In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [2]:
```
df = pd.read_csv("credit.csv")
df.head()
```

Out[2]:

	Income	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
0	14.891	3606	283	2	34	11	Male	No	Yes	Caucasian	333
1	106.025	6645	483	3	82	15	Female	Yes	Yes	Asian	903
2	104.593	7075	514	4	71	11	Male	No	No	Asian	580
3	148.924	9504	681	3	36	11	Female	No	No	Asian	964
4	55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331

In [3]:
```
x = df.drop("Balance", axis=1)
y = df["Balance"]
x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size=0.2, random_state=42
)
```

In [4]:
```
try:
    test_model = LinearRegression().fit(x_train, y_train)
except Exception as e:
    print("Error!:", e)
```

```
Error!: could not convert string to float: 'Female'
```

In [5]:
```
df.dtypes
```

Out[5]:
```
Income       float64
Limit          int64
Rating         int64
Cards          int64
Age            int64
Education      int64
Gender        object
Student       object
Married       object
Ethnicity     object
Balance        int64
dtype: object
```

In [6]:
```
numeric_features = ["Income", "Limit", "Rating", "Cards", "Age", "Education"]
model1 = LinearRegression().fit(x_train[numeric_features], y_train)

train_score = model1.score(x_train[numeric_features], y_train)
test_score = model1.score(x_test[numeric_features], y_test)
print("Train R2:", train_score)
print("Test R2:", test_score)
```

```
Train R2: 0.8884475002185803
Test R2: 0.8212531865399557
```

In [7]:
```
print(
    "In the train data, Ethnicity takes on the values:",
    list(x_train["Ethnicity"].unique()),
)
```

```
In the train data, Ethnicity takes on the values: ['Asian', 'Caucasian', 'African American']
```

In [8]:
```
x_train_design = pd.get_dummies(x_train, drop_first=True)
x_test_design = pd.get_dummies(x_test, drop_first=True)
x_train_design.head()
```

Out[8]:

	Income	Limit	Rating	Cards	Age	Education	Gender_Female	Student_Yes	Married_Yes	Ethnicity_Asian	Ethnicity_Caucasian
3	148.924	9504	681	3	36	11	1	0	0	1	0
18	49.570	6384	448	1	28	9	1	0	1	1	0
202	21.038	1448	145	2	58	13	1	0	1	0	1
250	10.363	2430	191	2	47	18	1	0	1	1	0
274	30.550	5869	439	5	81	9	1	0	0	0	0

In [9]:
```
x_train_design.dtypes
```

Out[9]:
```
Income                float64
Limit                   int64
Rating                  int64
Cards                   int64
Age                     int64
Education               int64
Gender_Female           uint8
Student_Yes             uint8
Married_Yes             uint8
Ethnicity_Asian         uint8
Ethnicity_Caucasian     uint8
dtype: object
```

In [10]:
```
model2 = LinearRegression().fit(x_train_design, y_train)

train_score = model2.score(x_train_design, y_train)
test_score = model2.score(x_test_design, y_test)
print("Train R2:", train_score)
print("Test R2:", test_score)
```

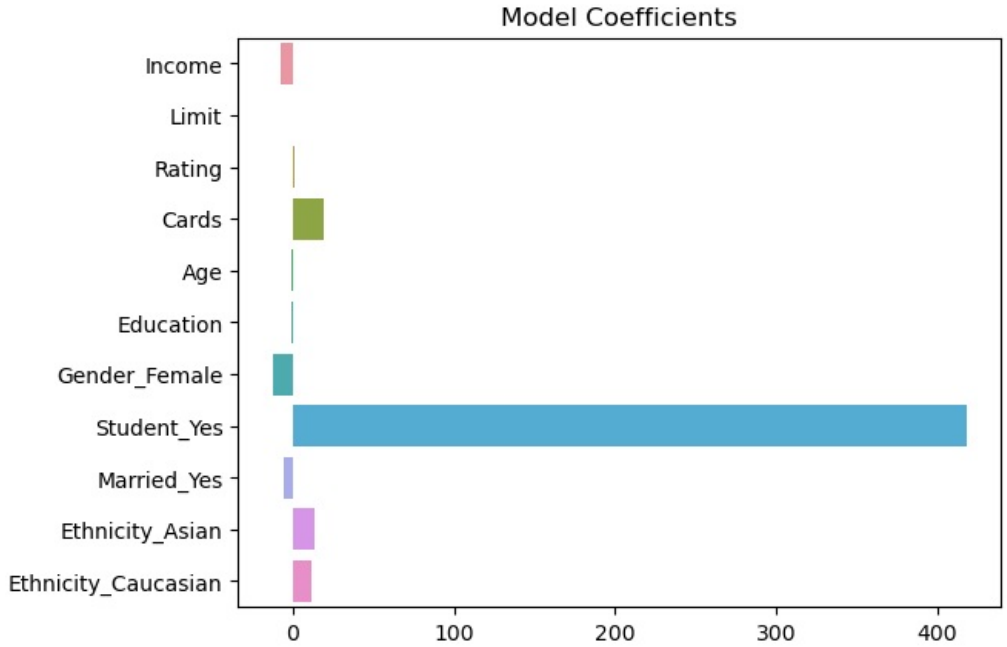
```
Train R2: 0.9552255011405162
Test R2: 0.9522674050276462
```

In [11]:
```
coefs = pd.DataFrame(model2.coef_, index=x_train_design.columns, columns=["beta_value"])
coefs
```

Out[11]:

	beta_value
Income	-7.554059
Limit	0.198058
Rating	0.971292
Cards	19.123826
Age	-0.603695
Education	-1.078674
Gender_Female	-12.493725
Student_Yes	418.435398
Married_Yes	-6.150365
Ethnicity_Asian	13.295548
Ethnicity_Caucasian	11.517153

In [12]:
```
sns.barplot(data=coefs.T, orient="h").set(title="Model Coefficients");
```



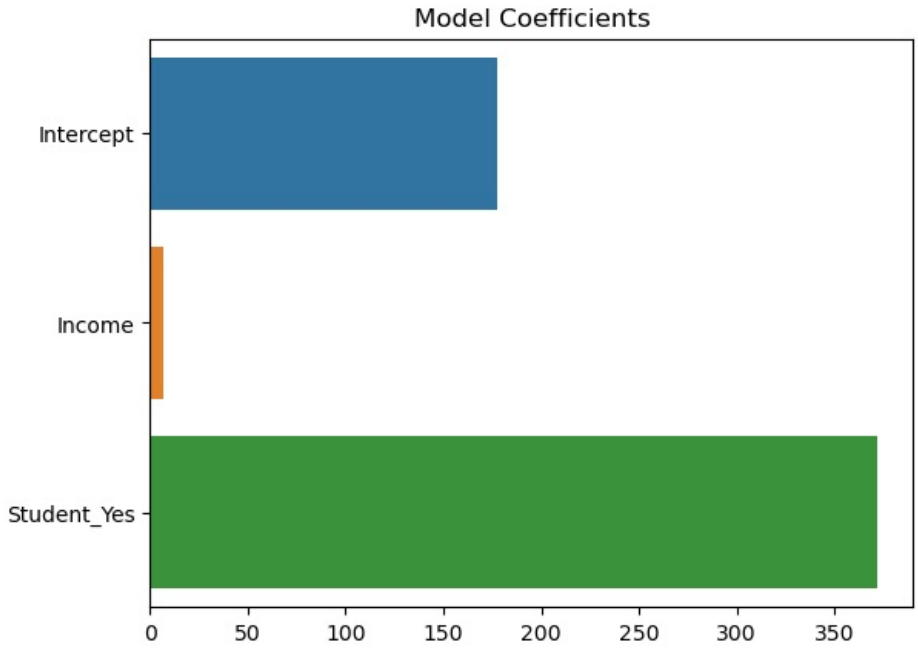
In [13]:
```
best_cat_feature = "Student_Yes"

features = ["Income", best_cat_feature]
model3 = LinearRegression()
model3.fit(x_train_design[features], y_train)

beta0 = model3.intercept_
beta1 = model3.coef_[features.index("Income")]
beta2 = model3.coef_[features.index(best_cat_feature)]

coefs = pd.DataFrame(
    [beta0, beta1, beta2], index=["Intercept"] + features, columns=["beta_value"]
```

```
)
coefs
```

Out[13]:

	beta_value
Intercept	177.658909
Income	6.773090
Student_Yes	371.895694

In [14]:
```
sns.barplot(data=coefs.T, orient="h").set(title="Model Coefficients");
```



In [15]:
```
x_space = np.linspace(x["Income"].min(), x["Income"].max(), 1000)

y_hat_yes = beta0 + beta1 * x_space + 1 * beta2
y_hat_no = beta0 + beta1 * x_space + 0 * beta2

ax = sns.scatterplot(
    data=pd.concat([x_train_design, y_train], axis=1),
    x="Income",
    y="Balance",
    hue=best_cat_feature,
    alpha=0.8,
)
ax.plot(x_space, y_hat_yes)
ax.plot(x_space, y_hat_no);
```