



Enabling Multilingual Code Understanding: Bridging the Gap in Diverse Programming Environments

Casey Higgins

Atorus Research



Agenda

- Problem Statement
- Approach
- Journey Forward

People



Language Expertise



Domain Knowledge



Innovation

Bridging the Skill Gap



Training



Initiatives



Project Work

Problem Statement



Decrease Language Constraint



Foundation for Innovation and Change Management



Focus on Career Development

Approach



Explain

Analyze

Translate

LLM Ways of Working

- Simplifying Complex Code
- Putting the Code into Categories
- Making the Code Human Readable

```
import React, { useEffect, useRef, useState } from 'react';

const selectedCharacterSet =
  "$@B%8&WM#*oahkbpqwmZ00QLCJUYXzcvunxrjft/|()1{[]?~<>i!lI;:,^`' . :â-";
const characterSetLength = selectedCharacterSet.length;

const calculateCharacter = (
  x: number,
  y: number,
  cols: number,
  rows: number,
  aspect: number,
  time: number,
): string => {
  const timeFactor = time * 0.000_08;
  const size = Math.min(cols, rows);
  const aspectRatio = aspect * 0.2;
  const position = {
    x: ((4 * (x - cols / 6.25)) / size) * aspectRatio,
    y: (5 * (y - rows / 4)) / size,
  };

  const index =
    Math.floor(
      Math.abs(
        Math.cos(position.x * position.x - position.y * position.y) -
          timeFactor,
      ) *
      characterSetLength +
      (Math.floor(x) % 2) * 2,
```

Analyze

Prompt: Explain the SAS Code

Output:

- Setting File Paths
- Closing Open HTML Output and Setting Title
- Setting Options and Output Destination
- Importing Data
- Creating Dummy Variables
- Performing Regression Analysis
- Closing PDF Output

```
options nodate;
ods pdf file="&win_path\slc_proc_reg_example.pdf";
proc import datafile="&csv_file";
out=cars
dbms=csv
replace;
getnames=yes;
run;
data cars_reg ;
set cars ;
if DriveTrain = "All";
then dummy1=1 ;
else dummy1=0 ;
if DriveTrain = "Front";
then dummy1=1 ;
else dummy1=0 ;
if DriveTrain = "Front";
then dummy2=1 ;
else dummy2=0 ;
if Make = "Acura";
then dummy3=1 ;
else dummy3=0 ;
run;
proc reg data = cars ;
model invoice = Cylinders EngineSize Horsepower Length MPG_City
MPG_Highway Weight Wheelbase dummy: / SELECTION=stepwise;
/* The colon after dummy: lists all the dummy variables for MAKE, TYPE,
DRIVETRAIN */
run;
ods pdf close;
```

Explain

Setting File Paths: The script starts by defining file paths for the CSV file containing the data and the output location for the PDF reported generated by the analysis

Closing Open HTML Output and Setting Title: The script closes any open HTML output destination and sets a title for the analysis report

Setting Options and Output Destination: The script specifies not to display the date in the output and sets the output destination for the PDF report

Importing Data: The PROC IMPORT step is used to import the data from the specified CSV file into a SAS dataset. The bdms=csv option indicates the format of the input file, and the getnames=yes tells SAS to use the first row of the CSV as variable names

Creating Dummy Variables: The script then creates a new dataset from the SAS dataset from the import and generates dummy data. Dummy variables are binary variables created for categorical predictors in regression model. This step is preparing the data for the regression analysis by converting the categorical variables into a format that can be used in the model.

Performing Regression Analysis: The PROC REG step performs the regression analysis. The model makes a prediction based on various variables including the dummy variables created in the previous step.

Closing PDF Output: Finally, the script closes the PDF output destination, finalizing the report

Translate

Prompt: Translate the SAS Code

Initial Output:

- **Setting File Paths**
- **Closing Open HTML Output and Setting Title**
- **Setting Options and Output Destination**
- **Importing Data**
- **Creating Dummy Variables**
- **Performing Regression Analysis**
- **Closing PDF Output**

Translated Output:

- **Import Libraries**
- **Load Data**
- **Create Dummy Variables**
- **Prepare Data for Regression**
- **Performing Regression Analysis**

Documentation

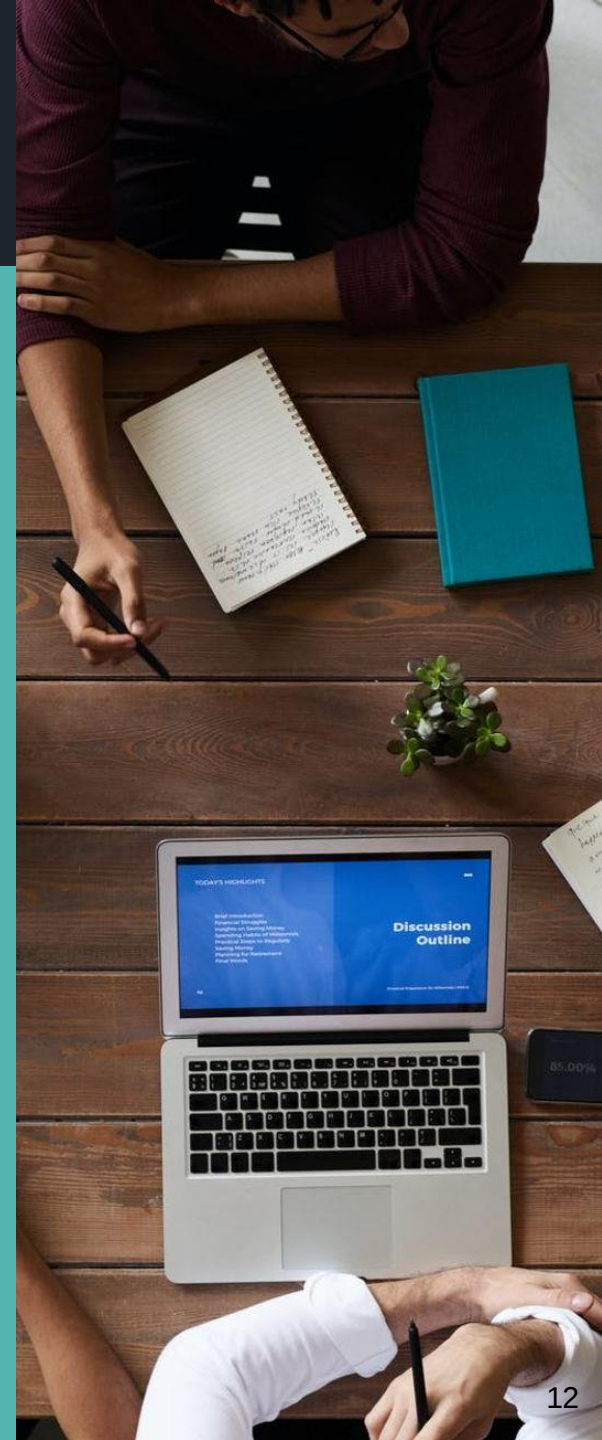
Wiki

Vignettes

Prompt: Write
Explanations in
Comments in Code

Learnings

- **Bad In = Bad Out**
- **Understanding how LLM Interprets the code**
- **Explanations are incomplete**
- **More context is needed for intend usage**
- **Learn by Doing**



Conclusions

Use LLM to Accelerate Upskilling

Foundation for Code Generation

Making Multilingual Accessible Requires Intent

Investigate and Interpreting Inconsistencies with Output

Thank You

Casey.Higgins@atorusresearch.com

Contributions from Patterns.app

