

# **IIIT - BANGALORE**

## **BFSI CREDIT RISK CASE**

### **STUDY**

By:

Shubham Gupta, Sarswati Hanmant Salunkhe & Shubham Rustagi

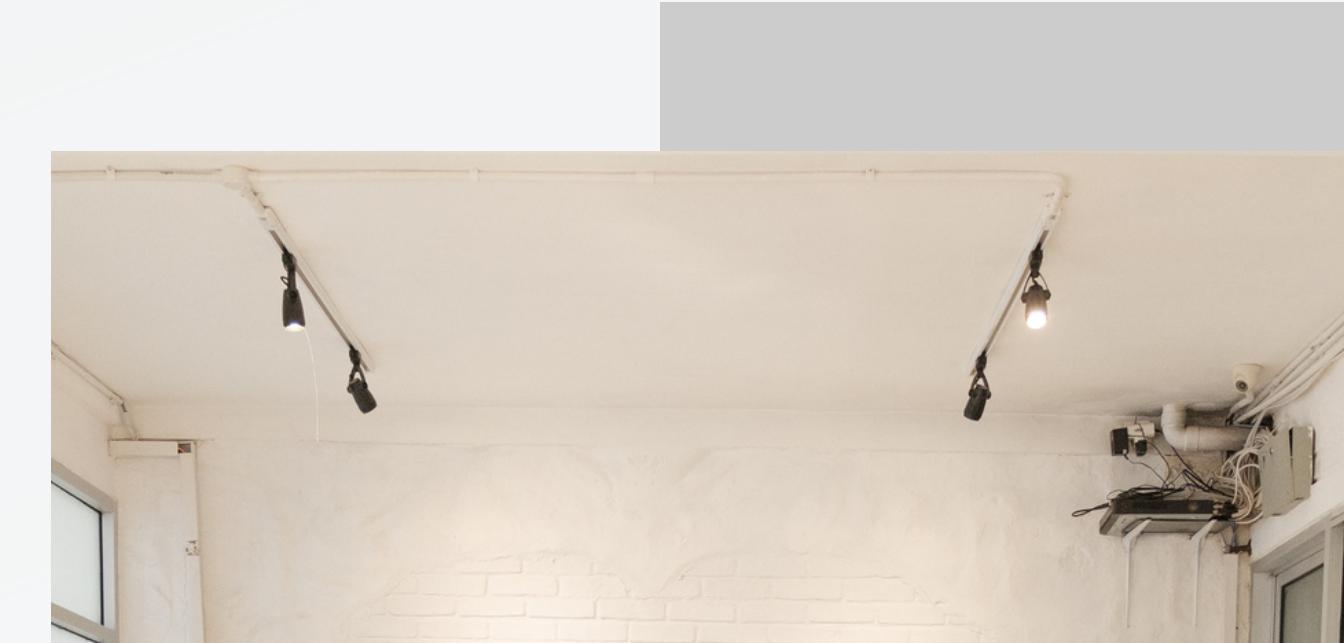
# OBJECTIVE



Focussing on the Loss Given Default (LGD) component of the ECL computation



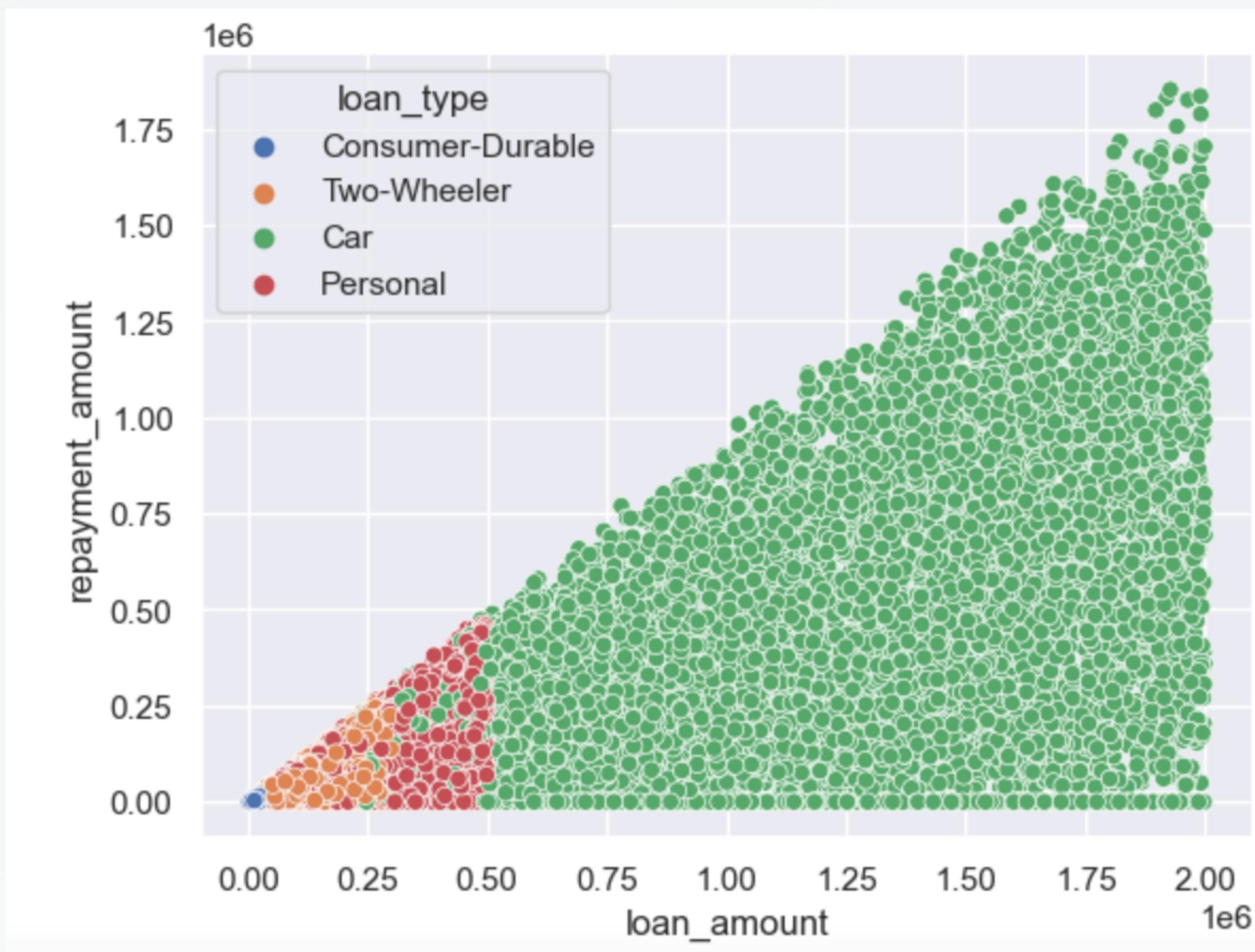
Building a statistical model to estimate borrowers' LGD.



# ANALYSIS

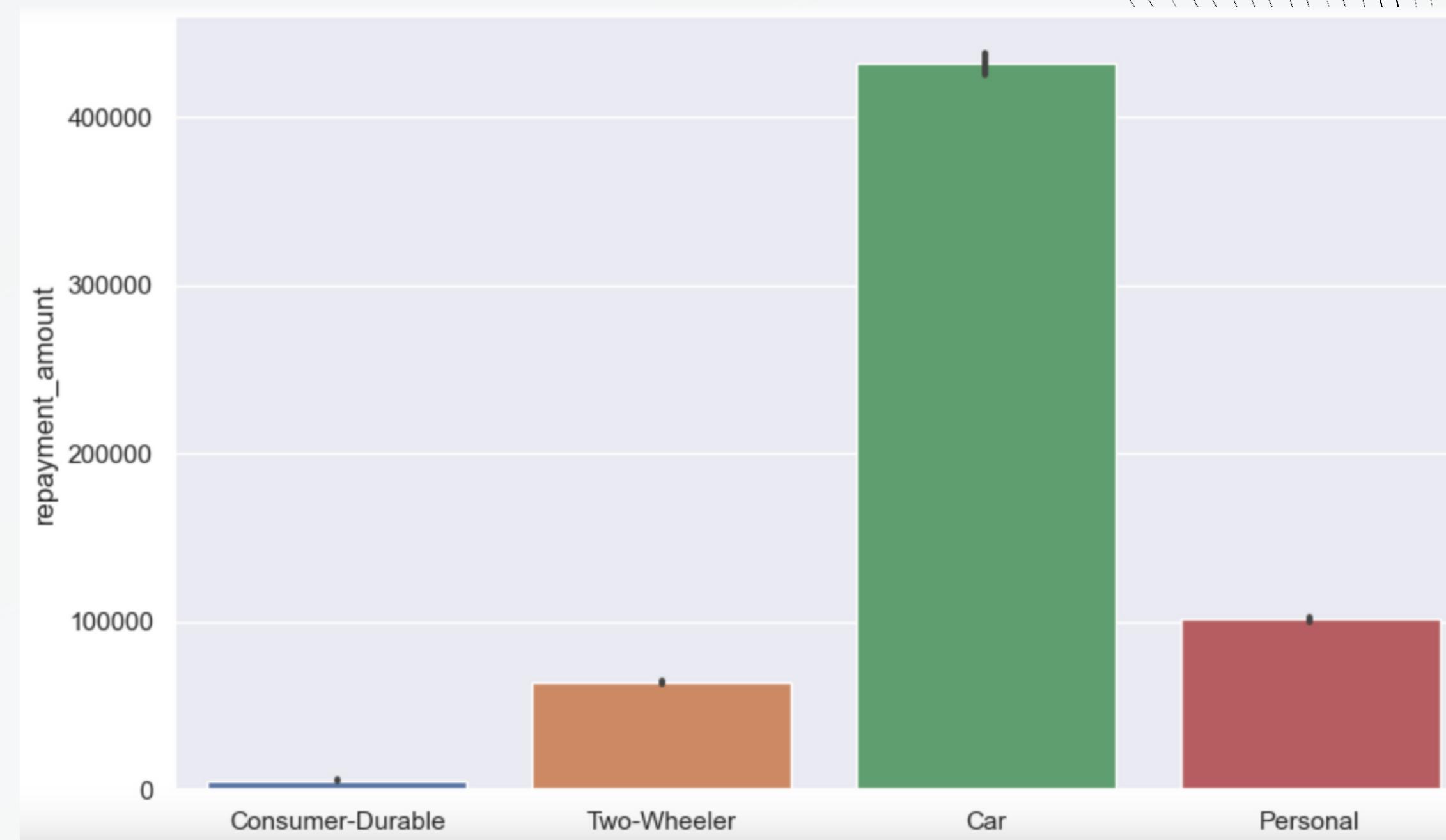
# REPAYMENT VS LOAN AMOUNT

- The majority of the loan amount is for cars followed by personal
- Two-wheelers have low loan amounts with less repayment amount



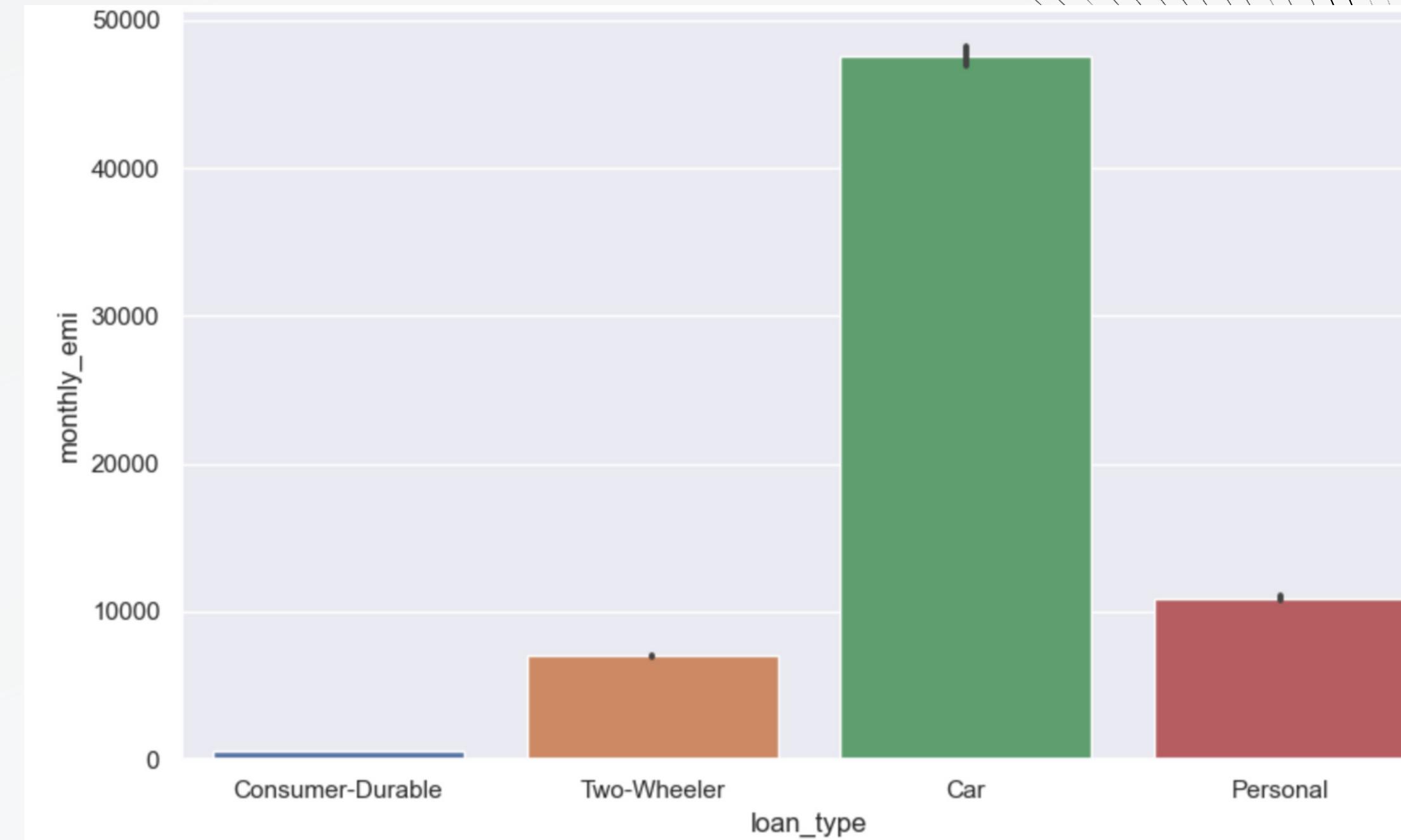
# REPAYMENT AMOUNT VS LOAN TYPE

- We can observe that the repayment amount and the loan amount are highest for cars.
- This is followed by personal



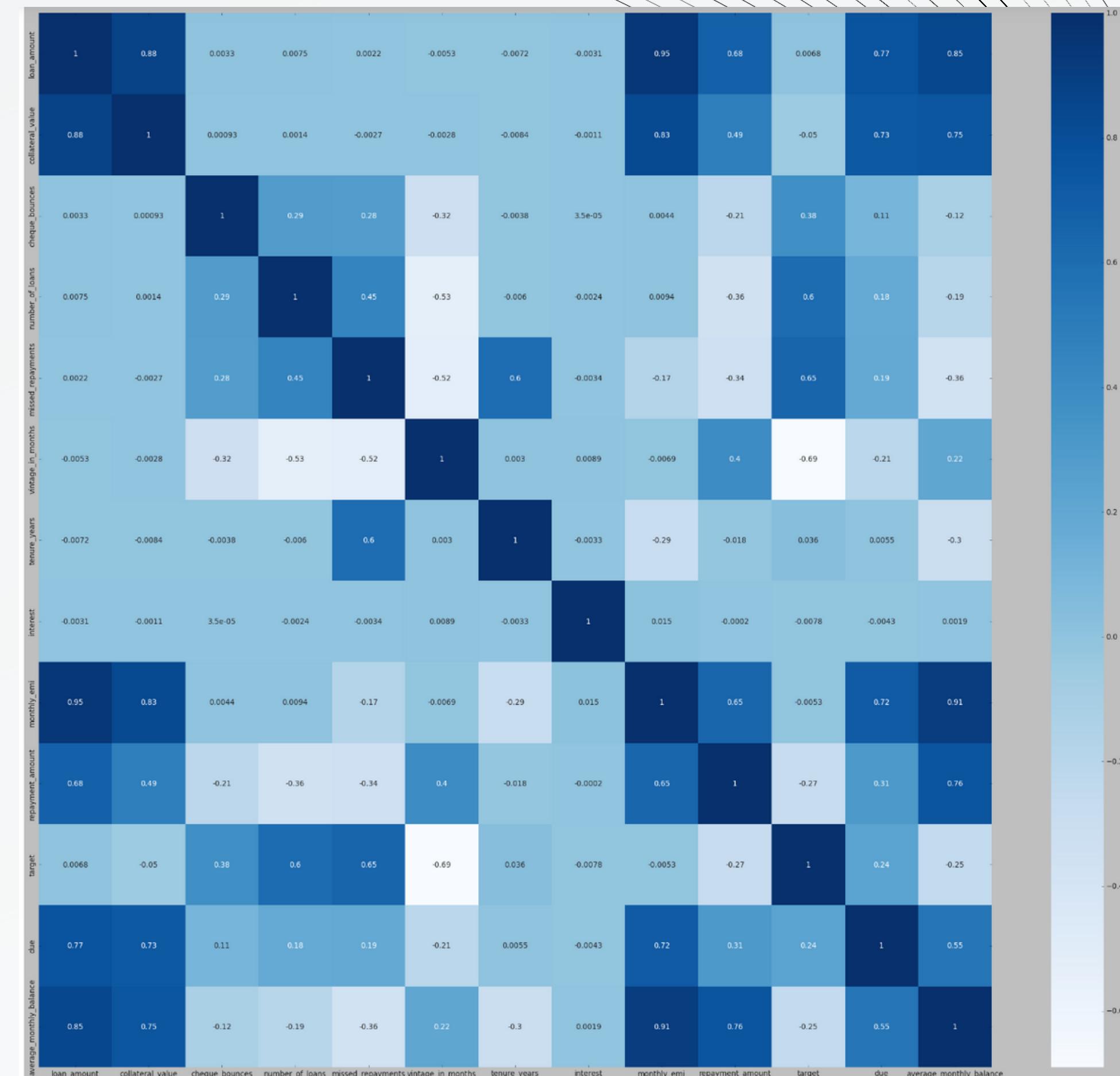
# MONTHLY EMI VS LOAN TYPE

- While the loan amount is highest of cars, we can observe that the monthly EMI is also the highest.



# HEATMAP

- Collateral value, monthly EMI, and loan amount are highly correlated.



# MODEL BUILDING USING RFE

- The model gave us a r2 score of 77%
- This represents that the regression model is fitting good on the data

```
from sklearn.linear_model import *
from sklearn import metrics
```

```
y_pred1 = rfe.predict(X_test)
metrics.r2_score(y_test, y_pred1)
```

---

```
0.7756349800595066
```

# MODEL BUILDING USING MLR

- The score provided by multiple regression model is 78%
- This model fits a little better than RFE

```
lm = LinearRegression()  
lm.fit(X_train, y_train)  
#Rsquared on test set  
y_pred1 = lm.predict(X_test)  
metrics.r2_score(y_test, y_pred1)
```

0.7811962854570624

# MODEL BUILDING USING RANDOM FOREST REGRESSOR

- The score provided by random forest regressor is almost 72% on both train and test set.
- This model fits a little less than RFE and MLR

```
y_train_pred = rf.predict(X_train)
print(r2_score(y_train, y_train_pred))
```

0.7245706459113583

```
y_test_pred = rf.predict(X_test)
print(r2_score(y_test, y_test_pred))
```

0.7273636647483164

# MODEL BUILDING USING GRADIENT BOOSTING REGRESSOR

- The score provided by the gradient boosting regressor is almost 90% on both train and test set.
- This model fits far better than RFE and MLR

```
y_train_pred = gbr.predict(X_train)
print("Train Set R_squared : ",r2_score(y_train, y_train_pred))
```

```
Train Set R_squared :  0.9040959918040924
```

```
y_test_pred = gbr.predict(X_test)
print("Test Set R_squared: ",r2_score(y_test, y_test_pred))
```

```
Test Set R_squared:  0.9009826242325177
```

# MODEL BUILDING USING XGBOOST REGRESSOR

- The score provided by the XGBoost regressor is almost 99% on both train and test set.
- This model fits way better than Gradient Boosting Regressor, MLR and RFE

```
XGB = m3.fit(X_train,y_train)
pred = XGB.predict(X_train)
print("Train set score: ", r2_score(y_train, pred))
y_test_pred = XGB.predict(X_test)
print("Test set score: ", r2_score(y_test, y_test_pred))
```

```
Train set score:  0.9961894342887798
Test set score:  0.9901404217164748
```

# MODEL BUILDING USING ADABOOST REGRESSOR

- The score provided by the adaboost is almost 77% on both train and test set.
- This model fits similar to RFE

```
from sklearn.ensemble import AdaBoostRegressor  
  
ar = AdaBoostRegressor(n_estimators=500, learning_rate=0.1)  
ab=ar.fit(X_train,y_train)  
pred=ab.predict(X_train)  
print("Train set score: ", r2_score(y_train, pred))  
y_test_pred = ab.predict(X_test)  
print("Test set score: ", r2_score(y_test, y_test_pred))
```

```
Train set score:  0.7675151817171677  
Test set score:  0.7722737116807638
```

# RECOMMENDATIONS

- XGBoost provided a great train and test score of 99%. So, this model should be used.
- The focus should be on car and personal loan types.
- Various factors such as tenure, collateral value and monthly EMI should be considered while estimating the LGD



**THANK YOU!**