دانشگاه اصفهان
دانشکده مهندسی کامپیوتر

مستند پروژه اول یادگیری ماشین

درس مبانی یادگیری ماشین

استاد درس: دکتر کیانی

امیرعلی لطفی (۴۰۰۳۶۱۳۰۵۳)

بهار ۱۴۰۳

مقدمه

در این مستند به گزارش تجزیه و تحلیل‌های انجام شده بر روی پروژه اول یادگیری ماشین پرداخته می‌شود و راه‌های مختلفی که آزمایش شدند با یکدیگر مقایسه می‌شوند.

بخش اول: آماده‌سازی داده

در این بخش به ۴ عمل مهم می‌توان اشاره کرد:

۱) تشخیص ستون‌های فاقد ارزش

ستون "CLIENTNUM" برای هر مشتری یکتا می‌باشد و در واقع شناسه آن مشتری در بانک است. این ستون هیچ ارزشی را به داده‌های مسئله اضافه نمی‌کند.
ستون "Unnamed 19" نیز برای همه داده‌ها بدون مقدار است.
این ۲ ستون را از دیتاست حذف می‌کنیم:

```
df = df.drop(columns=["CLIENTNUM", "Unnamed: 19"])
```

۲) حذف داده‌های پرت

با استفاده از فرمول زیر می‌توان داده‌های پرت را از هر ستون تشخیص داد و آن‌ها را حذف نمود:

```
IQR = Q3 - Q1
Lower Bound = Q1−1.5 * IQR
Upper Bound = Q3 + 1.5 * IQR
```

حال هر داده‌ای که از Lower Bound کوچک‌تر و یا از Upper Bound بزرگ‌تر باشد را حذف می‌کنیم.

۲) مدیریت مقادیر N/A

یکپارچه سازی مقادیر NULL

با جستجو در دیتاست، مشاهده می‌شود که در بعضی ستون‌ها مقادیر N/A به صورت رشته‌ای آمده‌اند.
این مقادیر را به منظور مدیریت بهتر و یکپارچه سازی داده‌ها به مقدار np.nan تبدیل می‌کنیم:

```
df = df.replace('Unknown', np.nan)
```

برخورد با مقادیر N/A

با مقادیر N/A می‌توان به ۲ صورت برخورد کرد:

۱- حذف کل آن سطر که شامل حداقل یک داده N/A می‌باشد:
در این روش حدود ۴۴ درصد داده‌ها حذف خواهند شد که عدد بسیار بزرگی است. پس این روش باعث از دست رفتن اطلاعات زیادی می‌شود. در نتیجه در این پروژه استفاده نشده است.

۲- پر کردن آن داده بر اساس داده‌های موجود:

۲.۱- استفاده از نتایج آماری:
برای مثال می‌توان مقادیر N/A هر ستون را با مد آن ستون جایگزین کرد.
عملکرد مدل در حالتی که مقادیر به صورت زیر پر شوند، به شرح زیر خواهد بود:

```
df = df.fillna({
   "Gender": df["Gender"].mode()[0],
   "Education_Level": df["Education_Level"].mode()[0],
   "Marital_Status": df["Marital_Status"].mode()[0],
   "Income_Category": df["Income_Category"].mode()[0],
   "Card_Category": df["Card_Category"].mode()[0],
   "Months_on_book": df["Months_on_book"].mean(),
   "Total_Relationship_Count": df["Total_Relationship_Count"].mean(),
})
```

```
Mean Squared Error: 11257063.139577858
R-squared: 0.8612194648911345
```

۲.۲- مدل‌های یادگیری ماشین:

در این روش می‌توان با استفاده از مدل‌های KNN یا KMeans مقادیر ناموجود را بر اساس دیگر داده‌ها حدس زد.
در روش KNN، عملکرد مدل به شرح زیر می‌باشد:

```
Mean Squared Error: 10274570.39934987
R-squared: 0.870661325954436
```

۳) حذف داده‌های تکراری

با حذف داده‌های تکراری از overfit شدن مدل جلوگیری می‌شود، اما ممکن است که اطلاعاتی را از دست بدهیم و به اصطلاح data loss داشته باشیم.

```
df.duplicated().sum()
```

Output: 35

مشاهده می‌شود که ۳۵ داده تکراری وجود دارد. آن‌ها از دیتاست حذف می‌کنیم:

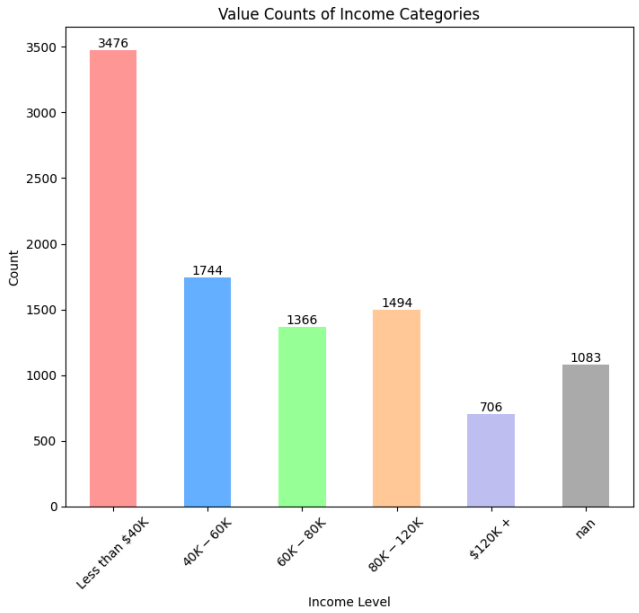
```
df = df.drop_duplicates()
```

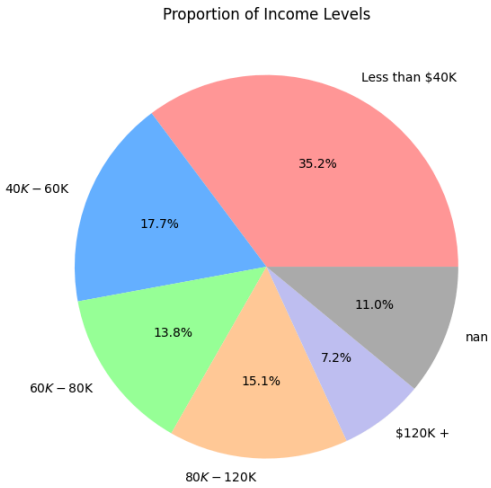
توزیع و نسبت داده‌ها

برای پیدا کردن دید نسبت به داده‌ها چند نمودار قرار داده شده است:

نسبت سطوح درآمدی

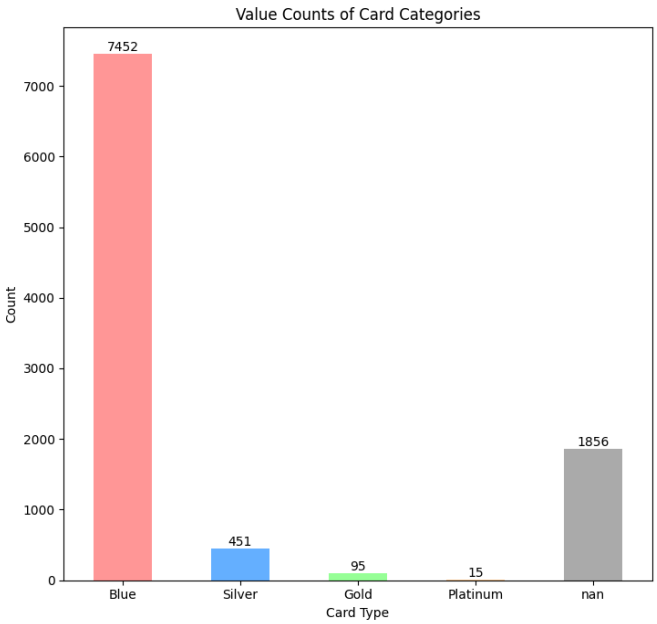
تقریبا با رشد سطح درآمد، نسبت کاهش پیدا می‌کند.

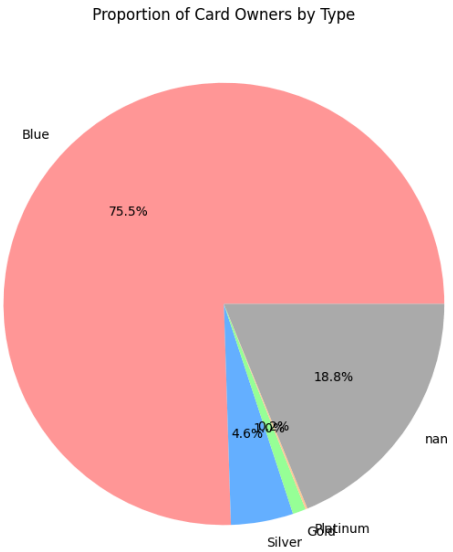




نسبت نوع کارت‌ها

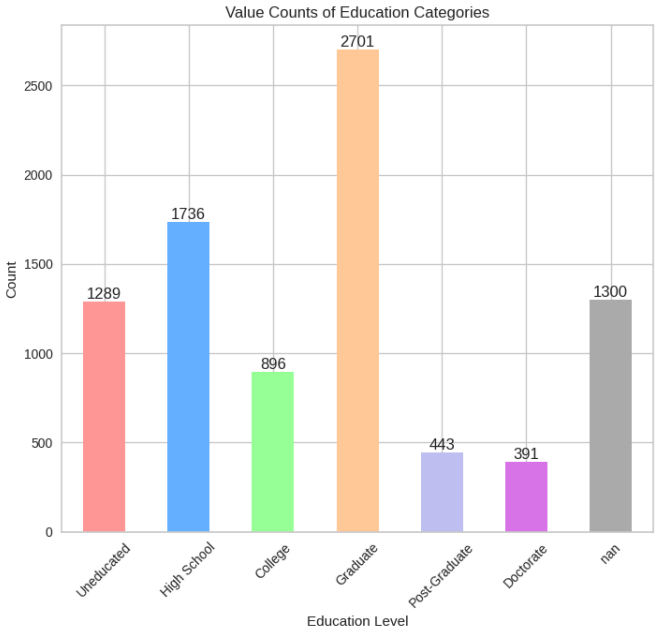
تقریبا حدود ۵ درصد از کارت‌هایی به غیر از Blue استفاده می‌کنند به طوری حدود ۷۵ درصد مشتریان دارای کارت Blue هستند و از ۱۸.۸ درصد داده‌ای در دسترس نیست.

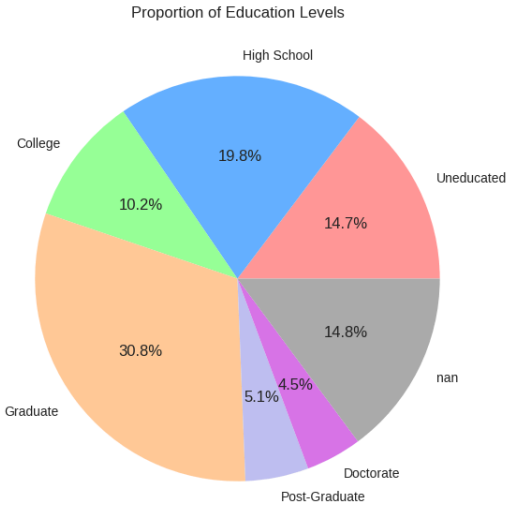




نسبت سطوح تحصیلات

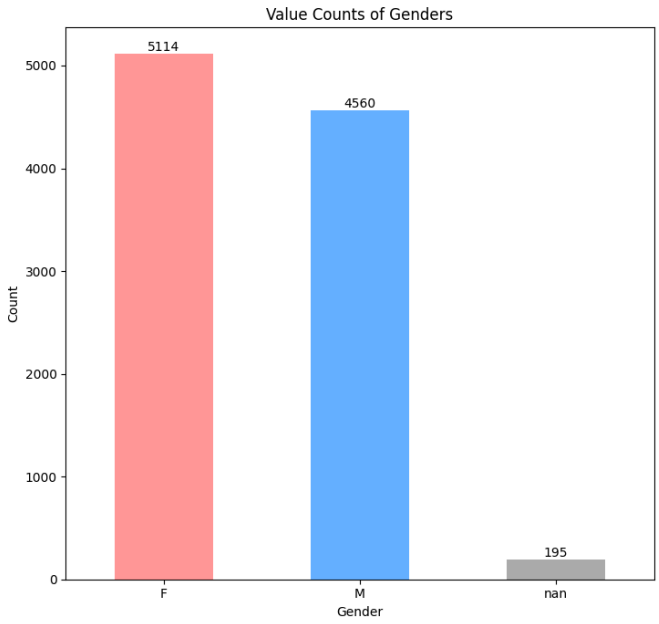
در نمودارهای زیر مشخص است که بیشترین درصد متعلق به افراد Graduate می‌باشد. همچنین داده‌های nan نیز حدود ۱۵ درصد داده‌های را تشکیل می‌دهند که به نسبت، درصد زیادی است.

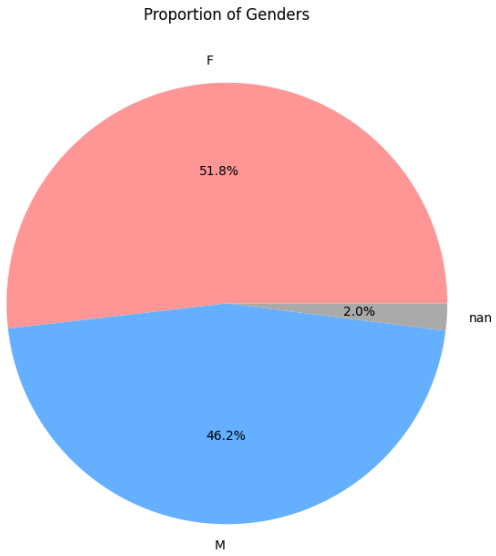




نسبت جنسیت

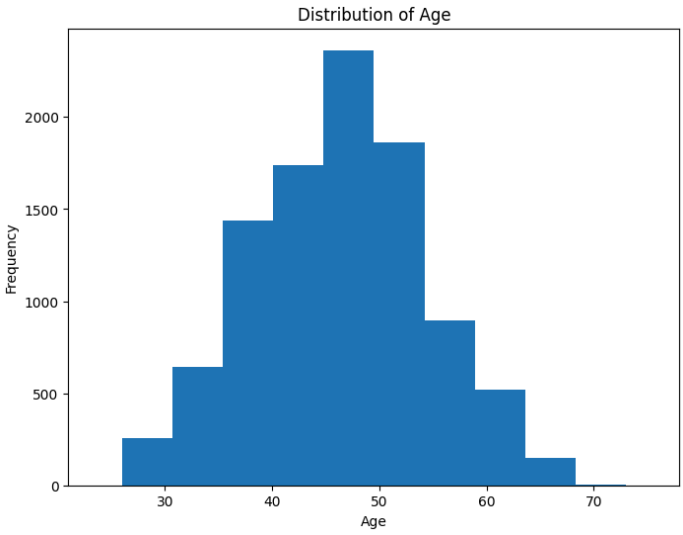
نسبت جنسیت زن و مرد بسیار نزدیک به هم می‌باشد.

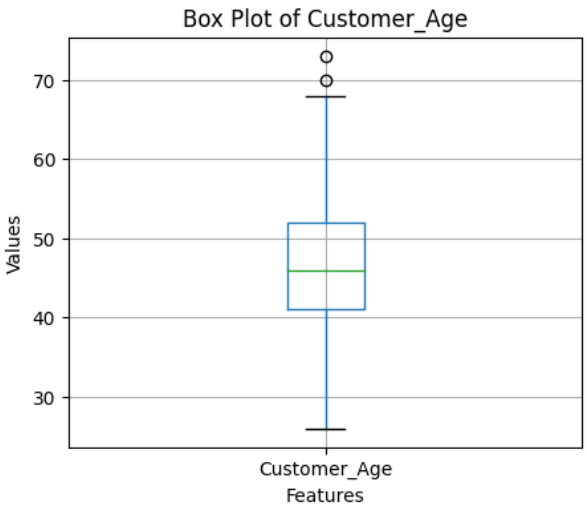




توزیع سن

طبق این نمودارها می‌توان دریافت کرد که سن افراد از یک توزیع نرمال پیروی می‌کند.

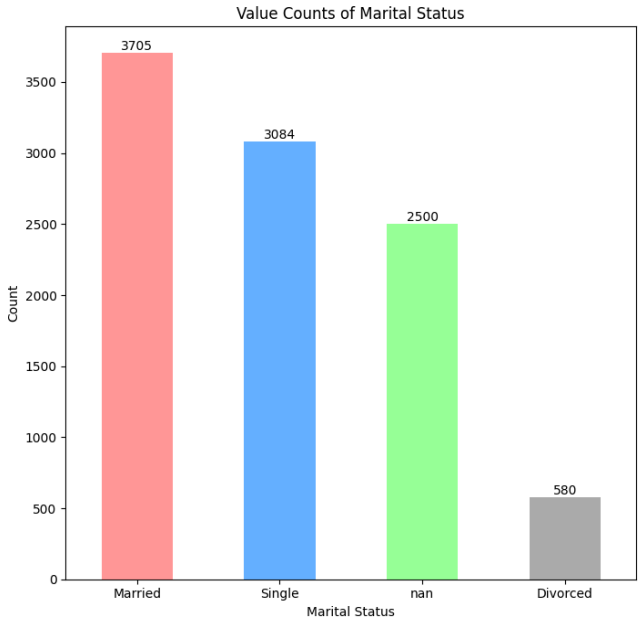


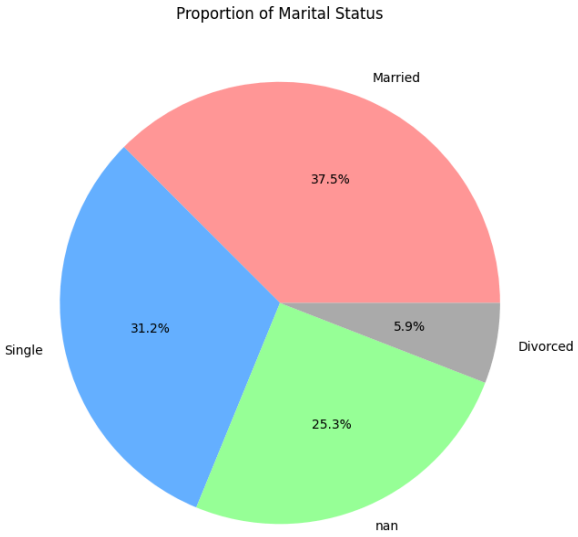


نسبت وضعیت تاهل

بیشترین درصد متعلق به افراد متاهل، و سپس با اختلاف کمی افراد مجرد قرار دارند. داده وضعیت تاهل ۲۵ درصد مشتریان در دسترس نیست. حدود ۶ درصد مشتریان نیز طلاق گرفته‌اند.

از آنجایی که نسبت افرادی که وضعیت تاهلشان مشخص نیست، بزرگ است، این را یک دسته جداگانه در نظر می‌گیریم.





مقایسه مدل‌ها

با کمک کتابخانه pycaret به مقایسه عملکرد مدل‌های مختلف پرداخته شد. نتایج به صورت زیر است:

Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
Random Forest Regressor	1334.9235	11778751.9325	3422.8344	0.8586	0.4235	0.2907	5.2470
Light Gradient Boosting Machine	1435.3349	11835470.0128	3426.0900	0.8582	0.4273	0.3030	1.9280
Gradient Boosting Regressor	1758.9642	12166495.3747	3482.6628	0.8539	0.4428	0.3503	1.5080

Extra Trees Regressor	1359.8433	12187763.5359	3480.2951	0.8536	0.4274	0.2966	3.1170
Extreme Gradient Boosting	1581.1646	13202859.6000	3623.8348	0.8417	0.4770	0.3312	0.3610
Decision Tree Regressor	1736.9758	22207422.4650	4687.6932	0.7339	0.5423	0.3412	0.0950
AdaBoost Regressor	3143.6964	22496025.7708	4735.6103	0.7296	0.6523	0.7792	0.1560
Lasso Least Angle Regression	4185.7802	33224433.1190	5762.6871	0.6007	0.8646	0.8834	0.0530
Ridge Regression	4184.2513	33228498.5363	5763.0387	0.6007	0.8695	0.8824	0.0330
Lasso Regression	4185.7801	33224432.0498	5762.6871	0.6007	0.8646	0.8834	0.0370
Linear Regression	4186.9032	33228883.2399	5763.0788	0.6007	0.8618	0.8839	0.6660
Least Angle Regression	4207.4679	33243953.2488	5764.4788	0.6004	0.8667	0.8968	0.0580
Elastic Net	5269.4364	52997302.8355	7275.7807	0.3639	0.8176	1.0704	0.0600
Huber Regressor	5103.8892	59806184.7916	7727.4020	0.2826	0.8057	0.8137	0.1700
Bayesian Ridge	6753.1257	80541007.4343	8970.4188	0.0333	1.0197	1.4920	0.0610
Orthogonal Matching Pursuit	6752.7758	80611662.1821	8974.3644	0.0325	1.0221	1.5012	0.0550
Dummy Regressor	6891.0459	83374674.4000	9127.8686	-0.0010	1.0421	1.5606	0.0280
K Neighbors Regressor	6764.3771	88212301.6000	9387.0979	-0.0591	1.0215	1.3874	0.1180

Passive Aggressive Regressor	11657.5442	201668887.2806	13978.5666	-1.4381	1.4516	3.0140	

با توجه به جدول بالا، مدل Random Forset Regressor نتایج بهتری نسبت به دیگر مدل‌ها داشته است.

تلاش برای بهبود مدل

برای بهبود دادن مدل، سعی بر استفاده از خوشه‌بندی به عنوان preprocess شد.
ابتدا با استفاده از الگوریتم K-Means، به ۳ خوشه تقسیم می‌شوند. (تعداد خوشه‌ها با آزمون و خطا بهینه شده است). سپس خوشه‌بندی روی داده‌های train انجام می‌شود و برای هر خوشه یک مدل RandomForestRegressor آموزش داده می‌شود.
مراکز به دست آمده از خوشه‌ها:

```
array([[-2.60581868e-02,  1.27002378e-03,  4.29382528e-02,
        -5.30971112e-02,  1.48325073e-02, -3.44072954e-02,
         2.95260176e-03,  1.42960234e-02,  7.01058972e-03,
         6.61656958e-02,  4.26513247e-02,  3.76544453e-03,
        -2.89291340e-02,  1.68923284e-02,  1.33758207e-02,
        -7.66243393e-01, -6.75376233e-01,  1.70734775e+00,
        -8.56826541e-03],
       [-3.38231701e-02, -3.41138034e-02, -4.93338133e-02,
        -2.02813439e-02, -4.38450223e-02,  2.53843167e-02,
         4.53628634e-03, -7.04375350e-02, -1.65136565e-02,
         1.52231262e-01,  2.05088403e-01,  2.49085585e-02,
        -3.71064337e-02, -2.71540730e-02,  5.86371897e-02,
        -7.66243393e-01,  1.25144052e+00, -5.85703763e-01,
        -1.43865710e-02],
       [ 4.69760793e-02,  2.95207662e-02,  1.60499918e-02,
         5.24570403e-02,  2.93855704e-02, -2.76755846e-04,
        -5.94883606e-03,  5.33871603e-02,  1.01445948e-02,
        -1.78296442e-01, -2.10070742e-01, -2.45966998e-02,
         5.17574331e-02,  1.32039397e-02, -6.08274072e-02,
         1.17823803e+00, -6.75376233e-01, -5.85703763e-01,
```

```
       1.83506010e-02]])
```

سپس در فاز predict، برای هر داده ابتدا نزدیک‌ترین خوشه به دست می‌آید، سپس با استفاده از مدل آن خوشه، مقدار Credit Limit حدس زده می‌شود.
عملکرد این مدل به شرح زیر است:

```
Mean Squared Error (MSE): 11972238.599267434
R-squared (R2) Score: 0.8492906850991695
```

اما متاسفانه این روش از حالتی که تنها یک مدل RandomForestRegressor آموزش داده شود، به طور ناچیز عملکرد بدتری داشت.

مقایسه مدل‌های نهایی

Model	MSE	R2
Linear Regression	حدود ۳۳ میلیون	حدود ۰.۶۶
Random Forest	حدود ۱۰ میلیون	حدود ۰.۸۷
KMeans + Random Forest	حدود ۱۱ میلیون	حدود ۰.۸۴