APPLIED DATASCIENCE PHASE - 3

CREDITCARD FRAUD DETECTION

PROBLEM STATEMENT:

The problem is to develop a machine learning-based system for real-time credit card fraud detection. The goal is to create a solution that can accurately identify fraudulent transactions while minimizing false positives. This project involves

- data preprocessing
- feature engineering
- model selection
- training
- evaluation to create a robust fraud detection system.

DATASET EXPLANATION:

The dataset that is used with this proposed approach is a real-world dataset obtained from Kaggle . It contains transactions made by credit cards

Dataset Link: **https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud**

The raw dataset taken for the study was sorted and pre-processed for the sole intention of improving the performance of the classifiers and reducing their training and operating time.

Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	-1.3598	-0.0728	2.53635	1.37816	-0.3383	0.46239	0.2396	0.0987	0.36379	0.09079	-0.5516	-0.6178	-0.9914	-0.3112	1.46818	-0.4704	0.20797	0.02579	0.40399	0.25141	-0.0183	0.27784	-0.1105	0.06693	0.12854	-0.1891	0.13356	-0.0211	149.62	0
0	1.19186	0.26615	0.16648	0.44815	0.06002	-0.0824	-0.0788	0.0851	-0.2554	-0.167	1.61273	1.06524	0.4891	-0.1438	0.63556	0.46392	-0.1148	-0.1834	-0.1458	-0.0691	-0.2258	-0.6387	0.10129	-0.3398	0.16717	0.12589	-0.009	0.01472	2.69	0
1	-1.3584	-1.3402	1.77321	0.37978	-0.5032	1.8005	0.79146	0.24768	-1.5147	0.20764	0.6245	0.06608	0.71729	-0.1659	2.34586	-2.8901	1.10997	-0.1214	-2.2619	0.52498	0.248	0.77168	0.90941	-0.6893	-0.3276	-0.1391	-0.0554	-0.0598	378.66	0
1	-0.9663	-0.1852	1.79299	-0.8633	-0.0103	1.2472	0.23761	0.37744	-1.387	-0.055	-0.2265	0.17823	0.50776	-0.2879	-0.6314	-1.0596	-0.6841	1.96578	-1.2326	-0.208	-0.1083	0.00527	-0.1903	-1.1756	0.64738	-0.2219	0.06272	0.06146	123.5	0
2	-1.1582	0.87774	1.54872	0.40303	-0.4072	0.09592	0.59294	-0.2705	0.81774	0.75307	-0.8228	0.5382	1.34585	-1.1197	0.17512	-0.4514	-0.237	-0.0382	0.80349	0.40854	-0.0094	0.79828	-0.1375	0.14127	-0.206	0.50229	0.21942	0.21515	69.99	0
2	-0.426	0.96052	1.14111	-0.1683	0.42099	-0.0297	0.4762	0.26031	-0.5687	-0.3714	1.34126	0.35989	-0.3581	-0.1371	0.51762	0.40173	-0.0581	0.06865	-0.0332	0.08497	-0.2083	-0.5598	-0.0264	-0.3714	-0.2328	0.10591	0.25384	0.08108	3.67	0
4	1.22966	0.141	0.04537	1.20261	0.19188	0.27271	-0.0052	0.08121	0.46496	-0.0993	-1.4169	-0.1538	-0.7511	0.16737	0.05014	-0.4436	0.00282	-0.612	-0.0456	-0.2196	-0.1677	-0.2707	-0.1541	-0.7801	0.75014	-0.2572	0.03451	0.00517	4.99	0
7	-0.6443	1.41796	1.07438	-0.4922	0.94893	0.42812	1.12063	-3.8079	0.61537	1.24938	-0.6195	0.29147	1.75796	-1.3239	0.68613	-0.0761	-1.2221	-0.3582	0.3245	-0.1567	1.94347	-1.0155	0.0575	-0.6497	-0.4153	-0.0516	-1.2069	-1.0853	40.8	0
7	-0.8943	0.28616	-0.1132	-0.2715	2.6696	3.72182	0.37015	0.85108	-0.392	-0.4104	-0.7051	-0.1105	-0.2863	0.07436	-0.3288	-0.2101	-0.4998	0.11876	0.57033	0.05274	-0.0734	-0.2681	-0.2042	1.01159	0.3732	-0.3842	0.01175	0.1424	93.2	0
9	-0.3383	1.11959	1.04437	-0.2222	0.49936	-0.2468	0.65158	0.06954	-0.7367	-0.3668	1.01761	0.83639	1.00684	-0.4435	0.15022	0.73945	-0.541	0.47668	0.45177	0.20371	-0.2469	-0.6338	-0.1208	-0.385	-0.0697	0.0942	0.24622	0.08308	3.68	0
10	1.44904	-1.1763	0.91386	-1.3757	-1.9714	-0.6292	-1.4232	0.04846	-1.7204	1.62666	1.19964	-0.6714	-0.5139	-0.095	0.23093	0.03197	0.25341	0.85434	-0.2214	-0.3872	-0.0093	0.31389	0.02774	0.50051	0.25137	-0.1295	0.04285	0.01625	7.8	0
10	0.38498	0.61611	-0.8743	-0.094	2.92458	3.31703	0.47045	0.53825	-0.5589	0.30976	-0.2591	-0.3261	-0.09	0.36283	0.9289	-0.1295	-0.81	0.35999	0.70766	0.12599	0.04992	0.23842	0.00913	0.99671	-0.7673	-0.4922	0.04247	-0.0543	9.99	0
10	1.25	-1.2216	0.38393	-1.2349	-1.4854	-0.7532	-0.6894	-0.2275	-2.094	1.32373	0.22767	-0.2427	1.20542	-0.3176	0.72567	-0.8156	0.87394	-0.8478	-0.6832	-0.1028	-0.2318	-0.4833	0.08467	0.39283	0.16113	-0.355	0.02642	0.04242	121.5	0
11	1.06937	0.28772	0.82861	2.71252	-0.1784	0.33754	-0.0967	0.11598	-0.2211	0.46023	-0.7737	0.32339	-0.0111	-0.1785	-0.6556	-0.1999	0.12401	-0.9805	-0.9829	-0.1532	-0.0369	0.07441	-0.0714	0.10474	0.54826	0.10409	0.02149	0.02129	27.5	0
12	-2.7919	-0.3278	1.64175	1.76747	-0.1366	0.8076	-0.4229	-1.9071	0.75571	1.15109	0.84456	0.79294	0.37045	-0.735	0.4068	-0.3031	-0.1559	0.77827	2.22187	-1.5821	1.15166	0.22218	1.02059	0.02832	-0.2327	-0.2356	-0.1648	-0.0302	58.8	0
12	-0.7524	0.34549	2.05732	-1.4686	-1.1584	-0.0778	-0.6086	0.0036	-0.4362	0.74773	-0.794	-0.7704	1.04763	-1.0666	1.10695	1.66011	-0.2793	-0.42	0.43254	0.26345	0.49962	1.35365	-0.2566	-0.0651	-0.0391	-0.0871	-0.181	0.12939	15.99	0
12	1.10322	-0.0403	1.26733	1.28909	-0.736	0.28807	-0.5861	0.18938	0.78233	-0.268	-0.4503	0.93671	0.70838	-0.4686	0.35457	-0.2466	-0.0092	-0.5959	-0.5757	-0.1139	-0.0246	0.196	0.0138	0.10376	0.3643	-0.3823	0.09281	0.03705	12.99	0
13	-0.4369	0.91897	0.92459	-0.7272	0.91568	-0.1279	0.70764	0.08796	-0.6653	-0.738	0.3241	0.27719	0.25262	-0.2919	-0.1845	1.14317	-0.9287	0.68047	0.02544	-0.047	-0.1948	-0.6726	-0.1569	-0.8884	-0.3424	-0.049	0.07969	0.13102	0.89	0
14	-5.4013	-5.4501	1.1863	1.73624	3.04911	-1.7634	-1.5597	0.16084	1.23309	0.34517	0.91723	0.97012	-0.2666	-0.4791	-0.5266	0.472	-0.7255	0.07508	-0.4069	-2.1968	-0.5036	0.98446	2.45859	0.04212	-0.4816	-0.6213	0.39205	0.94959	46.8	0
15	1.49294	-1.0293	0.45479	-1.438	-1.5554	-0.721	-1.0807	-0.0531	-1.9787	1.63808	1.07754	-0.632	-0.417	0.05201	-0.043	-0.1664	0.30424	0.55443	0.05423	-0.3879	-0.1776	-0.1751	0.04	0.29581	0.33293	-0.2204	0.0223	0.0076	5	0
16	0.69488	-1.3618	1.02922	0.83416	-1.1912	1.30911	-0.8786	0.44529	-0.4462	0.56852	1.01915	1.29833	0.42048	-0.3727	-0.808	-2.0446	0.51566	0.62585	-1.3004	-0.1383	-0.2956	-0.572	-0.0509	-0.3042	0.072	-0.4222	0.08655	0.0635	231.71	0
17	0.9625	0.32846	-0.1715	2.1092	1.12957	1.69604	0.10771	0.5215	-1.1913	0.7244	1.69033	0.40677	-0.9364	0.98374	0.71091	-0.6022	0.40248	-1.7372	-2.0276	-0.2693	0.144	0.40249	-0.0485	-1.3719	0.39081	0.19996	0.01637	-0.0146	34.09	0
18	1.16662	0.50212	-0.0673	2.26157	0.4288	0.08947	0.24115	0.13808	-0.9892	0.92217	0.74479	-0.5314	-2.1053	1.12687	0.00308	0.42442	-0.4545	-0.0989	-0.8166	-0.3072	0.0187	-0.062	-0.1039	-0.3704	0.6032	0.10856	-0.0405	-0.0114	2.28	0
18	0.24749	0.27767	1.18547	-0.0926	-1.3144	-0.1501	-0.9464	-1.6179	1.54407	-0.8299	-0.5832	0.52493	-0.4534	0.08139	1.5552	-1.3969	0.78313	0.43662	2.17781	-0.231	1.65018	0.20045	-0.1854	0.42307	0.82059	-0.2276	0.33663	0.25048	22.75	0
22	-1.9465	-0.0449	-0.4056	-1.0131	2.94197	2.95505	-0.0631	0.85555	0.04997	0.57374	-0.0813	-0.2157	0.04416	0.0339	1.19072	0.57884	-0.9757	0.04406	0.4886	-0.2167	-0.5795	-0.7992	0.8703	0.98342	0.3212	0.14965	0.70752	0.0146	0.89	0
22	-2.0743	-0.1215	1.32202	0.41001	0.2952	-0.9595	0.54399	-0.1046	0.47566	0.14945	-0.8566	-0.1805	-0.6552	-0.2798	-0.2117	-0.3333	0.01075	-0.4885	0.50575	-0.3867	-0.4036	-0.2274	0.74243	0.39853	0.24921	0.2744	0.35997	0.24323	26.43	0
23	1.17328	0.3535	0.28391	1.13356	-0.1726	-0.9161	0.36902	-0.3273	-0.2467	-0.0461	-0.1434	0.97935	1.49229	0.10142	0.76148	-0.0146	-0.5116	-0.3251	-0.3909	0.02788	0.067	0.22781	-0.1505	0.43505	0.72482	-0.3371	0.01637	0.03004	41.88	0
23	1.32271	-0.174	0.43456	0.57604	-0.8368	-0.8311	-0.2649	-0.221	-1.0714	0.86856	-0.6415	-0.1113	0.36149	0.17195	0.78217	-1.3559	-0.2169	1.27177	-1.2406	-0.523	-0.2844	-0.3234	-0.0377	0.34715	0.55964	-0.2802	0.04234	0.02882	16	0
23	-0.4143	0.90544	1.72745	1.47347	0.00744	-0.2003	0.74023	-0.0292	-0.5934	-0.3462	-0.0121	0.7868	0.63595	-0.0863	0.0768	-1.4059	0.77559	-0.9429	0.54397	0.09731	0.07724	0.45733	-0.0385	0.64252	-0.1839	-0.2775	0.18269	0.15266	33	0
23	1.05939	-0.1753	1.26613	1.18611	-0.786	0.57844	-0.7671	0.40105	0.6995	-0.0647	1.04829	1.00562	-0.542	-0.0399	-0.2187	0.00448	-0.1936	0.04239	-0.2778	-0.178	0.01368	0.21373	0.01446	0.00295	0.29464	-0.3951	0.08146	0.02422	12.99	0
24	1.23743	0.06104	0.38053	0.76156	-0.3598	-0.4941	0.00649	-0.1339	0.43881	-0.2074	-0.9292	0.52711	0.34868	-0.1525	-0.2184	-0.1916	-0.1166	-0.6338	0.34842	-0.0664	-0.2457	-0.5309	-0.0443	0.07917	0.50914	0.28886	-0.0227	0.01184	17.28	0
25	1.11401	0.08555	0.4937	1.33576	-0.3002	-0.0108	-0.1188	0.18862	0.20569	0.08226	1.13356	0.6267	-1.4928	0.52079	-0.6746	-0.5291	0.15826	-0.3988	-0.1457	-0.2738	-0.0532	-0.0048	-0.0315	0.19805	0.56501	-0.3377	0.02906	0.00445	4.45	0
26	-0.5299	0.87389	1.34725	0.14546	0.41421	0.10022	0.71121	0.17607	-0.2867	-0.4847	0.87249	0.85164	-0.5717	0.10097	-1.5198	-0.2844	-0.3105	-0.4042	-0.8234	-0.2903	0.04695	0.2081	-0.1855	0.00103	0.09882	-0.5529	-0.0733	0.02331	6.14	0
26	-0.5299	0.87389	1.34725	0.14546	0.41421	0.10022	0.71121	0.17607	-0.2867	-0.4847	0.87249	0.85164	-0.5717	0.10097	-1.5198	-0.2844	-0.3105	-0.4042	-0.8234	-0.2903	0.04695	0.2081	-0.1855	0.00103	0.09882	-0.5529	-0.0733	0.02331	6.14	0
26	-0.5354	0.86527	1.35108	0.14758	0.43368	0.08698	0.69304	0.17974	-0.2856	-0.4825	0.8718	0.85345	-0.5718	0.10225	-1.52	-0.2859	-0.3096	-0.4039	-0.8237	-0.2833	0.04953	0.20654	-0.1871	0.00075	0.09812	-0.5535	-0.0783	0.02543	1.77	0
26	-0.5354	0.86527	1.35108	0.14758	0.43368	0.08698	0.69304	0.17974	-0.2856	-0.4825	0.8718	0.85345	-0.5718	0.10225	-1.52	-0.2859	-0.3096	-0.4039	-0.8237	-0.2833	0.04953	0.20654	-0.1871	0.00075	0.09812	-0.5535	-0.0783	0.02543	1.77	0
27	-0.246	0.47327	1.69574	0.26241	-0.0109	-0.6108	0.79394	-0.2473	0.13888	-0.401	-0.8121	-0.1835	-0.6301	-0.2862	-0.337	-0.4281	-0.0306	-0.5046	0.35518	0.04572	-0.1946	-0.3351	-0.0782	0.39278	-0.0312	0.19804	-0.1753	-0.2036	30.49	0
27	-1.4522	1.76512	0.61167	1.17682	-0.446	0.24683	-0.2576	1.09247	-0.6075	0.04716	0.78373	1.09639	-0.2681	0.76865	-0.5244	-0.8088	0.71039	-0.1184	0.97929	0.00871	0.08228	0.32578	-0.0691	0.02096	-0.0447	-0.2434	0.14918	0.12056	1.8	0
29	0.99637	-0.1226	0.54682	0.70658	0.13456	1.157	-0.2946	0.40743	0.33786	-0.4082	0.68265	1.25372	0.39192	-0.0375	0.7953	-1.6682	1.32299	-2.83	-1.4634	-0.2035	-0.0763	0.10887	0.16223	-0.5756	0.10979	0.37381	0.05055	0.00511	20.53	0
29	1.11088	0.16872	0.51714	1.32541	-0.1916	0.0195	-0.0318	0.11762	0.01766	0.04486	1.34507	1.28634	-0.2523	0.27446	-0.8104	-0.587	0.08745	-0.5505	-0.1547	-0.1901	-0.0377	0.0957	-0.0482	0.23211	0.6062	-0.3421	0.03677	0.00748	6.54	0
32	1.24905	-0.6247	-0.7106	-0.9916	1.42997	3.69298	-1.0902	0.96729	0.85015	-0.3071	-0.4562	0.22998	-0.0169	-0.2208	0.36242	0.31522	-0.5123	0.11899	0.57472	0.09785	-0.0063	0.0092	-0.1295	1.11297	0.50038	1.19655	-0.0482	0.00509	29.89	0
32	-2.0089	2.19853	0.14424	1.15943	-0.8152	0.18229	-0.6171	1.53082	-0.5868	0.12988	0.65848	1.26505	-0.2065	1.0136	-0.5494	-0.708	0.86964	-0.0957	0.9444	0.02874	0.09492	0.29498	0.01108	0.01525	0.03421	-0.2361	0.12829	0.11799	2.35	0
33	-0.6079	1.03135	1.74045	1.23211	0.41859	0.11917	0.85089	-0.1763	-0.2435	0.14846	-0.387	0.3983	0.48192	-0.3654	0.23554	-1.3478	0.50465	-0.7984	0.75971	0.25432	-0.0873	0.25832	-0.2648	0.11828	0.17351	-0.217	0.09431	-0.033	14.8	0
33	-0.9357	0.17042	2.74626	-1.078	-0.3056	0.01158	-0.2962	0.40278	-0.0405	-0.852	1.55211	0.95356	0.41543	-0.2339	0.45234	0.84437	-0.8227	0.33882	-1.2142	-0.0054	0.40121	1.06486	-0.1583	0.29551	-0.2594	0.75419	0.04666	0.09395	9.1	0
34	-0.7623	0.12777	2.65006	-1.8757	-0.8934	-0.2687	-0.4199	0.14623	-0.8696	-0.0387	-0.6861	-0.5029	1.38083	-1.0391	0.65284	1.89875	-0.4584	-0.5598	-0.2951	0.35939	0.43335	1.2604	-0.4047	0.09495	0.48098	-0.0907	0.31922	0.1356	15.99	0
34	1.13832	0.05696	0.64942	0.87306	-0.4685	-0.4102	-0.0139	-0.0724	0.30679	-0.27	-0.0026	1.1243	0.74421	-0.1884	-0.0757	-0.7092	0.38122	-1.3721	-0.2737	-0.0784	-0.1642	-0.2474	0.05941	0.45629	0.361	0.27441	-0.0025	0.01711	21.34	0
34	-0.2915	0.44558	1.24975	-1.7357	0.08576	-0.1219	0.40772	0.09531	0.8159	-1.4912	-0.8462	0.05653	-0.059	0.15192	1.98259	-0.4433	-0.3183	0.06479	0.61351	-0.0335	-0.0649	-0.1204	-0.1565	-0.8002	-0.0006	-0.8352	0.131	0.0629	18.95	0
34	0.20169	0.49748	1.37391	0.57141	-0.6307	-0.5397	-0.0763	-0.9171	0.27001	-0.4803	-0.5125	0.68091	0.08992	-0.1574	-0.5888	-0.4779	0.22539	-0.6687	-0.0236	-0.2223	0.71896	-0.1719	-0.1662	0.77647	0.81846	0.44255	0.14276	0.21924	9.99	0
35	1.3864	-0.7942	0.77822	-0.8647	-1.0641	0.3513	-1.1915	0.05269	-0.3044	0.57652	-1.6311	0.04256	2.0479	-0.7393	1.45622	-0.272	-0.932	1.92653	-0.6599	-0.273	-0.2287	-0.1235	-0.131	-0.9297	0.18138	1.19493	0.00053	0.01991	30.9	0
35	-1.0632	1.41819	1.08667	1.24144	0.00231	0.0459	0.51412	0.24125	-0.1545	0.28067	-0.5319	0.58986	0.55509	-0.1164	0.16589	-1.2372	0.63841	-0.7761	0.69281	0.23501	-0.0572	0.31415	-0.1299	0.11428	-0.0278	-0.2599	0.11875	-0.009	20.22	0
35	1.19936	0.12995	0.86358	1.00263	-0.7838	-0.8847	-0.0407	-0.2081	0.39248	-0.2482	-0.0954	1.31413	1.09168	-0.4093	-0.578	-0.5813	0.21234	-0.8932	-0.0207	-0.0726	-0.0425	0.19847	-0.033	1.01329	0.5591	0.40182	-0.0059	0.01794	0.99	0
36	-1.0049	-0.986	-0.038	3.71006	-6.632	5.1221	4.37169	-2.0069	-0.2787	-0.2309	0.14516	-0.0632	-0.7996	-0.342	-0.9305	0.51051	0.09243	0.82398	1.1904	-0.002	1.39341	-0.3817	0.96972	0.01944	0.57092	0.33328	0.85737	-0.0755	1402.95	0

LOADING THE DATASET:

It is the process of copying and loading data or data sets from a source file, folder or application to a database or similar application.

It is usually implemented by copying digital data from a source and pasting or loading the data to a data storage or processing utility.

Importing necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pylab import rcParams
import warnings
warnings.filterwarnings('ignore')
```

reading dataset:

```
data=pd.read_csv('creditcard.csv')
```

PREPROCESSING STEPS:

Data Cleaning:

Identify and handle missing values, outliers and inconsistences in the dataset to ensure the data integrity and accuracy.

Handling missing data

```
data.isnull().sum()
```

Output

```
Time     0
V1     0
V2     0
V3     0
V4     0
V5     0
V6     0
V7     0
V8     0
V9     0
V10     0
V11     1
V12     1
V13     1
V14     1
V15     1
V16     1
V17     1
V18     1
V19     1
V20     1
V21     1
V22     1
V23     1
V24     1
V25     1
V26     1
V27     1
V28     1
Amount    1
Class     1
dtype: int64
```

Spliting the data

```
fraud_cases=len(data[data['Class']==1])
print(' Number of Fraud Cases:',fraud_cases)
non_fraud_cases=len(data[data['Class']==0])
print('Number of Non Fraud Cases:',non_fraud_cases)
```

output

```
Number of Fraud Cases: 150
```

```
Number of Non Fraud Cases: 51440
```

Test set and Training set

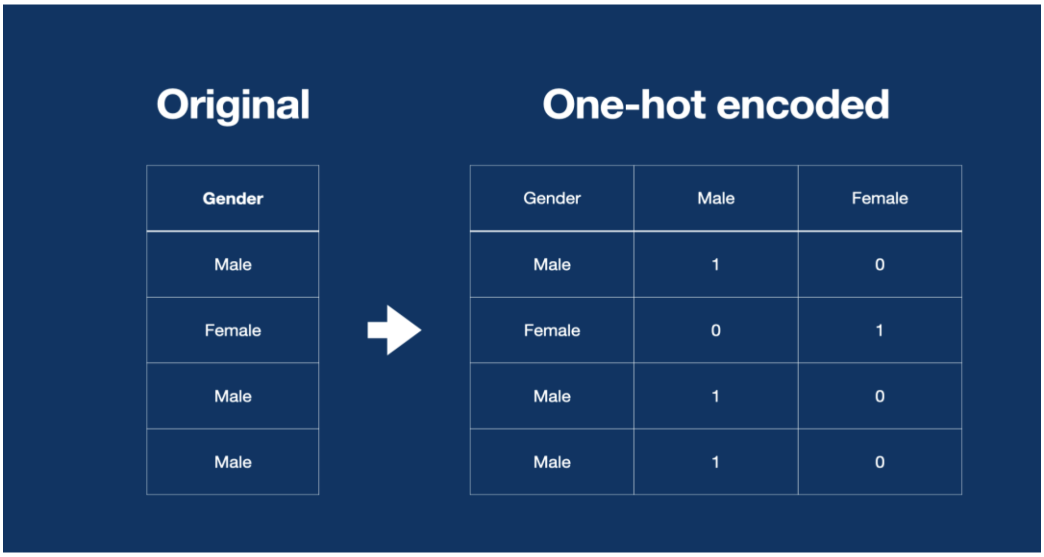
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
X = data.drop('Class', axis=1)
y = data.Class
X = scalar.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=1)
```

Determine which feature are relevant to project objectives and remove unnecessary variables.

ONE HOT ENCODING:

sklearn comes with a one-hot encoding tool built-in one hand encoder class. The onehot encoder class takes an array of data and can be used to one-hot encode the data.

```
from sklearn.preprocessing import OneHotEncoder
#creating instance of one-hot-encoder
encoder = OneHotEncoder(handle_unknown='ignore')
```



Data transformation:

Normalize or scale the data as needed to bring to consistent and comparable format. It includes

- Transforming units
- Aggregration
- Spatial scales

Information about valid transactions

```
print('Amount details of valid transaction')
valid_info= data[(data['Class']==0)]
valid_info.Amount.describe()
```

output

```
Amount details of valid transaction
count    51440.000000
mean        94.000267
std        253.580381
min          0.000000
25%          7.680000
50%         25.390000
75%         86.002500
max      12910.930000
Name: Amount, dtype: float64
```

information about fraud transactions

```
print('Amount details of fraud transaction')
fraud_info = data[data['Class'] ==1]
fraud_info.Amount.describe()
```

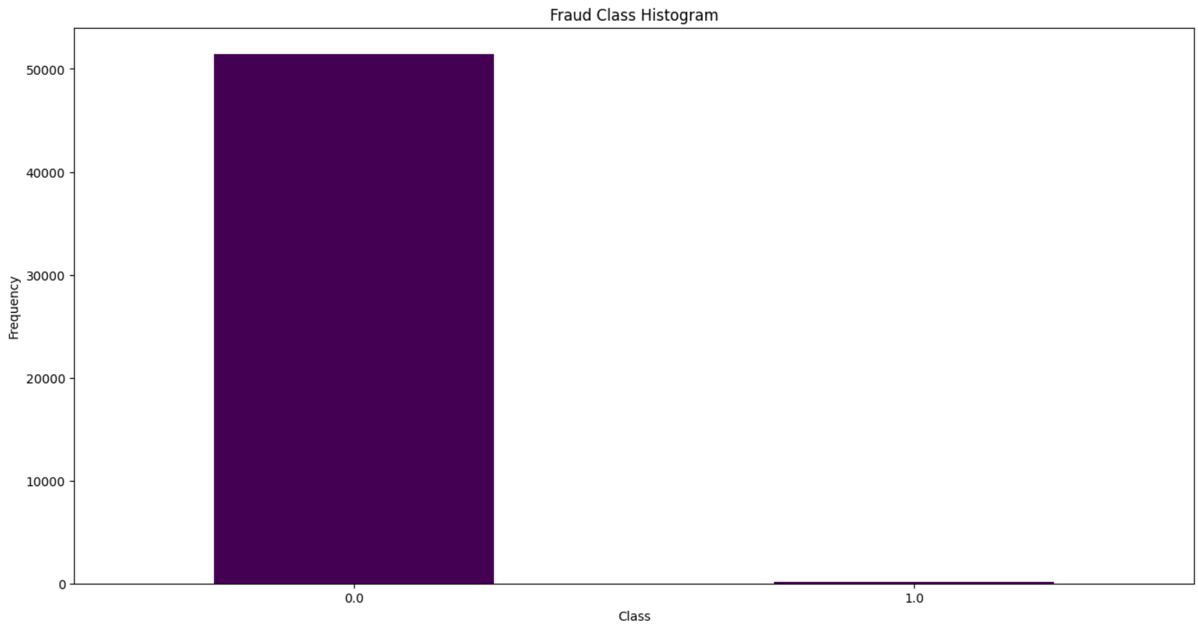
output

```
Amount details of fraud transaction
count     150.000000
mean       98.848400
std       232.056904
min         0.000000
25%         1.000000
50%         8.370000
75%        99.990000
max      1809.680000
Name: Amount, dtype: float64
```

Representation in Barplot

```
count_classes = pd.value_counts(data['Class'], sort = True
).sort_index()
count_classes.plot(kind = 'bar' ,rot = 0 ,colormap ='viridis')
```

```
plt.title ( "Fraud Class Histogram" )
plt.xlabel( "Class" )
plt.ylabel( "Frequency" )
```



Data Validation:

Check the data for consistency and accuracy, ensuring that it aligns with the project’s obejectives and it ready for subsequent analysis and modeling.