratio";
run;

/* Top 10 Future-Proof GPUs */
proc sort data=mylib.gpu_data out=future_proof_top10;
    by descending Future_Proof_Score;
run;
proc print data=future_proof_top10(obs=10);
    title "Top 10 Future-Proof GPUs";
run;

/*-----*/
/* Step 7: Performance Trends by Release Year */
/*-----*/
proc means data=mylib.gpu_scored noprint;
    class releaseYear;
    var gaming_score productivity_score;
    output out=yearly_performance mean=gaming_score productivity_score;
run;
proc print data=yearly_performance;
    title "Performance Trends by Release Year";
run;

/*-----*/
/* Step 8: Scatter Plots & Heatmaps */
/*-----*/
proc sgplot data=mylib.gpu_scored;
    scatter x=price_INR y=gaming_score / group=manufacturer;
    reg x=price_INR y=gaming_score;
    title "Price vs Gaming Performance";
run;
```

```
proc sgplot data=mylib.gpu_scored;
  scatter x=price_INR y=productivity_score / group=manufacturer;
  reg x=price_INR y=productivity_score;
  title "Price vs Productivity Performance";
run;

proc sgplot data=mylib.gpu_scored;
  heatmap x=gaming_score y=productivity_score / colorresponse=price_INR colormodel=twilight;
  title "Heatmap: Gaming Score vs Productivity Score with Price Intensity";
run;

/*-----*/
/* Step 9: Predictive Modeling (GPU Pricing & Performance) */
/*-----*/
proc arima data=mylib.gpu_scored;
  identify var=price_INR;
  estimate p=1 q=1;
  forecast lead=5 out=predicted_prices;
run;
proc print data=predicted_prices;
  title "Predicted GPU Prices for Next 5 Years";
run;

/* Regression Model for Gaming Performance */
proc reg data=mylib.gpu_scored;
  model gaming_score = memSize gpuClock memClock unifiedShader memBusWidth;
  title "Regression Model: Predicting Gaming Performance";
run;

/* GLM Model for Productivity Performance */
proc glm data=mylib.gpu_scored;
  model productivity_score = memSize gpuClock memClock unifiedShader memBusWidth / solution;
  title "GLM Model: Predicting Productivity Performance";
run;

/* Logistic Regression for Future-Proof Score */
proc logistic data=mylib.gpu_scored;
  class releaseYear;
  model Future_Proof_Score = gaming_score productivity_score releaseYear / expb;
run;

/* Decision Tree for GPU Price vs Performance */
proc hpsplit data=mylib.gpu_scored;
  model gaming_score = memSize gpuClock memClock unifiedShader memBusWidth price_INR;
  grow variance;
  prune costcomplexity;
run;

/*-----*/
/* Step 10: GPU Clustering */
/*-----*/
```

```
/*-----*/
proc fastclus data=mylib.gpu_scored maxclusters=3 out=clustered_gpus;
    var gaming_score productivity_score price_INR;
run;

proc print data=clustered_gpus(obs=10);
    title "GPU Clustering Based on Performance & Price Segments";
run;

proc sgplot data=clustered_gpus;
    scatter x=gaming_score y=productivity_score / group=Cluster;
    title "GPU Clusters Based on Gaming & Productivity Performance";
run;

/*-----*/
/* Step 11: Additional Visualizations */
/*-----*/
proc sgplot data=mylib.gpu_scored;
    bubble x=price_INR y=gaming_score size=productivity_score / transparency=0.3;
    title "Bubble Chart: Price vs Gaming Score (Bubble Size: Productivity Score)";
run;

proc sgscatter data=mylib.gpu_scored;
    matrix gaming_score productivity_score price_INR memSize gpuClock memClock unifiedShader / diagonal=(histogram);
    title "Pairwise Scatter Matrix of GPU Specifications";
run;

proc sgplot data=mylib.gpu_scored;
    histogram price_INR / scale=count;
    density price_INR / type=kernel;
    title "Distribution of GPU Prices";
run;

proc sgplot data=mylib.gpu_scored;
    vbox productivity_score / category=manufacturer;
    title "Productivity Score Distribution by Manufacturer";
run;

proc sgplot data=mylib.gpu_scored;
    vbox gaming_score / category=manufacturer;
    title "Gaming Score Distribution by Manufacturer";
run;
```

