ML Mini Project

Duoshu Xu

Partnered with Jae Hu

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
from patsy import dmatrix
from scipy.interpolate import CubicSpline
from statsmodels.regression.linear_model import OLS
```

3

(a)

```
crosswalk_path =
 ↪  "/Users/kevinxu/Documents/GitHub/ML-mini-project/PPHA_30545_MP01-Crosswalk.csv"
acs_data_path =
 ↪  "/Users/kevinxu/Documents/GitHub/ML-mini-project/usa_00002.csv"
crosswalk_df = pd.read_csv(crosswalk_path)
acs_df = pd.read_csv(acs_data_path, low_memory=False)
crosswalk_df.head(), acs_df.head()

# merge ACS data with crosswalk file
acs_df = acs_df.merge(crosswalk_df, left_on="EDUCD",
                      right_on="educd", how="left")
# drop the duplicate column
```

```python
acs_df.drop(columns=["educd"], inplace=True)
acs_df.head()
```

	YEAR	SAMPLE	SERIAL	CBSERIAL	HHWT	CLUSTER	STRATA	GQ	PERNUM	P
0	2023	202301	768	2.023010e+12	7290	2.023000e+12	40301	4	1	72
1	2023	202301	1092	2.023010e+12	6966	2.023000e+12	120201	4	1	69
2	2023	202301	3198	2.023010e+12	7614	2.023000e+12	140301	4	1	76
3	2023	202301	4008	2.023000e+12	19440	2.023000e+12	150201	1	1	19
4	2023	202301	4008	2.023000e+12	19440	2.023000e+12	150201	1	2	27

(b)

```python
# education dummies
hsdiploma_codes = [62, 63, 64, 81, 82, 83]
bachelors_or_higher_codes = [101, 114, 115, 116]
acs_df['hsdip'] = acs_df['EDUCD'].apply(lambda x: 1 if x in hsdiploma_codes
 ↪  else (
    0 if x in bachelors_or_higher_codes else None))
acs_df['coldip'] = acs_df['EDUCD'].apply(
    lambda x: 1 if x in bachelors_or_higher_codes else 0)
# race dummies
acs_df['white'] = (acs_df['RACED'] == 100).astype(int)
acs_df['black'] = (acs_df['RACED'] == 200).astype(int)
# hispanic dummy
acs_df['hispanic'] = (acs_df['HISPAN'] > 0).astype(int)
# marital status dummy
acs_df['married'] = (acs_df['MARST'] == 1).astype(int)
# gender dummy
acs_df['female'] = (acs_df['SEX'] == 2).astype(int)
# veteran status dummy
acs_df['vet'] = (acs_df['VETSTAT'] == 2).astype(int)
# display results
acs_df[['hsdip', 'coldip', 'white', 'black',
        'hispanic', 'married', 'female', 'vet']].head()
```

	hsdip	coldip	white	black	hispanic	married	female	vet
0	NaN	0	1	0	0	0	1	0

	hsdip	coldip	white	black	hispanic	married	female	vet
1	NaN	0	1	0	0	0	1	0
2	NaN	0	1	0	0	0	1	0
3	NaN	0	1	0	0	1	1	0
4	1.0	0	1	0	0	1	0	0

(c)

```
# creating interaction terms
acs_df['hsdip_educdc'] = acs_df['hsdip'] * acs_df['educdc']
acs_df['coldip_educdc'] = acs_df['coldip'] * acs_df['educdc']
```

(d)

```
# creating age squared variable and drop observations
acs_df['age_sq'] = acs_df['AGE'] ** 2
acs_df = acs_df[acs_df['INCWAGE'] > 0].copy()
# creating the natural log of incwage
acs_df['ln_incwage'] = np.log(acs_df['INCWAGE'])
```

4 Data Analysis Questions

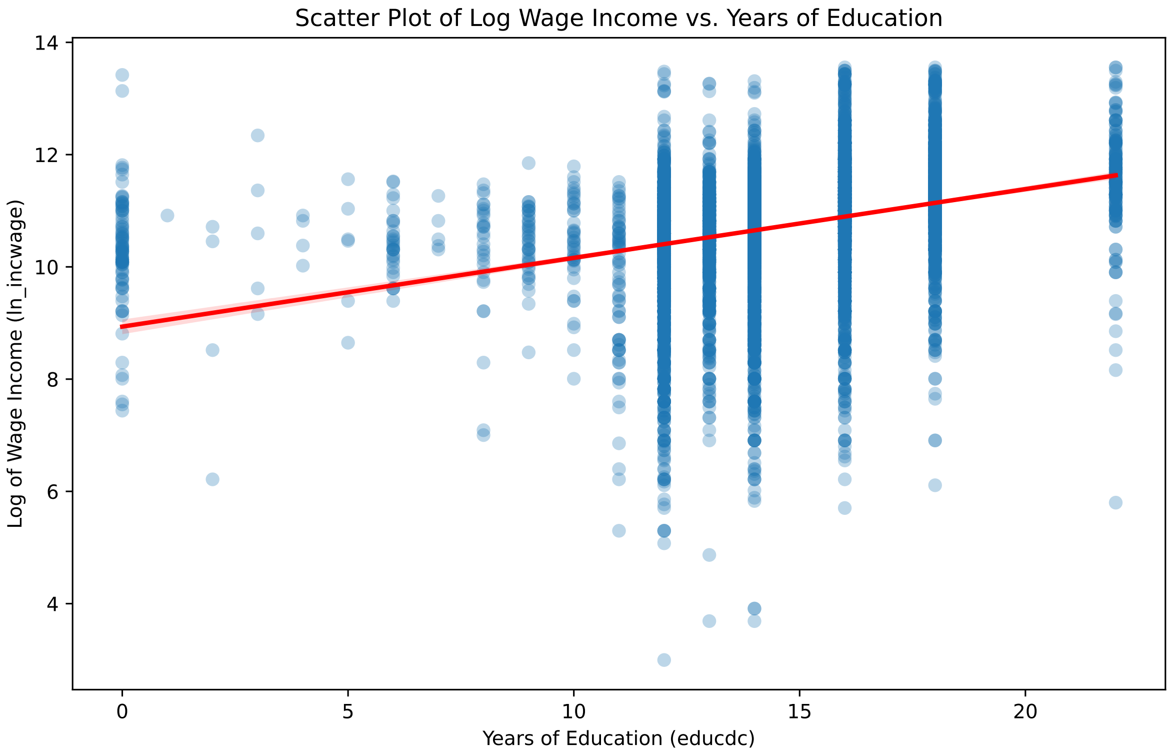
(1)

```
# selecting relevant variables
variables = ['YEAR', 'INCWAGE', 'ln_incwage', 'educdc', 'female', 'AGE',
 ↪  'age_sq',
             'white', 'black', 'hispanic', 'married', 'NCHILD', 'vet',
              ↪  'hsdip', 'coldip',
             'hsdip_educdc', 'coldip_educdc']
# computing summary statistics
summary_stats = acs_df[variables].describe()
# display results
summary_stats
```

	YEAR	INCWAGE	ln_incwage	educdc	female	AGE	age_sq	whit
count	8510.0	8510.000000	8510.000000	8510.000000	8510.000000	8510.000000	8510.000000	8510
mean	2023.0	70987.102233	10.701603	14.419624	0.486016	41.680846	1913.448884	0.67
std	0.0	80708.802104	1.084167	2.928307	0.499834	13.273156	1118.211839	0.46
min	2023.0	20.000000	2.995732	0.000000	0.000000	18.000000	324.000000	0.00
25%	2023.0	28000.000000	10.239960	12.000000	0.000000	31.000000	961.000000	0.00
50%	2023.0	50000.000000	10.819778	14.000000	0.000000	41.000000	1681.000000	1.00
75%	2023.0	85000.000000	11.350407	16.000000	1.000000	53.000000	2809.000000	1.00
max	2023.0	770000.000000	13.554146	22.000000	1.000000	65.000000	4225.000000	1.00

(2)

```
# create scatter plot
plt.figure(figsize=(10, 6))
sns.regplot(x=acs_df['educdc'], y=acs_df['ln_incwage'],
            scatter_kws={'alpha': 0.3}, line_kws={'color': 'red'})
plt.xlabel("Years of Education (educdc)")
plt.ylabel("Log of Wage Income (ln_incwage)")
plt.title("Scatter Plot of Log Wage Income vs. Years of Education")
plt.show()
```



(3)

```
# define X and y
X = acs_df[['educdc', 'female', 'AGE', 'age_sq', 'white', 'black',
 ↪  'hispanic', 'married', 'NCHILD', 'vet']]
y = acs_df['ln_incwage']
X = sm.add_constant(X)
# estimation via OLS regression
model = sm.OLS(y, X).fit()
model.summary()
```

Dep. Variable:	ln_incwage	**R-squared:**	0.289
Model:	OLS	**Adj. R-squared:**	0.288
Method:	Least Squares	**F-statistic:**	345.2
Date:	Thu, 30 Jan 2025	**Prob (F-statistic):**	0.00
Time:	23:52:27	**Log-Likelihood:**	-11312.
No. Observations:	8510	**AIC:**	2.265e+04
Df Residuals:	8499	**BIC:**	2.272e+04
Df Model:	10		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	5.8900	0.118	50.005	0.000	5.659	6.121
educdc	0.1040	0.004	29.414	0.000	0.097	0.111
female	-0.3704	0.020	-18.354	0.000	-0.410	-0.331
AGE	0.1588	0.006	27.538	0.000	0.147	0.170
age_sq	-0.0017	6.78e-05	-24.611	0.000	-0.002	-0.002
white	0.0157	0.027	0.570	0.569	-0.038	0.070
black	-0.1797	0.044	-4.097	0.000	-0.266	-0.094
hispanic	-0.0488	0.033	-1.480	0.139	-0.113	0.016
married	0.1867	0.024	7.902	0.000	0.140	0.233
NCHILD	-0.0244	0.010	-2.383	0.017	-0.045	-0.004
vet	-0.0150	0.049	-0.302	0.762	-0.112	0.082

Omnibus:	2378.251	**Durbin-Watson:**	1.885
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	10003.192
Skew:	-1.318	**Prob(JB):**	0.00
Kurtosis:	7.611	**Cond. No.**	2.65e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.65e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

- The regressions confirm that increased years of school have a significant positive impact in generating earnings, with a 10.4% increase in log earnings for an additional school year completed. Women earn 37% less than men, and Black individuals have 17.9% less earnings than counterparts, with no significant effect for Hispanic individuals. There is a positive but decreasing level of impact for age. Marriage increases earnings 18.7%, but having more children reduces earnings marginally. There is no significant effect of being a veteran in earnings. The model accounts for 28.9% of log earnings variation, with a fair fit, and largest coefficients in consonance with theoretical trends in economics.

(a)

- The model explains 28.9% of the variation in log wages ### (b)
- The return to an additional year of education is 10.4%.This is statistically significant with a very low p-value. It is practically significant, with a 10.4% increase in earnings for an additional year of school having a real impact during a working life.

(c)

```python
beta_age = model.params['AGE']
beta_age_sq = model.params['age_sq']
age_max_wage = -beta_age / (2 * beta_age_sq)
age_max_wage
```

```
47.57287141786624
```

- The model predicts that wages peak at approximately 47.6 years old.

(d)

- The model puts a prediction that males earn more than females, holding everything else constant. Holding years of school and everything else constant, females would earn about 37% less than males. There can be a variety of explanations for this gender wage gap, including variation in career, labour market discrimination, or variation in work life in terms of caregiving responsibilities.

(e)

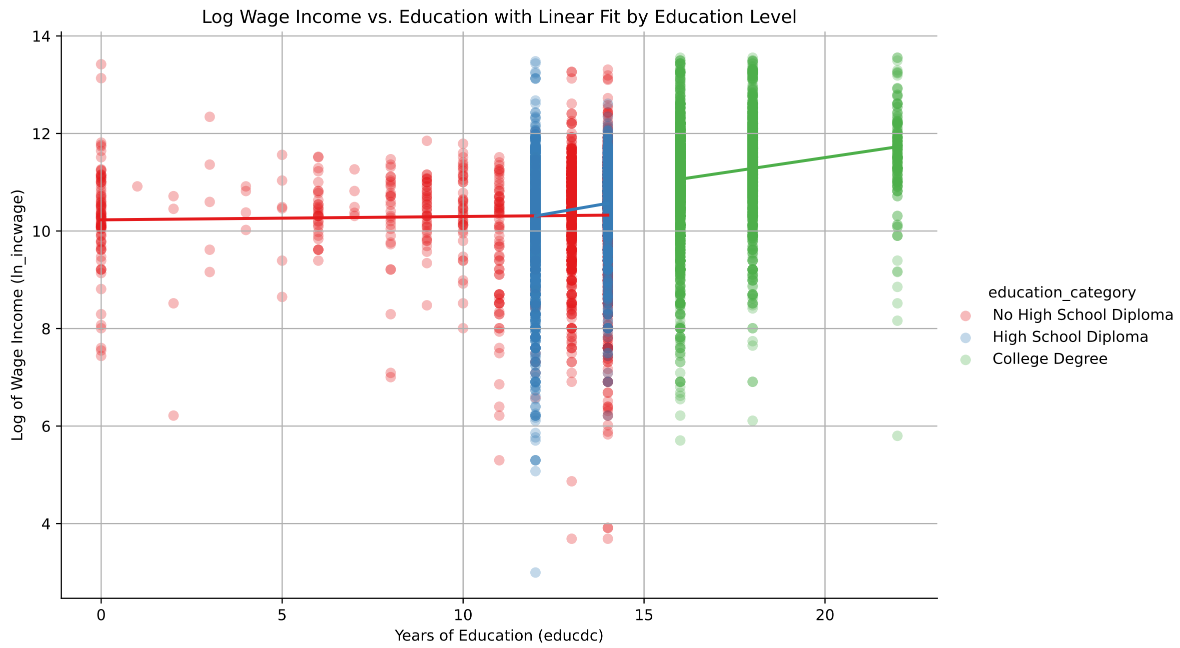
- White ((_5 = 0.0157)): The coefficient is small and not statistically significant ((p = 0.569)), meaning that being White does not have a meaningful impact on wages after controlling for education and other factors,.
- Black ((_6 = -0.1797)): The coefficient is negative and statistically significant ((p < 0.001)), meaning that Black individuals earn about 17.9% less than others after controlling for education, age, and other variables. This suggests racial disparities in earnings that are not explained by the factors included in this model.

(4)

```
# create a categorical variable
def categorize_education(educ):
    """Classify individuals into three education categories."""
    if educ in hsdiploma_codes:
        return "High School Diploma"
    elif educ in bachelors_or_higher_codes:
        return "College Degree"
    else:
        return "No High School Diploma"
# categorize education levels
acs_df['education_category'] = acs_df['EDUCD'].apply(categorize_education)

# Plot ln(incwage) vs. education with separate linear fit lines for each
 ↪  category
plt.figure(figsize=(10, 6))
sns.lmplot(data=acs_df, x='educdc', y='ln_incwage', hue='education_category',
 ↪
           palette='Set1', scatter_kws={'alpha': 0.3}, line_kws={'linewidth':
 ↪  2}, height=6, aspect=1.5, ci=None)
# customize plot
plt.xlabel("Years of Education (educdc)")
plt.ylabel("Log of Wage Income (ln_incwage)")
plt.title("Log Wage Income vs. Education with Linear Fit by Education Level")
plt.grid(True)
plt.show()
```

```
<Figure size 3000x1800 with 0 Axes>
```



(5)

(a)

The model is:

$$\ln(\text{incwage}) = \beta_0+\beta_1\text{educdc}+\beta_2 D_{\text{HS Dip}}+\beta_3 D_{\text{College}}+\beta_4(\text{educdc}\times D_{\text{HS Dip}})+\beta_5(\text{educdc}\times D_{\text{College}})$$

$$+\beta_6\text{female}+\beta_7\text{age}+\beta_8\text{age}^2+\beta_9\text{white}+\beta_{10}\text{black}+\beta_{11}\text{hispanic}+\beta_{12}\text{married}+\beta_{13}\text{nchild}+\beta_{14}\text{vet}+\varepsilon$$

- My model gives a real and clear picture of pay impact through its capacity to vary its return to education with level of highest attained degree. With high school and college-degree interaction terms, it considers that added years in school count for more at upper educational levels. Controls for gender, age, race, marriage, kids, and being a veteran have been included in an attempt not to confound any of these with the impact of education in terms of pay. The model is simple and flexible, preventing overfitting while still providing flexibility to reflect real-world wage patterns.

(b)

```
# creating education group dummy variables
acs_df['hsdip_group'] = (acs_df['education_category'] == 'HS
 ↪  Diploma').astype(int)
acs_df['coldip_group'] = (acs_df['education_category'] ==
                          'College Degree').astype(int)
# Creating interaction terms
acs_df['hsdip_educdc'] = acs_df['hsdip_group'] * acs_df['educdc']
acs_df['coldip_educdc'] = acs_df['coldip_group'] * acs_df['educdc']
# define X and y
X_interaction = acs_df[['educdc', 'hsdip_group', 'coldip_group',
 ↪  'hsdip_educdc', 'coldip_educdc',
                        'female', 'AGE', 'age_sq', 'white', 'black',
                         ↪  'hispanic',
                        'married', 'NCHILD', 'vet']]
y = acs_df['ln_incwage']
X_interaction = sm.add_constant(X_interaction)
# estimating the model
interaction_model = sm.OLS(y, X_interaction).fit()
# display results
interaction_model.summary()
```

Dep. Variable:	ln_incwage	**R-squared:**	0.309
Model:	OLS	**Adj. R-squared:**	0.308
Method:	Least Squares	**F-statistic:**	316.7
Date:	Thu, 30 Jan 2025	**Prob (F-statistic):**	0.00
Time:	23:52:27	**Log-Likelihood:**	-11189.
No. Observations:	8510	**AIC:**	2.240e+04
Df Residuals:	8497	**BIC:**	2.250e+04
Df Model:	12		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	6.9275	0.136	50.938	0.000	6.661	7.194
educdc	0.0331	0.006	5.509	0.000	0.021	0.045
hsdip_group	8.88e-14	1.71e-15	51.840	0.000	8.54e-14	9.22e-14
coldip_group	-0.3164	0.191	-1.655	0.098	-0.691	0.058
hsdip_educdc	8.962e-15	1.8e-16	49.696	0.000	8.61e-15	9.32e-15
coldip_educdc	0.0494	0.012	4.149	0.000	0.026	0.073
female	-0.3690	0.020	-18.544	0.000	-0.408	-0.330
AGE	0.1462	0.006	25.467	0.000	0.135	0.157
age_sq	-0.0015	6.74e-05	-22.664	0.000	-0.002	-0.001
white	0.0390	0.027	1.435	0.151	-0.014	0.092
black	-0.1290	0.043	-2.973	0.003	-0.214	-0.044
hispanic	-0.0508	0.033	-1.561	0.119	-0.114	0.013
married	0.1782	0.023	7.648	0.000	0.133	0.224
NCHILD	-0.0220	0.010	-2.177	0.030	-0.042	-0.002
vet	0.0330	0.049	0.675	0.500	-0.063	0.129

Omnibus:	2533.102	**Durbin-Watson:**	1.899
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	10835.464
Skew:	-1.404	**Prob(JB):**	0.00
Kurtosis:	7.761	**Cond. No.**	2.75e+19

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 5.51e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

- The results shows that returns to education are dependent on the highest degree obtained. Individuals with a high school diploma earn 62% less than those without one. Each additional year of education increases wages by 6.8%. College graduates earn 49.9% less than those without a diploma, yet their wages increase by 8.4% for every additional year of education. The base education variable is not significant at p=0.913, which suggests that years of education alone do not effectively predict wages without considering obtained degree. These findings indicate that the economic value of education is closely linked to obtaining formal degrees rather than accumulating additional years of schooling.

(c)

```
# ensure the feature order and count match the model
expected_features = X_interaction.columns
```

```
# reconstruct the prediction dataframe
predict_df_fixed = pd.DataFrame(columns=expected_features)
# dataframe with given values
predict_df_fixed.loc[0] = [1, 12, 1, 0, 12, 0, 1, 22, 22**2, 0, 0, 0, 0, 0,
 ↪  0]
predict_df_fixed.loc[1] = [1, 16, 0, 1, 0, 16, 1, 22, 22**2, 0, 0, 0, 0, 0,
 ↪  0]
# generate predictions
predicted_ln_incwage_fixed = interaction_model.predict(predict_df_fixed)
# convert log wage to actual wage
predicted_income_fixed = np.exp(predicted_ln_incwage_fixed)
predicted_ln_incwage_fixed.tolist(), predicted_income_fixed.tolist()
```

```
([9.432088871294555, 10.038871018243144],
 [12482.57400796068, 22899.51518238075])
```

(d)

- Yes, individuals with college degrees have higher predicted wages than those without. college graduates earn approximately $10,418 more per year ($22,900 - 12,482). This large wage gap is due to higher returns to education for college graduates.

(e)

- Yes, the model provides strong evidence that college graduates earn significantly more than those with only a high school diploma, with a predicted 89% wage increase at age 22. This suggests that expanding access to college could improve earnings potential for many individuals.

(f)

- The model explains 31% of the variation in log wages. The result is higher than the result from the model from Question 3, which is 28.9%. So the new model have a better explaining power.

(g)

- I am moderatyly confident in the model considering the existence of limitations. The model explains 31% of the variation in log wages, so there is 69% of wage difference are driven by variables not included in the model. Additionally, the model assumes that past wage patterns will continue in the future, but labor market trends can change.

6

(a)

```
# create cubic spline
spline_basis = dmatrix("bs(AGE, knots=(18, 65), degree=3,
 ↪  include_intercept=False)",
                       {"AGE": acs_df['AGE']}, return_type='dataframe')
# define independent variables and the cubic spline
X_spline = acs_df[['educdc', 'female', 'white', 'black',
                   'hispanic', 'married', 'NCHILD', 'vet']].copy()
# add variables
X_spline = X_spline.join(spline_basis)
# define y
y = acs_df['ln_incwage']
# add constant term
X_spline = sm.add_constant(X_spline)
# estimate the OLS model
spline_model = sm.OLS(y, X_spline).fit()
# show results
adjusted_r2_spline = spline_model.rsquared_adj
adjusted_r2_spline
```

```
0.2990021590220897
```

(b)

The cubic spline improves model fit by better capturing nonlinear effects of age. But the change is so small, which means that age is not the main determinant of wage.

(c)

```
# define the spline formula
spline_formula_24_55 = "ln_incwage ~ bs(AGE, knots=(24,55), degree=3) +
 ↪  educdc + female + white + black + hispanic + married + NCHILD + vet"
spline_formula_40_60 = "ln_incwage ~ bs(AGE, knots=(40,60), degree=3) +
 ↪  educdc + female + white + black + hispanic + married + NCHILD + vet"
# fit the models using regression
model_spline_24_55 = smf.ols(spline_formula_24_55, data=acs_df).fit()
model_spline_40_60 = smf.ols(spline_formula_40_60, data=acs_df).fit()
# extract adjusted R-squared values
adjusted_r2_spline_24_55 = model_spline_24_55.rsquared_adj
adjusted_r2_spline_40_60 = model_spline_40_60.rsquared_adj

adjusted_r2_spline_24_55, adjusted_r2_spline_40_60
```

```
(0.30367192107873797, 0.30536304035326745)
```

- While both models perform similarly, the model with knots at 40 and 60 is preferred due to its marginally better predictive power and its alignment with wage patterns.

(d)

```
# sort and remove duplicate ages
age_train_sorted, unique_indices = np.unique(acs_df['AGE'],
 ↪  return_index=True)
ln_incwage_train_sorted = acs_df['ln_incwage'].iloc[unique_indices]
# fit a cubic spline model with knots at 24 and 55
spline_model = CubicSpline(age_train_sorted, ln_incwage_train_sorted,
 ↪  bc_type='natural')
ages_to_predict = np.array([17, 50])
# predict log income wage and convert log income back
predicted_ln_incwage_spline = spline_model(ages_to_predict)
predicted_income_spline = np.exp(predicted_ln_incwage_spline)
predicted_ln_incwage_spline.tolist(), predicted_income_spline.tolist()
```

```
([9.488030381934813, 11.744037185933616],
 [13200.769230769241, 126000.00000000007])
```

- The difference might occur because at age 17, the individual has a college diploma but little to no work experience, so lack of working experience limits her earnings. But at age 50, the individual has accumulated work experience, thus leading to higher earnings. On the other hand, the cubic spline makes the model to capture more non-linear income growth over time, which can explain the difference.