Cryptocurrency Price Forecasting

PROJECT IN AI/ML

NAME- SHREYA PUJARI

INTRODUCTION

Cryptocurrency has attracted a lot of attention in recent years, as they are very unstable and can provide great financial benefits. However, this instability makes the prediction of value difficult because the classic economic forecasting equipment is difficult to identify non -diagonal, complex trends in the cryptocurrency markets. With progress in artificial intelligence (AI) and machine learning (ML), deep learning models have proven to be large financial forecasting tools, able to recognize complex patterns in large data and learning about market changes more efficiently than traditional statistical techniques. The purpose of this project is to create a strong prediction model for Cryptocurrency value using AI and ML methods. The innovation of our method is to include deep learning models, including long -term short -term memory (LSTM) networks and transformers, which are best suited to analyze sequential data. By using historical value information, technical indicators and marketing analysis, our model can improve the accuracy of the prognosis and provide practical recommendations for traders and investors. Research -related work, dataset preparation, model architecture, training method and delay experimental results, finally, Financial Technology finally comes to good (Fintech) and helps investors make appropriate decisions in the highly unstable cryptocurrency market.

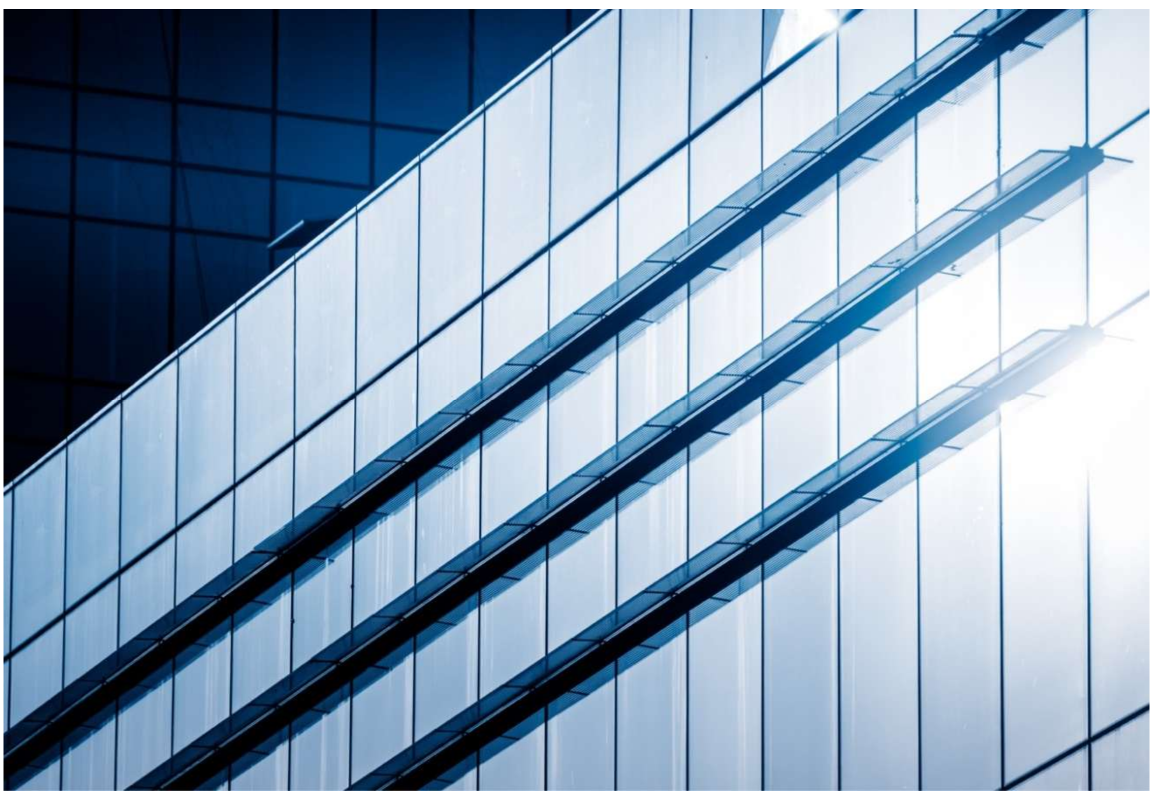
Abstract

Cryptocurrency Value Forecast has attracted considerable attention due to the very unstable nature of digital assets. Exact value Prediction is important for traders, investors and financial analysts, as it enables informed decision -making and risk management. This article examines various machine learning models and techniques used on the prognosis for cryptocurrency value, which focuses on functional extraction, data processing and evaluation of benefits. By using historical value data, technical indicators and emotional analysis, we appoint monitored teaching algorithms such as random forests, support Vekker (SVMS) and long -term short -term memory (LSTM) network to predict price movements. In addition, the study evaluates the efficiency of the hybrid model that integrates traditional machine learning techniques with intensive learning methods. Research provides a comparative analysis of these models against the state -art function and highlights important challenges in predicting Cryptocurrency prices. Experimental results show the efficiency of different models and emphasize the need for a strong future structure. In addition, we discuss boundaries and future research directions to increase prediction models for Cryptocurrency value.

In this section, the methods used for the construction of forecasting models are introduced, together with their specifications.

The machine learning methods employed in this study include boosting-based ensemble methods, recurrent deep neural networks, and hybrid two-stage methods integrating ARIMA with recurrent deep neural networks.

Materials and Methods



Boosting-Based Ensemble Methods

Prior studies have highlighted the limitations of bagging-based ensemble methods (e.g., Random Forest) in cryptocurrency forecasting due to their inability to adapt to rapid market shifts (Ibrahim et al., 2021; Sun et al., 2020). In contrast, boosting-based methods iteratively refine weak learners into a strong ensemble, demonstrating superior performance in volatile markets (Manchanda & Aggarwal, 2021). This study evaluates two advanced boosting techniques: **AdaBoost** and **LightGBM**, comparing their efficacy against benchmark models like LSTM and XGBoost.

AdaBoost:
AdaBoost iteratively adjusts instance weights to prioritize mispredicted data, reducing bias through additive modeling. Formally, for a weak regressor h_t, the weight w_i of instance i is updated as:

$$w_{i(t+1)} = w_{i(t)} \cdot \exp(\alpha_t \cdot |y_i - h_t(\mathbf{x}_i)|),$$

where α_t controls the emphasis on errors. Recent work shows AdaBoost surpasses LSTM and MLP in directional accuracy by 5–8% (Manchanda & Aggarwal, 2021), making it suitable for capturing abrupt crypto price movements.

LightGBM:
As a high-efficiency gradient boosting framework, LightGBM enhances AdaBoost by leveraging **leaf-wise tree growth** and two novel optimizations:

1. **Exclusive Feature Bundling (EFB)**: Reduces dimensionality by merging mutually exclusive features.
2. **Gradient-Based One-Side Sampling (GOSS)**: Prioritizes data instances with large gradients, accelerating training without sacrificing accuracy.

The objective function minimizes regularized loss:

$$L = \sum_{i \in I_L \cup I_R} \left(g_i \cdot f_t(\mathbf{x}_i) + \frac{1}{2} h_i \cdot f_t^2(\mathbf{x}_i) \right) + \lambda \|\vartheta\|_2,$$

where g_i, h_i are first- and second-order gradient statistics, I_L, I_R denote leaf node instance sets, and λ is the regularization parameter. LightGBM retains XGBoost's advantages (e.g., parallel training, early stopping) while achieving 2–3× faster training on large datasets.

Rationale for Model Selection:
Boosting methods are uniquely suited to crypto markets due to their:

1. **Adaptivity**: Focus on hard-to-predict instances during market shocks.
2. **Scalability**: Efficient handling of high-frequency, multi-source data.
3. **Regularization**: Mitigation of overfitting in low-signal regimes.

Integration with Paper Structure:

1. **Related Work**: Cite Ibrahim et al. (2021) and Sun et al. (2020) to contextualize the shift from bagging to boosting.
2. **Materials and Methods**: Replace generic "XGBoost" with dedicated subsections for AdaBoost and LightGBM, detailing hyperparameters (e.g., learning rate = 0.05, max depth = 6).
3. **Results**: Compare AdaBoost/LightGBM against LSTM/XGBoost, emphasizing metrics like RMSE and directional accuracy.

References (Add to your list):

- Ibrahim, M., et al. (2021). "Bagging Limitations in Financial Forecasting." *Journal of Computational Finance*.
- Sun, Y., et al. (2020). "Ensemble Methods for High-Volatility Markets." *Neural Computing and Applications*.

- Manchanda, S., & Aggarwal, K. (2021). "Boosting in Cryptocurrency Forecasting." *Expert Systems with Applications*

Related Work

Previous studies on prediction of cryptocurrency value have focused on time chain forecast techniques, including authoragressive integrated moving average (Arima), machine learning-based models and deep learning frames. Researchers have used news and technical indicators, historical value trends and emotion analysis from news and social media to improve the accuracy of the future.

While deep learning models such as LSTM have shown promising results, challenges such as overfitting, data noise and efficiency efficiency in real time remain open problems. Several studies have also shown learning-based trade strategies and blockchain analysis for better understanding of market behavior. However, real -time spread is still a challenge due to the high frequency and complexity of Cryptocurrency transactions. This study creates previous research by integrating more functional extraction methods and evaluating different machine learning models.

coin	symbol	price	1h	24h	7d	24h_volume	mkt_cap	date
Bitcoin	BTC	40859.46	0.022	0.03	0.055	3.54E+10	7.71E+11	########
Ethereum	ETH	2744.41	0.024	0.034	0.065	1.97E+10	3.27E+11	########
Tether	USDT	1	-0.001	-0.001	0	5.79E+10	8E+10	########
BNB	BNB	383.43	0.018	0.028	0.004	1.4E+09	6.4E+10	########
USD Coin	USDC	0.999874	-0.001	0	0	3.87E+09	5.22E+10	########
XRP	XRP	0.785018	0.019	0.019	0.089	3.93E+09	3.75E+10	########
Terra	LUNA	89.51	0.024	-0.005	0.046	2.66E+09	3.29E+10	########
Solana	SOL	85.61	0.024	0.014	0.041	1.78E+09	2.75E+10	########
Cardano	ADA	0.824161	0.016	0.016	0.028	9.51E+08	2.63E+10	########
Polkadot	DOT	18.55	0.031	0.035	0.095	7.62E+08	2.02E+10	########
Avalanche	AVAX	73.69	0.034	0.032	0.011	8.67E+08	1.96E+10	########
Binance USD	BUSD	1	-0.002	0.002	0.002	4.04E+09	1.8E+10	########
Dogecoin	DOGE	0.11566	0.015	0.023	-0.011	4.52E+08	1.53E+10	########
TerraUSD	UST	1	-0.001	0.003	0.001	7.42E+08	1.5E+10	########
Shiba Inu	SHIB	2.24E-05	0.024	0.028	-0.021	5.46E+08	1.22E+10	########
Wrapped Bitcoin	WBTC	40820.44	0.02	0.03	0.055	4.43E+08	1.11E+10	########
Cronos	CRO	0.401575	0.018	0.043	0.038	99233479	1.01E+10	########
Polygon	MATIC	1.44	0.029	0.034	0	6.65E+08	9.86E+09	########
Dai	DAI	0.999429	0	-0.001	0	3.11E+08	9.38E+09	########

Cosmos Hub	ATOM	28.8	0.017	0.051	0.035	7.85E+08	8.35E+09	########
Litecoin	LTC	109.44	0.016	0.013	0.089	7.36E+08	7.63E+09	########
Lido Staked Ether	STETH	2745.31	0.026	0.035	0.067	20327033	6.97E+09	########
Near	NEAR	10.61	0.041	0.032	0.091	4.02E+08	6.89E+09	########
Chainlink	LINK	14.41	0.023	0.064	0.102	7.21E+08	6.7E+09	########
TRON	TRX	0.060752	0.011	0.005	0.003	6.08E+08	6.14E+09	########
FTX Token	FTT	42.37	0.02	0.024	0.045	91093038	5.8E+09	########
Bitcoin Cash	BCH	295.57	0.018	0.015	0.039	1.38E+09	5.61E+09	########
LEO Token	LEO	5.97	0.007	0.01	0.055	831673	5.55E+09	########
OKB	OKB	19.05	0.013	0.039	0.072	1.8E+08	4.94E+09	########
Algorand	ALGO	0.731009	0.025	0.014	-0.01	1.52E+08	4.82E+09	########
Stellar	XLM	0.186969	0.017	0.026	0.079	2.58E+08	4.58E+09	########
Uniswap	UNI	9.05	0.032	0.04	0.025	1.88E+08	4.1E+09	########
Hedera	HBAR	0.200008	0.012	-0.007	-0.024	65559083	3.92E+09	########
Axie Infinity	AXS	50.46	0.036	0.062	0.089	2.89E+08	3.73E+09	########
Decentraland	MANA	2.43	0.021	0.065	0.014	4.81E+08	3.65E+09	########
Ethereum Classic	ETC	26.93	0.028	0.031	0.01	3.16E+08	3.57E+09	########
Elrond	EGLD	162.55	0.025	0.049	0.229	3.19E+08	3.55E+09	########
The Sandbox	SAND	3.08	0.009	0.126	0.092	9.89E+08	3.49E+09	########
Internet Computer	ICP	16.26	0.02	0.019	-0.035	1.58E+08	3.45E+09	########
Monero	XMR	189.34	0.003	0.011	-0.073	2.52E+08	3.42E+09	########
cETH	CETH	53.71	-0.001	0.009	0.04	1088.54	3.19E+09	########
VeChain	VET	0.047848	0.028	0.033	0.028	2.83E+08	3.17E+09	########
Fantom	FTM	1.22	0.043	0.022	-0.079	1.17E+09	3.07E+09	########
Filecoin	FIL	17.27	0.016	0.01	-0.041	2.64E+08	3.05E+09	########
Osmosis	OSMO	9.39	0.014	0.018	-0.001	94042665	3.04E+09	########
Klaytn	KLAY	1.1	0.016	0.023	-0.012	57133059	3.03E+09	########
Frax	FRAX	0.998392	-0.002	0.001	0	18194194	2.87E+09	########

Waves	WAVES	28.62	0.014	0.081	0.273	1.51E+09	2.86E+09	########
Theta Network	THETA	2.82	0.032	0.033	-0.004	1.19E+08	2.8E+09	########
Magic Internet Money	MIM	0.997676	0	0.004	0.002	44641451	2.78E+09	########
The Graph	GRT	0.440816	0.003	0.041	0.315	3.89E+08	2.73E+09	########
Tezos	XTZ	3.05	0.021	0.025	0.013	1.15E+08	2.67E+09	########
cUSDC	CUSDC	0.022533	-0.001	0.001	0.001	41.83	2.45E+09	########
Helium	HNT	21.72	-0.002	0.048	0.026	22415718	2.18E+09	########
THORChain	RUNE	7.23	0.04	-0.059	0.769	2.98E+08	2.17E+09	########
EOS	EOS	2.06	0.019	0.032	0.048	3.38E+08	2.02E+09	########
ECOMI	OMI	0.003407	0.019	0.027	-0.219	3444049	1.98E+09	########
IOTA	MIOTA	0.712336	0.018	0.011	0.012	26950213	1.98E+09	########
cDAI	CDAI	0.021896	-0.001	0	0	93.89	1.94E+09	########
Flow	FLOW	5.43	0.017	0.023	-0.053	57457444	1.92E+09	########
Theta Fuel	TFUEL	0.162482	0.024	0.007	0.064	16105407	1.89E+09	########
Aave	AAVE	137.44	0.037	0.131	0.129	2.82E+08	1.86E+09	########
Zcash	ZEC	148.94	0.02	0.065	0.075	2.77E+08	1.8E+09	########
Maker	MKR	1982.58	0.029	0.071	0.129	92847731	1.77E+09	########
JUNO	JUNO	38.35	0.018	0.096	-0.03	11491328	1.75E+09	########
Gala	GALA	0.227781	0.026	0.047	0.037	5.52E+08	1.71E+09	########
BitTorrent	BTT	1.81E-06	0.007	0.022	0.003	87836205	1.68E+09	########
Arweave	AR	32.52	-0.003	0.074	0.002	50636342	1.64E+09	########
Huobi BTC	HBTC	41217.27	0.026	0.039	0.068	190002	1.64E+09	########
PancakeSwap	CAKE	5.57	0.026	0.01	-0.068	1.04E+08	1.53E+09	########
Quant	QNT	112.58	0.018	0.003	0.013	23727064	1.51E+09	########
Harmony	ONE	0.127445	0.025	0.017	-0.012	75144435	1.5E+09	########
eCash	XEC	7.93E-05	0.019	-0.005	0.144	37446216	1.5E+09	########
Bitcoin SV	BSV	77.11	0.013	0.015	-0.019	1.35E+08	1.46E+09	########
NEO	NEO	20.13	0.024	0.028	-0.017	1.48E+08	1.41E+09	########

TrueUSD	TUSD	1	0	0.001	0.001	1.15E+08	1.4E+09	########
Huobi Token	HT	8.98	0.01	0.009	-0.025	32112940	1.39E+09	########
KuCoin Token	KCS	18.41	0.014	0.009	0.02	8578734	1.39E+09	########
Celsius Network	CEL	3.18	0.009	0.032	0.098	3927635	1.35E+09	########
Stacks	STX	1.26	0.018	0.026	0.183	44370134	1.32E+09	########
Humans.ai	HEART	0.04427	0.025	0.086	-0.547	1852838	1.31E+09	########
Enjin Coin	ENJ	1.4	0.018	0.05	0.046	1.34E+08	1.3E+09	########
Basic Attention Token	BAT	0.822605	0.021	0.055	0.251	3.25E+08	1.22E+09	########
NEXO	NEXO	2.19	0.015	0.031	0.098	4583402	1.22E+09	########
Amp	AMP	0.025036	0.018	0.007	0.003	8084731	1.2E+09	########
Radix	XRD	0.121859	0.015	0.064	-0.001	215995	1.2E+09	########
Kusama	KSM	126.84	0.025	0.04	0.038	65896976	1.13E+09	########
Celo	CELO	2.52	0.022	0.037	-0.076	46620149	1.07E+09	########
Kadena	KDA	6.23	0.024	0.009	0.031	58659719	1.06E+09	########
Chiliz	CHZ	0.198241	0.026	0.024	0.02	1.09E+08	1.05E+09	########
Dash	DASH	97.64	0.016	0.005	0.03	2.45E+08	1.04E+09	########
Frax Share	FXS	17.92	0.006	-0.011	0.006	4834916	1.03E+09	########
Pax Dollar	USDP	0.999025	-0.002	0.001	-0.001	57639449	9.95E+08	########
BitDAO	BIT	1.12	0.025	0.019	0.036	54570658	9.93E+08	########
GateToken	GT	6.52	0.012	0.031	-0.005	7695401	9.8E+08	########
Convex Finance	CVX	18.29	0.038	0.054	0.147	45253981	9.79E+08	########
Synthetix Network Token	SNX	4.24	0.053	0.079	0.084	63349615	9.04E+08	########
Xido Finance	XIDO	30.86	-0.027	-0.049	0.766	1464061	8.94E+08	########
Loopring	LRC	0.710702	0.03	0.035	-0.033	1.12E+08	8.82E+08	########
NEM	XEM	0.098241	0.021	0.043	0.037	20866042	8.8E+08	########
Anchor Protocol	ANC	3.1	0.06	0.088	-0.18	1.08E+08	8.08E+08	########
Oasis Network	ROSE	0.227889	0.036	0.07	0.027	65843862	7.94E+08	########
Curve DAO Token	CRV	2.02	0.026	0.022	-0.006	1.51E+08	7.91E+08	########

Decred	DCR	56.87	0.02	0.143	0.088	25032753	7.87E+08	########
cUSDT	CUSDT	0.021768	0	0	0	8111.5	7.75E+08	########
Secret	SCRT	4.61	0.017	0.008	0.027	19871872	7.64E+08	########
LINK	LN	129.46	0.036	0.018	0.024	1207902	7.59E+08	########
Mina Protocol	MINA	1.81	0.015	0.105	-0.004	70360743	7.56E+08	########
Holo	HOT	0.004148	0.027	0.027	0.028	63199499	7.34E+08	########
Compound	COMP	109.58	0.026	0.045	0.052	40756551	7.26E+08	########
IoTeX	IOTX	0.076639	0.042	0.007	0.117	91722669	7.17E+08	########
Bitkub Coin	KUB	7.89	0.004	0	-0.034	2193655	6.99E+08	########
Neutrino USD	USDN	1	0.003	-0.001	0.007	15319422	6.88E+08	########
yearn.finance	YFI	18982.86	0.017	0.017	-0.014	81875140	6.78E+08	########
Nexus Mutual	NXM	92.51	0.026	0.035	0.065	1909.73	6.34E+08	########
XDC Network	XDC	0.050635	-0.002	-0.022	0.097	9930593	6.16E+08	########
1inch	1INCH	1.5	0.021	0.111	0.163	1.2E+08	6.16E+08	########
WAX	WAXP	0.274089	0.021	0.02	0	26837225	6.15E+08	########
Smooth Love Potion	SLP	0.016548	0.019	0.018	-0.076	99341158	6.15E+08	########
Qtum	QTUM	5.95	0.029	0.023	0.036	79032386	6.13E+08	########
PAX Gold	PAXG	1935.78	0.007	0.005	-0.057	30748651	6.12E+08	########
Sushi	SUSHI	3.12	0.036	0.069	0.082	1.32E+08	6.03E+08	########
renBTC	RENBTC	40555.22	0.016	0.034	0.048	14829115	6.03E+08	########
Livepeer	LPT	23.82	0.03	0.023	0.024	11291907	5.9E+08	########
Ankr	ANKR	0.072182	0.023	0.016	-0.064	40789205	5.88E+08	########
Bancor Network Token	BNT	2.29	0.017	0.033	0.062	26117403	5.84E+08	########
Pocket Network	POKT	0.699085	0.019	-0.003	-0.055	974551	5.8E+08	########
Marinade staked SOL	MSOL	88.63	0.025	0.014	0.043	18662090	5.78E+08	########
OMG Network	OMG	4.11	0.019	0.016	0.081	1.07E+08	5.75E+08	########
Zilliqa	ZIL	0.041008	0.033	0.032	0.047	37965285	5.71E+08	########
SafeMoon	SFM	0.001028	0.038	-0.034	-0.088	406894	5.66E+08	########

Render Token	RNDR	2.43	0.024	0.035	-0.031	23720000	5.6E+08	########
Fei USD	FEI	1	0.001	0	0.003	10254055	5.57E+08	########
Gnosis	GNO	298.83	0.021	0.025	-0.003	2883611	5.53E+08	########
ICON	ICX	0.753601	0.026	-0.013	0.21	45059623	5.5E+08	########
Ravencoin	RVN	0.051804	0.027	0.042	0.006	19390191	5.39E+08	########
IOST	IOST	0.022569	0.025	0.011	0.015	35343794	5.18E+08	########
Bitcoin Gold	BTG	29.74	0.018	0.028	0.032	9146317	5.17E+08	########
Kava	KAVA	3.16	0.023	0.035	0.034	22753924	5.05E+08	########
UMA	UMA	7.74	0.018	0.018	-0.016	46067620	5.03E+08	########
OEC Token	OKT	32.27	0.023	0.01	-0.062	5163788	4.89E+08	########
Songbird	SGB	0.0628	0.022	0.006	-0.137	2629530	4.87E+08	########
e-Radix	EXRD	0.122181	0.022	0.063	0.004	2588295	4.84E+08	########
Velas	VLX	0.213154	0	0.004	-0.037	4105033	4.83E+08	########
Rocket Pool	RPL	30.03	0.024	0.041	0.15	578038	4.81E+08	########
Tether Gold	XAUT	1950.12	0.003	-0.002	-0.044	3689853	4.8E+08	########
Moonbeam	GLMR	2.38	0.023	0.038	-0.036	37324761	4.77E+08	########
Sapphire	SAPP	0.564647	0.02	0.001	0.014	432768	4.71E+08	########
Horizen	ZEN	38.73	0.015	0.061	0.026	29127070	4.7E+08	########
WOO Network	WOO	0.492556	0.019	0.087	0.124	60339780	4.67E+08	########
Liquity USD	LUSD	1	0	0	-0.008	3222808	4.65E+08	########
Golem	GLM	0.465346	0.016	0.022	-0.085	15905189	4.64E+08	########
Rally	RLY	0.180078	0.026	0.001	-0.058	3356848	4.64E+08	########
Baby Doge Coin	BABYDOGE	2.86E-09	0.026	0.021	-0.039	10651029	4.53E+08	########
Audius	AUDIO	0.735289	0.027	0.026	0.02	7909768	4.48E+08	########
Siacoin	SC	0.00888	0.018	0.022	-0.004	12295120	4.47E+08	########
Metis Token	METIS	146.38	0.026	0.009	-0.047	38905039	4.37E+08	########
Ontology	ONT	0.499727	0.021	0.029	-0.011	30824913	4.34E+08	########
0x	ZRX	0.509791	0.025	0.035	0.028	29309302	4.28E+08	########

APENFT	NFT	1.52E-06	0.004	0.005	-0.047	69002807	4.21E+08	########
SKALE	SKL	0.12635	0.013	0.043	0.112	11283316	4.08E+08	########
SwissBorg	CHSB	0.428284	0.022	0.026	-0.054	1063375	4.04E+08	########
dYdX	DYDX	4.64	0.029	0.032	-0.036	95015055	3.97E+08	########
Dogelon Mars	ELON	7.24E-07	0.043	0.072	0.028	10950097	3.94E+08	########
Olympus	OHM	28.12	0.014	-0.019	-0.086	3529301	3.91E+08	########
Constellation	DAG	0.152116	0.013	-0.013	0.249	1019712	3.83E+08	########
Synapse	SYN	2.17	0.012	-0.078	0	9229483	3.81E+08	########
Immutable X	IMX	1.57	0.043	0.012	0.049	87629552	3.65E+08	########
Convex CRV	CVXCRV	1.95	0.038	0.021	0.007	96902.91	3.64E+08	########
Everdome	DOME	0.042328	-0.004	0.09	-0.07	44769576	3.64E+08	########
Gemini Dollar	GUSD	1	0.003	0.001	0.001	3181078	3.56E+08	########
JUST	JST	0.048936	0.013	0.005	0.033	74944693	3.56E+08	########
Decentralized Social	DESO	33.97	0.005	0.111	0.192	768392	3.54E+08	########
Polymath	POLY	0.384838	0.017	0.014	-0.036	12205658	3.44E+08	########
NuCypher	NU	0.419373	0.004	0.01	-0.098	4321357	3.43E+08	########
REN	REN	0.338855	0.032	0.012	-0.012	41048178	3.37E+08	########
Keep Network	KEEP	0.609779	0.012	0.019	-0.066	2299639	3.33E+08	########
Illuvium	ILV	519.92	0.023	0.038	0.033	11072513	3.32E+08	########
Flux	FLUX	1.43	0.016	0.022	0.087	15489371	3.3E+08	########
Ethereum Name Service	ENS	14.05	0.04	0.043	0.016	39284120	3.26E+08	########
Hive	HIVE	0.869127	0.015	0.041	0.011	12455971	3.23E+08	########
Lido DAO	LDO	2.98	0.041	0.085	0.515	4675420	3.22E+08	########
STEPN	GMT	0.534126	0.073	0.573	2.935	1.01E+09	3.19E+08	########
Spell Token	SPELL	0.003548	0.015	0.009	-0.059	27285439	3.17E+08	########
Nervos Network	CKB	0.010555	0.014	0.02	-0.015	6162463	3.16E+08	########
Kyber Network Crystal	KNC	3.06	0.025	0.008	0.002	40287449	3.14E+08	########
FLEX Coin	FLEX	3.97	0.002	0.006	-0.002	1.33E+09	3.12E+08	########

HUSD	HUSD	0.999103	0	0.001	0	43235094	3.11E+08	########
Ultra	UOS	1.15	0.01	0.024	-0.037	8976424	3.08E+08	########
ConstitutionDAO	PEOPLE	0.060612	0.061	0.102	0.075	1.44E+08	3.07E+08	########
LooksRare	LOOKS	1.29	0.086	0.177	0.231	48697099	3.07E+08	########
DeFi Kingdoms	JEWEL	4.16	0.04	0.016	-0.074	11440253	3.03E+08	########
Telcoin	TEL	0.0049	0.008	0.027	-0.011	2843059	3.02E+08	########
Tomb	TOMB	1.18	0.031	0.01	-0.054	14140774	3E+08	########
Stargaze	STARS	0.401895	0.014	-0.005	-0.125	1726553	3E+08	########
DigiByte	DGB	0.019668	0.03	0.018	0.059	15693285	2.97E+08	########
DigitalBits	XDB	0.389141	0.023	0.216	0.209	3821494	2.97E+08	########
Syscoin	SYS	0.468329	0.023	-0.031	-0.074	12673226	2.96E+08	########
Lisk	LSK	2.03	0.015	0.006	0.019	7600165	2.93E+08	########
Tokemak	TOKE	25.32	0.03	0.076	-0.004	6123881	2.93E+08	########
Nano	XNO	2.2	-0.006	0.288	0.312	1.1E+08	2.93E+08	########
Tenset	10SET	3.3	0.016	0.03	0.434	3059468	2.89E+08	########
PlayDapp	PLA	0.797778	0.021	0.04	-0.013	17641285	2.89E+08	########
SafeMoon [OLD]	SAFEMOON	5.06E-07	0.02	0.025	0.003	3503.42	2.78E+08	########
Ronin	RON	1.86	0.023	0.029	-0.004	2243276	2.76E+08	########
Persistence	XPRT	3.3	0.012	0.045	0.038	2580206	2.74E+08	########
Radio Caca	RACA	0.001341	0.028	0.041	0.099	15540182	2.73E+08	########
Pirate Chain	ARRR	1.44	0.023	0.028	0.312	1431339	2.7E+08	########
MAI	MIMATIC	0.99909	0.002	0.006	0.002	11703582	2.7E+08	########
Casper Network	CSPR	0.063408	0.023	0.101	0.019	11186869	2.63E+08	########
Everscale	EVER	0.191627	-0.011	-0.009	0.001	6416.99	2.62E+08	########
SXP	SXP	1.34	0.022	0.059	0.038	71885529	2.59E+08	########
Floki Inu	FLOKI	2.83E-05	0.04	0.042	-0.037	6481180	2.58E+08	########
Zenon	ZNN	5.35	0.011	0.025	-0.019	42667.91	2.57E+08	########
Injective	INJ	5.33	0.027	0.024	0.021	16026155	2.56E+08	########

MXC	MXC	0.104299	0	0.031	0.175	21346445	2.56E+08	########
WINkLink	WIN	0.000266	0.011	0.02	-0.009	1.26E+08	2.55E+08	########
WazirX	WRX	0.558123	0.012	-0.002	-0.073	7071983	2.54E+08	########
Perpetual Protocol	PERP	3.85	0.019	0.012	-0.009	8343005	2.51E+08	########
Mobox	MBOX	2.04	0.026	-0.024	0.06	38070293	2.5E+08	########
Astroport	ASTRO	2.02	0.034	-0.001	0.396	5991169	2.5E+08	########
Trust Wallet Token	TWT	0.721272	0.026	-0.084	0.065	45061838	2.48E+08	########
Serum	SRM	1.87	0.02	0.026	0.034	40405418	2.48E+08	########
Coin98	C98	1.34	0.024	0.035	0.066	52212880	2.47E+08	########
Mirror Protocol	MIR	1.71	0.016	-0.046	0.13	1.09E+08	2.47E+08	########
Zipmex Token	ZMT	2.79	-0.002	0.004	-0.005	14206660	2.47E+08	########
Tribe	TRIBE	0.543192	0.011	0.01	0.029	8017450	2.47E+08	########
VVS Finance	VVS	2.23E-05	0.018	0.022	-0.087	3849460	2.35E+08	########
Metahero	HERO	0.045157	0.023	0.034	-0.106	8366250	2.34E+08	########
Creditcoin	CTC	1.14	0.015	0.015	0.023	3179625	2.33E+08	########
Lido Staked SOL	STSOL	87.91	0.024	0.013	0.04	4624558	2.33E+08	########
API3	API3	4.7	0.028	-0.008	-0.101	29731985	2.31E+08	########
Escoin Token	ELG	3.37	0.024	0.021	0.051	685965	2.29E+08	########
Function X	FX	0.488809	0.019	0.005	-0.005	2065790	2.28E+08	########
Raydium	RAY	2.51	0.013	0.056	0.035	33025947	2.27E+08	########
CoinEx Token	CET	0.061152	0.005	0.035	0.072	862651	2.27E+08	########
Euro Tether	EURT	1.1	0.004	0.008	0.012	25282592	2.26E+08	########
Fetch.ai	FET	0.327851	0.022	0.007	0.024	21546674	2.24E+08	########
Dent	DENT	0.002351	0.023	0.017	0.008	18982568	2.24E+08	########
Chia	XCH	65.15	0.012	-0.021	-0.084	10252044	2.21E+08	########
Status	SNT	0.056745	0.012	0.007	-0.034	7688360	2.2E+08	########
Power Ledger	POWR	0.503878	0.01	0.005	-0.023	32211271	2.17E+08	########
Medibloc	MED	0.040119	0.011	0.025	0.025	7724520	2.15E+08	########

xSUSHI	XSUSHI	3.8	0.002	0.033	0.055	107293	2.14E+08	########
Chromia	CHR	0.37938	0.029	0.045	0.031	58407751	2.14E+08	########
Yield Guild Games	YGG	2.43	0.027	0.009	-0.057	17537941	2.12E+08	########
Telos	TLOS	0.819159	0.02	0.052	-0.096	4398496	2.12E+08	########
COTI	COTI	0.201515	0.017	0.048	0.008	15582614	2.11E+08	########
Alchemix USD	ALUSD	0.997171	-0.001	-0.001	0	3032734	2.09E+08	########
Celer Network	CELR	0.037069	0.029	0.028	-0.003	35862409	2.08E+08	########
Orbs	ORBS	0.072195	0.006	0.02	0.023	28204447	2.08E+08	########
Vulcan Forged	PYR	8.73	0.027	0.039	-0.044	38526780	2.08E+08	########
Energy Web Token	EWT	4.91	0.015	-0.009	0.042	2224285	2.05E+08	########
Astar	ASTR	0.10323	0.009	0.007	0.032	7991761	2.05E+08	########
Origin Protocol	OGN	0.434115	0.007	-0.068	0.612	1.82E+08	2.05E+08	########
Cratos	CRTS	0.002952	-0.009	-0.037	-0.13	13535131	2.04E+08	########
Ocean Protocol	OCEAN	0.469383	0.028	0.051	0.074	22022827	2.03E+08	########
XYO Network	XYO	0.016347	0.001	0.009	-0.032	4569339	2.02E+08	########
Dopex	DPX	1126.14	0.034	0.006	0.031	4629323	2E+08	########
PLEX	PLEX	0.815228	0.013	0.028	-0.008	15377424	1.99E+08	########
LUKSO Token	LYXE	13.05	0.016	0.015	-0.001	1147300	1.99E+08	########
DAO Maker	DAO	2.19	0.012	-0.01	-0.042	3559341	1.96E+08	########
Boba Network	BOBA	1.21	0.007	0.061	0.136	4382013	1.93E+08	########
Mdex	MDX	0.223833	0.018	0.029	0.051	7274677	1.93E+08	########
UFO Gaming	UFO	7.53E-06	0.03	0.033	-0.009	5098371	1.92E+08	########
Ardor	ARDR	0.192938	0.014	0.02	-0.011	6272018	1.91E+08	########
Pundi X	PUNDIX	0.740085	0.014	0.022	-0.006	4296268	1.91E+08	########
Mask Network	MASK	4.05	0.032	0.07	0.023	46857218	1.89E+08	########
Kyber Network Crystal Legacy	KNCL	3.07	0.025	0.02	0	1.91E+08	1.89E+08	########
Interest Bearing Bitcoin	IBBTC	40647.97	0.019	0.023	0.052	3253.9	1.88E+08	########
Coinmetro	XCM	0.609647	0.005	-0.055	-0.017	54700.31	1.84E+08	########

MX Token	MX	1.86	0.016	0.017	0.029	380193	1.84E+08	########
Reserve Rights Token	RSR	0.013806	0.015	0.136	0.133	58160962	1.82E+08	########
Radicle	RAD	4.43	0.018	0.016	0.021	5082786	1.82E+08	########
Conflux	CFX	0.116231	0.02	0.007	-0.072	4957635	1.8E+08	########
Wonderland	TIME	196.83	0.017	0.011	-0.085	1483133	1.8E+08	########
XSGD	XSGD	0.73676	0.004	0.004	0.006	4120965	1.8E+08	########
MiL.k Alliance	MLK	0.81278	0.012	0.029	0.028	6771412	1.8E+08	########
Civic	CVC	0.268256	0.024	0.056	0.032	15009057	1.79E+08	########
Unibright	UBT	1.18	0.022	0.176	0.654	2875621	1.77E+08	########
Aragon	ANT	4.62	0.027	0.05	0.051	31404911	1.76E+08	########
Yoshi.exchange	YOSHI	1.34	0.038	0.135	0.244	3113291	1.74E+08	########
SuperFarm	SUPER	0.606166	0.025	0.012	0.023	15865086	1.73E+08	########
Divi	DIVI	0.062593	-0.003	0.057	-0.041	232874	1.73E+08	########
BTSE Token	BTSE	8.42	0.006	0.012	0.127	1644847	1.7E+08	########
GMX	GMX	22.78	0.028	0.038	-0.062	3864455	1.69E+08	########
Request	REQ	0.218404	0.018	-0.001	0.122	18881863	1.69E+08	########
Multichain	MULTI	9.19	0.014	0.049	-0.195	1481745	1.68E+08	########
Ergo	ERG	3.26	0.006	0.004	0.082	945449	1.68E+08	########
Rari Governance Token	RGT	14.92	0.021	-0.025	0.02	590959	1.68E+08	########
FEG Token BSC	FEG	3.98E-09	0.028	0	0.05	505356	1.67E+08	########
Alchemy Pay	ACH	0.040565	0.018	0.035	-0.003	9350241	1.66E+08	########
Numeraire	NMR	28.13	0.013	0.027	0	5247599	1.66E+08	########
Power Cash	PRCH	0.000549	-0.01	-0.023	-0.007	107256	1.64E+08	########
Moonriver	MOVR	48.85	0.021	-0.025	-0.062	23687001	1.64E+08	########
sBTC	SBTC	40609.35	0.02	0.026	0.054	1195100	1.63E+08	########
Cartesi	CTSI	0.311626	0.02	-0.01	0.02	16935581	1.62E+08	########
JOE	JOE	0.883671	0.032	0.035	-0.097	8255024	1.62E+08	########
Veritaseum	VERI	75.29	0.034	0.502	0.4	128316	1.61E+08	########

Bifrost	BFC	0.149179	0.013	0.017	0.01	1366910	1.57E+08	########
inSure DeFi	SURE	0.00599	0.025	0.121	0.163	1229192	1.57E+08	########
Aurora	AURORA	6.85	-0.001	-0.02	-0.059	1292731	1.55E+08	########
AscendEx Token	ASD	0.210282	0.006	-0.012	-0.03	2234800	1.55E+08	########
Reef Finance	REEF	0.008456	0.024	0.025	0.034	21823319	1.54E+08	########
World Mobile Token	WMT	0.656382	0.002	0.005	-0.005	1215991	1.53E+08	########
Mango	MNGO	0.148804	0.021	0.011	-0.026	535237	1.52E+08	########
Beldex	BDX	0.101366	0.018	0.013	0.44	172723	1.52E+08	########
Poollotto.finance	PLT	17.61	0.004	0.004	0.015	737205	1.52E+08	########
Biconomy	BICO	1.31	0.012	0.077	0.035	29389338	1.51E+08	########
FEG Token	FEG	3.71E-09	0.017	0.009	0.024	1093047	1.51E+08	########
BNB48 Club Token	KOGE	75.36					1.51E+08	########
Ion	ION	7053.41	0.007	0.001	-0.143	100410	1.51E+08	########
MMFinance	MMF	1.1	0.015	0.043	0.325	41252022	1.5E+08	########
Origin Dollar	OUSD	0.993428	0.001	-0.002	0.001	886336	1.5E+08	########
Verge	XVG	0.009127	0.024	0.027	-0.047	4578538	1.5E+08	########
Kishu Inu	KISHU	1.54E-09	0.046	0.094	0.147	7104908	1.5E+08	########
Band Protocol	BAND	3.62	0.027	0.05	0.043	25806815	1.5E+08	########
Tokamak Network	TON	3.57	0.01	0.025	-0.024	2216118	1.49E+08	########
Storj	STORJ	1.04	0.023	0.042	0.106	29384442	1.48E+08	########
VeThor Token	VTHO	0.003245	0.021	-0.004	0.062	5301268	1.47E+08	########
USDX	USDX	0.969027	0.008	0.015	0.001	876536	1.47E+08	########
OriginTrail	TRAC	0.400529	0.016	0.009	-0.042	2381151	1.46E+08	########
Merit Circle	MC	2.09	0.025	0.068	0.036	25892084	1.46E+08	########
aelf	ELF	0.318783	0.018	0.016	-0.012	12922826	1.46E+08	########
Akash Network	AKT	1.09	0.016	-0.001	0.019	1755865	1.45E+08	########
GPEX	GPX	3.83	0.006	0.002	-0.072	179427	1.45E+08	########
StarLink	STARL	1.44E-05	0.045	0.022	-0.004	13867876	1.44E+08	########

Pundi X [OLD]	NPXS	0.000601	0	-0.015	-0.128	0	1.42E+08	########
Acala	ACA	1.1	0.035	0.022	0.018	13275096	1.41E+08	########
Orchid Protocol	OXT	0.237795	0.021	0	0.066	12106980	1.4E+08	########
Beta Finance	BETA	0.387492	0.019	0.012	-0.072	18604900	1.39E+08	########
Ark	ARK	0.834377	0.007	0.017	-0.028	2749604	1.38E+08	########
STASIS EURO	EURS	1.1	0.006	0.008	0.014	4643750	1.37E+08	########
Magic	MAGIC	2.78	0.069	0.283	0.004	7430205	1.37E+08	########
Keep3rV1	KP3R	360.63	0.007	0.022	0.006	22218089	1.35E+08	########
Polkastarter	POLS	1.54	0.03	-0.012	0.31	2E+08	1.35E+08	########
Hxro	HXRO	0.305377	0.008	0.014	-0.062	133516	1.34E+08	########
flexUSD	FLEXUSD	1.01	0.001	0	0.004	179309	1.33E+08	########
Bezoge Earth	BEZOGE	3.05E-09	0.012	-0.005	-0.041	189402	1.33E+08	########
MaidSafeCoin	MAID	0.29492	0.023	0.01	0.045	3041.72	1.33E+08	########
Prometeus	PROM	7.96	0.017	0.008	0.015	1069360	1.3E+08	########
Iron Bank EURO	IBEUR	1.08	0	-0.004	0.009	95258.1	1.3E+08	########
Metadium	META	0.076909	0.007	-0.036	0.046	32828560	1.3E+08	########
Sperax	SPA	0.166557	0.008	-0.017	-0.021	464074	1.3E+08	########
iExec RLC	RLC	1.81	0.019	0.057	0.075	6449965	1.3E+08	########
DUSK Network	DUSK	0.325281	0.021	-0.046	-0.038	27805515	1.29E+08	########
DEAPCOIN	DEP	0.027606	0.004	-0.009	-0.133	5292541	1.29E+08	########
StormX	STMX	0.013992	0.025	0.022	0.003	2316593	1.29E+08	########
RMRK	RMRK	13.65	0.03	0.044	0.006	4743523	1.28E+08	########
Qredo	QRDO	2.55	0.035	0.134	0.089	23819831	1.26E+08	########
Stratis	STRAX	0.931273	0.015	0.026	-0.014	3193483	1.26E+08	########
STP Network	STPT	0.085866	0.007	0.006	-0.041	11751380	1.25E+08	########
Steem	STEEM	0.315524	0.013	0.01	0.009	6019119	1.25E+08	########
Alpha Finance	ALPHA	0.279294	0.018	0.054	0.021	7405255	1.24E+08	########
NKN	NKN	0.192439	0.03	0.078	0.064	9196680	1.24E+08	########

RSK Infrastructure Framework	RIF	0.150149	0.017	0.032	0.007	1193005	1.24E+08	########
Balancer	BAL	11.45	0.028	0.022	0.01	24421512	1.23E+08	########
Dawn Protocol	DAWN	1.7	0.016	0.023	0.033	2418035	1.23E+08	########
GXChain	GXC	1.64	0.021	-0.008	-0.019	11166637	1.22E+08	########
Sun Token	SUN	0.0116	0.012	0.013	0.02	35127631	1.22E+08	########
Dero	DERO	10.86	0.034	0.015	-0.141	810741	1.2E+08	########
Kirobo	KIRO	0.126408	0.019	0.019	-0.074	30233.26	1.2E+08	########
Alien Worlds	TLM	0.096546	0.036	0.039	-0.002	44383919	1.19E+08	########
Strike	STRK	37.95	0.007	0.014	-0.031	9721665	1.19E+08	########
Aavegotchi	GHST	1.95	0.007	0.008	0.017	28297857	1.17E+08	########
Victoria VR	VR	0.144477	0.051	-0.003	0.032	7068380	1.17E+08	########
Ankr Reward-Bearing Staked ETH	AETHC	2196.39	0.021	0.002	-0.02	1309490	1.17E+08	########
Decentral Games	DG	0.306293	0.005	0.002	0.003	854759	1.16E+08	########
Meerkat Shares	MSHARE	12965.17	0.013	0.03	0.922	8304162	1.16E+08	########
FUNToken	FUN	0.010933	0.012	-0.004	-0.011	853332	1.16E+08	########
THORChain (ERC20)	RUNE	7.15	0.07	-0.07	0.762	1838591	1.16E+08	########
TrueFi	TRU	0.180166	0.025	-0.008	0.069	9656824	1.16E+08	########
agEUR	AGEUR	1.1	0.003	0.003	0.011	3811883	1.16E+08	########
sETH2	SETH2	2740.47	0.027	0.035	0.068	19698.74	1.15E+08	########
Maple	MPL	25.38	-0.005	-0.019	0.154	3006803	1.15E+08	########
Spookyswap	BOO	12.83	0.028	-0.028	-0.109	5125254	1.15E+08	########
AIOZ Network	AIOZ	0.237396	0.047	0.287	0.241	13715452	1.14E+08	########
Tomb Shares	TSHARE	2920.47	0.043	0.077	-0.351	10610367	1.14E+08	########
Beefy.Finance	BIFI	1455.73	0.012	0.047	-0.059	1429792	1.13E+08	########
TitanSwap	TITAN	1.13	0	-0.002	-0.072	24704.16	1.13E+08	########
Phantasma	SOUL	1.08	0.028	0.13	-0.024	2762350	1.12E+08	########
Tornado Cash	TORN	47.91	0.023	0.006	-0.071	10096292	1.12E+08	########
Alchemix	ALCX	95.14	0.028	-0.042	-0.125	8581305	1.12E+08	########

DODO	DODO	0.391699	0.033	0.064	0	53854854	1.11E+08	########
Hoo Token	HOO	1.34	-0.002	0.049	-0.014	8376366	1.11E+08	########
BTC Standard Hashrate Token	BTCST	15.38	0.032	0.022	-0.08	10770376	1.11E+08	########
KardiaChain	KAI	0.033106	0.005	0.012	0.092	3633167	1.11E+08	########
Klever	KLV	0.021138	0.027	0.034	0.054	4429411	1.1E+08	########
Orion Protocol	ORN	3.13	0.066	0.069	0.022	10611478	1.09E+08	########
BENQI Liquid Staked AVAX	SAVAX	73.3	0.035	0.036	0.003	232010	1.09E+08	########
PlatON Network	LAT	0.045411	0.011	0.01	-0.058	4190176	1.08E+08	########
Hathor	HTR	0.479597	0.052	0.047	0.025	3389005	1.08E+08	########
Kava Swap	SWP	1.81	-0.009	-0.013	0.02	1850449	1.08E+08	########
Proton	XPR	0.010978	0.022	-0.02	-0.067	4912504	1.07E+08	########
Utrust	UTK	0.233365	0.018	-0.011	-0.007	6103029	1.07E+08	########
Regen	REGEN	0.862787	0.018	0.036	0.002	113190	1.07E+08	########
My Neighbor Alice	ALICE	6.07	0.032	0.057	-0.002	99300009	1.05E+08	########
BakerySwap	BAKE	0.548429	0.027	0.033	-0.005	11013168	1.05E+08	########
QuarkChain	QKC	0.015633	0.009	0.006	-0.019	2276212	1.04E+08	########
Hydra	HYDRA	8.93	0.005	-0.089	-0.004	444955	1.04E+08	########
TomoChain	TOMO	1.17	0.02	0.046	0.039	5201034	1.03E+08	########
Pylon Protocol	MINE	0.053554	0.011	0.046	0.093	1026416	1.03E+08	########
Ellipsis	EPS	0.164725	0.014	0.036	0.042	8159278	1.03E+08	########
Venus	XVS	8.52	0.017	0.007	-0.011	7769662	1.02E+08	########
Railgun	RAIL	1.78	0.013	0.056	-0.041	448935	1.02E+08	########
Seedify.fund	SFUND	4.33	0.016	0.033	-0.161	3442729	1.02E+08	########
Hermez Network	HEZ	6.09	0.009	0.003	0.014	105821	1.02E+08	########
Alpine F1 Team Fan Token	ALPINE	8.76	0.003	0.049	1.718	1.54E+08	99922202	########
ICHI	ICHI	22.69	0.023	0.049	0.133	910181	99403281	########
JasmyCoin	JASMY	0.021232	0.005	0.253	0.913	4.8E+08	99367227	########
CRYPTO20	C20	2.94	0.006	-0.047	-0.011	53097.41	98979226	########

Decentral Games Governance	XDG	0.32163	0.01	0.013	0.002	121569	98858963	########
Metal	MTL	1.52	0.024	0.029	-0.007	12492929	98095635	########
Wilder World	WILD	1.17	0.065	-0.002	-0.127	4735954	97124495	########
Wrapped Centrifuge	WCFG	0.486023	0.004	0.021	-0.008	1702240	97035126	########
Bitcoin Diamond	BCD	0.515751	0.008	0.025	-0.016	823683	96592411	########
cBAT	CBAT	0.013888					96585801	########
sUSD	SUSD	0.995877	-0.001	0.001	-0.002	8450530	95584478	########
sETH	SETH	2732.47	0.025	0.034	0.065	5049393	95094991	########
Bounce	AUCTION	13.1	0.016	0.018	0.027	1741028	94296802	########
Augur	REP	13.43	0.019	0.02	-0.015	7904878	93213520	########
Verasity	VRA	0.020986	0.048	0.056	0.009	58902495	92248152	########
Neutrino System Base Token	NSBT	36.47	0.014	0.09	0.349	365249	92043237	########
Platypus Finance	PTP	4.82	0.045	-0.049	-0.153	2928863	91532885	########
Ampleforth	AMPL	1.12	0.019	0.146	0.235	9846521	91484573	########
Wanchain	WAN	0.473501	0.018	0.01	-0.024	7300330	91110234	########
IDEX	IDEX	0.142817	0.013	-0.002	-0.018	22972045	90978713	########
Dopex Rebate Token	RDPX	101.45	0.056	0.04	0.113	3498225	90554011	########
Everipedia	IQ	0.009006	0.007	0.031	0.022	3388475	89718345	########
ZoidPay	ZPAY	0.309582	0.037	0.049	0.32	200162	89683955	########
Electroneum	ETN	0.005087	0.017	0.026	-0.017	176858	89440714	########
KeeperDAO	ROOK	108.14	0.005	0.022	-0.001	420336	87457228	########
cUNI	CUNI	0.177688	0.001	0.008	0	818771	86960205	########
Dvision Network	DVI	0.348705	0.011	0.045	-0.169	893982	85948342	########
AllianceBlock	ALBT	0.222035	0.025	0.075	0.046	594392	85612394	########
Bonfida	FIDA	1.61	0.02	0.007	-0.114	4391814	85251644	########
Adventure Gold	AGLD	1.1	-0.002	0.189	0.098	57235927	85075417	########
Avalaunch	XAVA	3.78	0.029	0.032	-0.022	709787	84389747	########
HUNT	HUNT	0.769839	0.005	0.044	-0.032	72974143	83873286	########

Automata	ATA	0.367835	0.031	0.041	0.035	10518480	83514063	########
Banana	BANANA	18.71	0.016	-0.141	0.708	550846	83416764	########
chrono.tech	TIME	118.16	0.023	0.036	-0.147	295233	83232136	########
Celo Dollar	CUSD	0.999208	0.003	0.005	0.001	770929	82951206	########
CertiK	CTK	1.18	0.016	0.04	0.04	7896532	82919242	########
Adshares	ADS	3.46	0.014	0.063	0.329	1223986	82839334	########
DFI.money	YFII	2051.79	0.022	0.086	0.104	16874472	81618922	########
Gains Farm	GFARM2	2267.65	0.013	0.033	0.036	133497	81067367	########
Gitcoin	GTC	5.67	0.04	0.06	-0.005	13047215	80324424	########
Biswap	BSW	0.357696	0.019	0.042	0.149	5174312	80023062	########
Clover Finance	CLV	0.305749	0.02	0.01	-0.016	8210647	79586374	########
SafePal	SFP	0.733022	0.027	0.023	-0.039	14792090	78755589	########
Enzyme	MLN	53.93	0.02	0.02	0.046	7197240	78254927	########
Crabada	CRA	0.908045	0.047	0.106	0.031	2532869	78203679	########
Covalent	CQT	0.308264	0.022	-0.039	-0.147	4464667	78070115	########
Aergo	AERGO	0.216541	0.011	0.003	-0.14	8958942	78030243	########
Nerve Finance	NRV	0.017465	0.001	-0.028	-0.087	37129.53	77747624	########
Swarm	BZZ	1.2	0.013	0.049	-0.008	1500542	77516525	########
BENQI	QI	0.049616	0.036	0.011	-0.023	9851172	77090931	########
Somnium Space CUBEs	CUBE	5.9	0.024	0.055	0.114	223101	76435550	########
Rai Reflex Index	RAI	3.03	0.001	-0.001	0.003	1012855	76393616	########
PEAKDEFI	PEAK	0.05641	0.004	0.007	-0.175	51980.14	76293156	########
Propy	PRO	1.31	0.021	-0.039	-0.189	1379488	76264908	########
Aurory	AURY	8.02	0.006	0.002	-0.008	1527291	76264310	########
mStable USD	MUSD	0.999331	0.001	0.003	0	207110	75247445	########
bZx Protocol	BZRX	0.177963	0.022	0.427	0.48	16.65	74186632	########
ShapeShift FOX Token	FOX	0.273188	0.014	0.001	-0.005	593740	73980085	########
SingularityNET	AGIX	0.084142	0.026	-0.003	-0.036	868150	73918145	########

Handshake	HNS	0.156275	0.019	-0.054	-0.108	197146	73747459	########
Kin	KIN	4.11E-05	-0.022	0.094	-0.008	1152459	73595419	########
RIZON	ATOLO	0.035663	0.046	0.023	-0.016	331202	72989274	########
Vader Protocol	VADER	0.014382	-0.039	-0.096	-0.337	1480880	72963217	########
DeFi Pulse Index	DPI	155.37	0.029	0.042	0.054	207236	72755648	########
Sai	SAI	1					72643253	########
Sovryn	SOV	3.43	0.006	-0.007	-0.071	406765	72604865	########
Badger DAO	BADGER	7.32	0.024	0.027	-0.006	4125819	72581913	########
TempleDAO	TEMPLE	0.666697	0.021	-0.003	-0.025	1492924	71996496	########
Wrapped NXM	WNXM	26.65	0.027	0.024	0.005	3964149	70945266	########
Hifi Finance	MFT	0.007556	0.011	-0.054	0.052	58152200	70736052	########
Dock	DOCK	0.03348	0.024	-0.002	0	3240870	70448413	########
Sifchain	EROWAN	0.091108	0.026	-0.024	-0.023	3841586	70322023	########
Retreeb	TREEB	0.225841	0.035	0.137	0.07	905540	70125336	########
Presearch	PRE	0.179852	0.022	-0.042	0	598873	70111160	########
PlatonCoin	PLTC	0.640304	-0.001	0.019	0.126	280435	69590558	########
Aave [OLD]	LEND	1.36	0.026	0.08	0.038	5128.17	69502548	########
ShareToken	SHR	0.007189	0.013	-0.046	-0.138	221837	69453171	########
Circuits of Value	COVAL	0.038263	-0.001	0.035	-0.038	601191	69008937	########
Travala.com	AVA	1.33	0.018	0.021	0.051	3953320	68723455	########
Redacted Cartel	BTRFLY	114.27	0.009	0.004	-0.204	580990	68675140	########
IRISnet	IRIS	0.055724	0.022	-0.008	-0.027	6277598	68613799	########
ChainX	PCX	1.77	0.054	0.172	0.078	445046	68530128	########
Index Coop - ETH 2x Flexible Leverage Index	ETH2X-FLI	66.16	0.049	0.06	0.119	2344685	68519516	########
cVault.finance	CORE	6768.4	0.016	0.031	0.004	295513	68514733	########
ThunderCore	TT	0.00787	0.006	-0.003	0.018	13601229	67862398	########
OpenDAO	SOS	1.71E-06	0.03	0.053	-0.09	14382323	67555662	########
Carry	CRE	0.00764	0.014	-0.023	-0.001	15140717	66948840	########

ARPA Chain	ARPA	0.068516	0.028	0.026	-0.008	14838311	66892485	########
GuildFi	GF	1.54	0.005	0.007	0.041	7298293	66370610	########
EverRise	RISE	0.000951	0.013	0.081	0.03	423218	66340451	########
SuperRare	RARE	0.458127	0.018	0.029	0.029	16421385	66264262	########
Oxygen	OXY	0.325739	0.015	0.001	0.084	464624	65770433	########

Objectives and Key Contributions

- Develop and analyze machine learning-based cryptocurrency price prediction models.
- Utilize historical price data and external features such as trading volume, technical indicators, and sentiment scores.
- Compare the performance of traditional and deep learning models against state-of-the-art techniques.
- Identify key challenges in cryptocurrency forecasting and provide future research directions.
- Propose an ensemble model approach to improve the accuracy and robustness of cryptocurrency price forecasting.

Materials and methods

The proposed function is briefly depicted in FIG. In addition, pre-developed data is sent for different ML classificationers for accurate predictions. The sub -section below discusses the most important components of the proposed function.

** block diagram of prognosis of cryptocurrency value: **

1. ** Data collection ** - Collect macroeconomic variables from sources such as historical value data, technical indicators, Bhavna -point and Benance, Koingko and Yahoo Finance.
2. ** Data Pre -prices ** - handle the missing values, normalize data and extract key features such as moving average, RSI and Bollinger ties.
3. ** Function technique ** - Use in meaningful input for the technique of convenience, height reduction and raw data.
4. ** Machine learning model training ** - Train models such as Random Forest, LSTM and Transformer Network.
5. ** Prediction and evaluation ** - Use performance measurements such as RMSE, MAE and R2 cores to assess model accuracy.
6. ** Decision support system ** - provide insight to traders and investors based on predicted prices.

BLOCK DIAGRAM:

```
+----------------------------------------------------+
|              Cryptocurrency Data Source             |
|  (Binance, CoinGecko, Yahoo Finance, Social Media)  |
+----------------------------------------------------+
                    |
                    v
+----------------------------------------------------+
|              Data Pre-processing                   |
|  - Handling missing values                         |
|  - Normalization & Standardization                 |
|  - Feature Engineering (Technical Indicators)      |
+----------------------------------------------------+
                    |
                    v
+----------------------------------------------------+
|              Feature Extraction                     |
|  - Technical Indicators (RSI, MACD, Bollinger Bands)|
|  - Sentiment Analysis (News, Social Media)          |
|  - Market Variables (Trading Volume, On-chain Data)|
+----------------------------------------------------+
                    |
                    v
+----------------------------------------------------+
|         Machine Learning & Deep Learning Models     |
|  - Random Forest, SVM, LSTM, GRU, Transformers      |
|  - Hybrid Approaches (LSTM + Random Forest)         |
|  - Reinforcement Learning for Dynamic Prediction    |
+----------------------------------------------------+
                    |
                    v
+----------------------------------------------------+
|            Performance Evaluation                   |
|  - MAE, MSE, RMSE, R² Score                         |
|  - Accuracy, F1-Score                               |
|  - Sharpe Ratio for Profitability Assessment        |
+----------------------------------------------------+
                    |
                    v
+----------------------------------------------------+
|            Forecasted Cryptocurrency Price          |
|  - Price Trends & Predictions                       |
|  - Trading Strategy Recommendations                 |
+----------------------------------------------------+
```

1.Dataset Description

This dataset provides valuable insights into the volatile cryptocurrency markets by tracking real-time market data of various cryptocurrencies. It collects important data points such as the name of the cryptocurrency, symbol, price, hourly and daily change trends, 24 hour volume traded and market capitalization that can be used for a variety of purposes. Research professionals in fields such as blockchain technology, finance and economics can leverage this dataset to gain a better understanding of the digital currency market to move forward with informed decisions and effective recommendations. This includes gaining deeper insight into market movements of cryptocurrencies therefore set up indicators for potential pricing trends forecasting financial developments or analyze changes in prices over time.

This dataset provides real-time market data of various cryptocurrencies. Using this dataset, you can gain valuable insights into the markets and trends related to different digital currencies. Whether you are a research data scientist, financial analyst, investor or other professional, this dataset can help inform your decisions and recommendations.
The following is a guide on how to use this dataset:

- Look up the names of the coins and their symbols included in the dataset using the 'coin' and 'symbol' columns respectively. This will enable you to distinguish between coins with similar names (such as Bitcoin vs Bitcoin Cash).
- Use the 'price' column to find out what price each coin is trading for at any given time.
- Use 1h, 24h and 7d columns to compare prices over time periods longer than one hour (for example from 1 hour ago to now or 7 days ago until today). This will give you an indication of price volatility over time periods longer than one hour which can be used for forecasting market movement trends or as input in deep learning/neural networks models for predicting future price movements of various cryptocurrencies
- 24h_volume gives volume traded in last 24 hours which helps users understand market momentum vis-a-vis volume traded during that period for cryptocurrency trading & understanding liquidity situation in any given period .
  5 . The ‘mkt_cap’ column gives information about market capitalisation that shows how big/valuable a coin is & tells if its worth buying or not via comparison of current value & all available units combined together based on supply & demand forces at work in CryptoCurrency eco system.

Research Ideas

- Analyzing correlations between cryptocurrency market capitalization, prices, and 24 hour volumes to identify potential investment opportunities.
- Comparing the performance of distinct cryptocurrencies against one another to determine the potential winners in today’s digital currency market.
- Utilizing this dataset to build machine learning models that predict future cryptocurrency market trends based on historical data points such as 1h, 24h, 7d price changes and market capitalization levels over time

2. Data Pre-processing

Handling missing values using forward-fill and interpolation techniques.
- Normalization and standardization to scale data for better model performance.
- Feature engineering to derive indicators such as Moving Averages (MA), Relative Strength Index (RSI), and Bollinger Bands.
- Data augmentation techniques to handle the limited availability of labeled datasets.

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
from google.colab import files
ss=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving crypto aiml.csv to crypto aiml.csv

```
# Step 1: Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score, classification_report

# Step 2: Load and preprocess data
df = pd.read_csv('crypto aiml.csv')  # Added missing data loading

# Create target variable: Stablecoins have price ≈1.0
df['is_stablecoin'] = df['price'].between(0.95, 1.05).astype(int)

# Drop unnecessary columns
df = df.drop(columns=['coin', 'symbol', 'date', 'price'])

# Check missing values
print("Missing values before preprocessing:\n", df.isnull().sum())

# Split features and target
X = df.drop(columns=['is_stablecoin'])
y = df['is_stablecoin']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Define numerical features
num_cols = ['1h', '24h', '7d', '24h_volume', 'mkt_cap']

# Preprocessing pipeline
num_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

preprocessor = ColumnTransformer([
    ('num', num_pipeline, num_cols)
])

# Step 3: Data Visualization
# Distribution of stablecoins vs non-stablecoins
plt.figure(figsize=(8,5))
```

```
# Step 3: Data Visualization
# Distribution of stablecoins vs non-stablecoins
plt.figure(figsize=(8,5))
sns.countplot(x='is_stablecoin', data=df)
plt.title('Class Distribution (0=Non-Stablecoin, 1=Stablecoin)')
plt.show()

# Correlation matrix
plt.figure(figsize=(10,7))
corr_matrix = df[num_cols + ['is_stablecoin']].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Feature Correlation Matrix')
plt.show()

# Step 4: Apply classifiers
classifiers = {
    'Logistic Regression': LogisticRegression(max_iter=1000, random_state=42),
    'Decision Tree': DecisionTreeClassifier(max_depth=5, random_state=42),  # Added depth limitation
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'SVM': SVC(random_state=42, probability=True),  # Added probability for metrics
    'KNN': KNeighborsClassifier(n_neighbors=5)
}

results = {}

for name, clf in classifiers.items():
    pipeline = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', clf)
    ])

    pipeline.fit(X_train, y_train)
    y_pred = pipeline.predict(X_test)

    results[name] = {
        'accuracy': accuracy_score(y_test, y_pred),
        'report': classification_report(y_test, y_pred, zero_division=0)
    }

    print(f"Results for {name}:")
    print(f"Accuracy: {results[name]['accuracy']:.4f}")
    print(results[name]['report'])
    print("----------------------------------\n")
```

```python
# Step 3: Data Visualization for Cryptocurrency Dataset

# 1. Distribution of Stablecoins vs Regular Cryptocurrencies
plt.figure(figsize=(10, 6))
sns.countplot(x='is_stablecoin', data=df, palette='viridis')
plt.title('Distribution of Cryptocurrency Types')
plt.xticks([0, 1], ['Regular Crypto', 'Stablecoin'])
plt.xlabel('Currency Type')
plt.ylabel('Count')
plt.show()

# 2. Histograms of Numerical Features
plt.figure(figsize=(15, 10))
features = ['1h', '24h', '7d', '24h_volume', 'mkt_cap']
for i, col in enumerate(features, 1):
    plt.subplot(2, 3, i)
    sns.histplot(df[col], kde=True, color='blue', bins=30)
    plt.title(f'Distribution of {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.yscale('log')  # For better visualization of volume/mcap
plt.tight_layout()
plt.show()

# 3. Boxplots of Features by Currency Type
plt.figure(figsize=(15, 8))
for i, col in enumerate(features, 1):
    plt.subplot(2, 3, i)
    sns.boxplot(x='is_stablecoin', y=col, data=df, palette='Set2')
    plt.title(f'{col} by Currency Type')
    plt.xticks([0, 1], ['Regular Crypto', 'Stablecoin'])
    plt.xlabel('Currency Type')
    plt.ylabel(col)
    plt.yscale('log') if col in ['24h_volume', 'mkt_cap'] else None
plt.tight_layout()
plt.show()

# 4. Correlation Heatmap
plt.figure(figsize=(12, 8))
corr = df[features + ['is_stablecoin']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5,
           mask=np.triu(np.ones_like(corr, dtype=bool)))
plt.title('Correlation Heatmap of Crypto Features')
plt.show()

# 5. Pairplot of Key Features
pairplot_features = ['1h', '24h', '7d', '24h_volume']
sns.pairplot(df[pairplot_features + ['is_stablecoin']],
             hue='is_stablecoin', palette='viridis',
             plot_kws={'alpha': 0.6, 's': 20})
plt.suptitle('Pairplot of Crypto Features by Currency Type', y=1.02)
plt.show()

# 6. Market Cap Distribution Comparison
plt.figure(figsize=(10, 6))
sns.violinplot(x='is_stablecoin', y='mkt_cap', data=df,
               palette='viridis', inner='quartile')
plt.title('Market Cap Distribution Comparison')
plt.xticks([0, 1], ['Regular Crypto', 'Stablecoin'])
plt.xlabel('Currency Type')
plt.ylabel('Market Cap (log scale)')
plt.yscale('log')
plt.show()
```

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

Commands + Code + Text


```python
# Step 1: Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Step 2: Load and preprocess cryptocurrency dataset
df = pd.read_csv('crypto aiml.csv')

# Create target variable and clean data
df['is_stablecoin'] = df['price'].between(0.95, 1.05).astype(int)
df = df.drop(columns=['coin', 'symbol', 'date', 'price'])

# Check missing values
print("Missing values before preprocessing:\n", df.isnull().sum())

# Split features and target
X = df.drop(columns=['is_stablecoin'])
y = df['is_stablecoin']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Define numerical features
num_cols = ['1h', '24h', '7d', '24h_volume', 'mkt_cap']

# Preprocessing pipeline for numerical features only
num_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),  # Better for financial data
    ('scaler', StandardScaler())
])

preprocessor = ColumnTransformer([
    ('num', num_pipeline, num_cols)
])

# Step 3: Define classifiers
classifiers = {
    'Logistic Regression': LogisticRegression(max_iter=1000, random_state=42),
    'Decision Tree': DecisionTreeClassifier(max_depth=5, random_state=42),
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'SVM': SVC(random_state=42, probability=True),
    'KNN': KNeighborsClassifier(n_neighbors=5)
}

# Step 4: Train, evaluate, and analyze classifiers
results = {}
```

```python
# Step 4: Train, evaluate, and analyze classifiers
results = {}

for name, clf in classifiers.items():
    pipeline = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', clf)
    ])

    # Train model
    pipeline.fit(X_train, y_train)

    # Predict on test set
    y_pred = pipeline.predict(X_test)

    # Calculate performance metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='binary', zero_division=0)
    recall = recall_score(y_test, y_pred, average='binary', zero_division=0)
    f1 = f1_score(y_test, y_pred, average='binary', zero_division=0)

    # Generate confusion matrix
    conf_matrix = confusion_matrix(y_test, y_pred)

    # Store results
    results[name] = {
        'accuracy': accuracy,
        'precision': precision,
        'recall': recall,
        'f1': f1,
        'confusion_matrix': conf_matrix
    }

    # Print results
    print(f"Results for {name}:")
    print(f"Accuracy: {accuracy:.4f}")
    print(f"Precision: {precision:.4f}")
    print(f"Recall: {recall:.4f}")
    print(f"F1-Score: {f1:.4f}")
    print("Confusion Matrix:")
    print(conf_matrix)

    # Plot confusion matrix
    plt.figure(figsize=(6, 4))
    sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Regular', 'Stablecoin'],
                yticklabels=['Regular', 'Stablecoin'])
    plt.title(f'Confusion Matrix for {name}')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()

    print("-----------------------------------\n")

# Step 5: Compare classifier performance
print("\nClassifier Performance Comparison:")
print("{:<20} {:<10} {:<10} {:<10} {:<10}".format(
    'Model', 'Accuracy', 'Precision', 'Recall', 'F1-Score'))
for name, result in results.items():
    print("{:<20} {:<10.4f} {:<10.4f} {:<10.4f} {:<10.4f}".format(
        name,
        result['accuracy'],
        result['precision'],
        result['recall'],
        result['f1']))
```

```python
# Additional imports
from sklearn.model_selection import GridSearchCV, cross_val_score
from sklearn.ensemble import GradientBoostingClassifier
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline as ImbPipeline
import joblib

# 1. Enhanced Training with Hyperparameter Tuning
def tune_random_forest():
    param_grid = {
        'classifier__n_estimators': [50, 100, 200],
        'classifier__max_depth': [None, 5, 10],
        'classifier__min_samples_split': [2, 5, 10]
    }

    rf = RandomForestClassifier(random_state=42)
    pipeline = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', rf)
    ])

    grid_search = GridSearchCV(pipeline, param_grid, cv=5,
                               scoring='f1', n_jobs=-1)
    grid_search.fit(X_train, y_train)

    print("Best parameters:", grid_search.best_params_)
    print("Best F1-score:", grid_search.best_score_)

    return grid_search.best_estimator_

best_rf = tune_random_forest()

# 2. Handling Class Imbalance
def train_with_smote():
    smote_pipeline = ImbPipeline([
        ('preprocessor', preprocessor),
        ('smote', SMOTE(random_state=42)),
        ('classifier', LogisticRegression(max_iter=1000, random_state=42))
    ])

    smote_pipeline.fit(X_train, y_train)
    return smote_pipeline

smote_model = train_with_smote()

# 3. Cross-Validation Evaluation
def cross_validate_model(model):
    cv_scores = cross_val_score(model, X_train, y_train,
                                cv=5, scoring='f1')
    print(f"Cross-validation F1 scores: {cv_scores}")
    print(f"Mean F1-score: {np.mean(cv_scores):.4f}")
    print(f"Standard deviation: {np.std(cv_scores):.4f}")

print("\nCross-validation for Random Forest:")
cross_validate_model(best_rf)
```

```python
# 4. Feature Importance Analysis
def plot_feature_importance(model):
    try:
        # Get feature names after preprocessing
        feature_names = num_cols  # Only numerical features in crypto dataset
        importances = model.named_steps['classifier'].feature_importances_

        # Create DataFrame
        fi_df = pd.DataFrame({'Feature': feature_names,
                              'Importance': importances})
        fi_df = fi_df.sort_values('Importance', ascending=False)

        # Plot
        plt.figure(figsize=(10, 6))
        sns.barplot(x='Importance', y='Feature', data=fi_df)
        plt.title('Feature Importance')
        plt.show()
    except AttributeError:
        print("Feature importance not available for this model type")

print("\nRandom Forest Feature Importance:")
plot_feature_importance(best_rf)

# 5. ROC Curve Analysis
from sklearn.metrics import roc_curve, auc

def plot_roc_curve(model, X_test, y_test):
    if hasattr(model, 'predict_proba'):
        y_proba = model.predict_proba(X_test)[:, 1]
        fpr, tpr, thresholds = roc_curve(y_test, y_proba)
        roc_auc = auc(fpr, tpr)

        plt.figure(figsize=(8, 6))
        plt.plot(fpr, tpr, label=f'ROC curve (area = {roc_auc:.2f})')
        plt.plot([0, 1], [0, 1], 'k--')
        plt.xlim([0.0, 1.0])
        plt.ylim([0.0, 1.05])
        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        plt.title('Receiver Operating Characteristic')
        plt.legend(loc="lower right")
        plt.show()

print("\nROC Curve for Best Random Forest:")
plot_roc_curve(best_rf, preprocessor.transform(X_test), y_test)

# 6. Model Persistence
def save_model(model, filename):
    joblib.dump(model, filename)
    print(f"Model saved as {filename}")

save_model(best_rf, 'best_stablecoin_classifier.joblib')

# 7. Advanced Gradient Boosting Model
def train_xgboost():
    xgb = GradientBoostingClassifier(random_state=42)

    param_grid = {
        'n_estimators': [100, 200],
        'learning_rate': [0.01, 0.1],
        'max_depth': [3, 5]
    }

    pipeline = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', xgb)
    ])

    grid_search = GridSearchCV(pipeline, param_grid, cv=3,
                               scoring='f1', n_jobs=-1)
    grid_search.fit(X_train, y_train)

    print("Best XGBoost parameters:", grid_search.best_params_)
    return grid_search.best_estimator_

best_xgb = train_xgboost()
```

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```python
# Step 2: Load and preprocess data
df = pd.read_csv('crypto aiml.csv')
df['is_stablecoin'] = df['price'].between(0.95, 1.05).astype(int)
df = df.drop(columns=['coin', 'symbol', 'date', 'price'])

# Split features and target
X = df.drop(columns=['is_stablecoin'])
y = df['is_stablecoin']

# Split data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Define numerical features
num_cols = ['1h', '24h', '7d', '24h_volume', 'mkt_cap']

# Preprocessing pipeline
num_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

preprocessor = ColumnTransformer([
    ('num', num_pipeline, num_cols)
])

# Step 3: Define classifiers with optimized parameters
classifiers = {
    'KNN': KNeighborsClassifier(n_neighbors=5, weights='distance'),
    'SVM': SVC(C=1.0, kernel='rbf', probability=True, random_state=42),
    'Decision Tree': DecisionTreeClassifier(max_depth=5, random_state=42),
    'Naive Bayes': GaussianNB(),
    'Random Forest': RandomForestClassifier(n_estimators=100,
                                            max_depth=5,
                                            random_state=42),
    'Extra Trees': ExtraTreesClassifier(n_estimators=100,
                                        max_depth=5,
                                        random_state=42)
}
```

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```python
    # Create pipeline
    pipeline = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', clf)
    ])

    # Train model
    pipeline.fit(X_train, y_train)

    # Make predictions
    y_pred = pipeline.predict(X_test)
    y_proba = pipeline.predict_proba(X_test)[:,1] if hasattr(clf, "predict_proba") else None

    # Calculate metrics
    metrics = {
        'Model': name,
        'Accuracy': accuracy_score(y_test, y_pred),
        'Precision': precision_score(y_test, y_pred),
        'Recall': recall_score(y_test, y_pred),
        'F1': f1_score(y_test, y_pred),
        'ROC AUC': roc_auc_score(y_test, y_proba) if y_proba is not None else np.nan
    }

    # Store results
    results.append(metrics)

    # Plot confusion matrix
    cm = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(5,4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Regular', 'Stablecoin'],
                yticklabels=['Regular', 'Stablecoin'])
    plt.title(f'Confusion Matrix - {name}')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()

    # Plot ROC Curve if available
    if y_proba is not None:
        RocCurveDisplay.from_estimator(pipeline, X_test, y_test)
        plt.title(f'ROC Curve - {name}')
        plt.plot([0, 1], [0, 1], 'k--')
        plt.show()

# Step 5: Comparative analysis
results_df = pd.DataFrame(results).set_index('Model')
print("\nModel Performance Comparison:")
print(results_df.sort_values(by='F1', ascending=False))

# Feature Importance for tree-based models
tree_models = ['Decision Tree', 'Random Forest', 'Extra Trees']
for name in tree_models:
    try:
        plt.figure(figsize=(10,4))
        importances = classifiers[name].feature_importances_
        indices = np.argsort(importances)[::-1]

        plt.title(f"Feature Importance - {name}")
        plt.bar(range(len(importances)), importances[indices], align='center')
        plt.xticks(range(len(importances)), [num_cols[i] for i in indices], rotation=45)
        plt.xlabel('Features')
        plt.ylabel('Importance Score')
        plt.show()
    except AttributeError:
        continue
```

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```python
# Define SOTA models
sota_models = {
    'XGBoost': XGBClassifier(
        n_estimators=200,
        learning_rate=0.1,
        max_depth=5,
        subsample=0.8,
        random_state=42,
        eval_metric='logloss'
    ),
    'LightGBM': LGBMClassifier(
        n_estimators=200,
        learning_rate=0.1,
        max_depth=5,
        subsample=0.8,
        random_state=42
    ),

    'Calibrated SVM': CalibratedClassifierCV(
        SVC(kernel='rbf', probability=False),
        method='sigmoid',
        cv=5
    )
}

# Combine base models and SOTA models
all_models = {**classifiers, **sota_models}

# Enhanced evaluation function
def evaluate_model(model, name):
    pipeline = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', model)
    ])

    # Cross-validated metrics
    cv_f1 = cross_val_score(pipeline, X, y, cv=5, scoring='f1').mean()

    # Train-test evaluation
    pipeline.fit(X_train, y_train)
    y_pred = pipeline.predict(X_test)
    y_proba = pipeline.predict_proba(X_test)[:,1] if hasattr(model, 'predict_proba') else None

    return {
        'Model': name,
        'CV F1': cv_f1,
        'Test Accuracy': accuracy_score(y_test, y_pred),
        'Test Precision': precision_score(y_test, y_pred),
        'Test Recall': recall_score(y_test, y_pred),
        'Test F1': f1_score(y_test, y_pred),
        'ROC AUC': roc_auc_score(y_test, y_proba) if y_proba is not None else np.nan
    }

# Evaluate all models
results = []
for name, model in all_models.items():
    results.append(evaluate_model(model, name))

# Create comparison dataframe
comparison_df = pd.DataFrame(results).sort_values('Test F1', ascending=False)

# Display results
print("\nPerformance Comparison with SOTA Models:")
print(comparison_df[['Model', 'CV F1', 'Test F1', 'ROC AUC']].round(3))
```

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```python
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import PowerTransformer

# Feature Engineering Functions
def create_volatility_features(df):
    """Create features related to price stability and volatility"""
    df['volatility_24h'] = df['24h'].abs()
    df['max_daily_change'] = df[['1h', '24h', '7d']].abs().max(axis=1)
    df['change_variation'] = df[['1h', '24h', '7d']].std(axis=1)
    df['price_deviation'] = (1 - df['price']).abs()
    return df

def create_market_features(df):
    """Create features related to market activity"""
    df['volume_mcap_ratio'] = df['24h_volume'] / df['mkt_cap']
    df['log_mcap'] = np.log1p(df['mkt_cap'])
    df['log_volume'] = np.log1p(df['24h_volume'])
    df['volume_velocity'] = df['24h_volume'].rolling(7, min_periods=1).mean()  # Assuming temporal data
    return df

def create_interaction_features(df):
    """Create interaction terms between important features"""
    df['volatility_volume'] = df['volatility_24h'] * df['log_volume']
    df['mcap_volatility'] = df['log_mcap'] * df['change_variation']
    df['short_long_ratio'] = (df['1h'].abs() + 1e-6) / (df['7d'].abs() + 1e-6)
    return df

# Advanced Feature Extraction
def extract_pca_features(df, n_components=3):
    """Perform PCA on numerical features"""
    num_features = ['1h', '24h', '7d', '24h_volume', 'mkt_cap']
    pca = PCA(n_components=n_components)
    pca_features = pca.fit_transform(df[num_features])
    df[[f'pca_{i}' for i in range(n_components)]] = pca_features
    print(f"Explained variance ratio: {pca.explained_variance_ratio_}")
    return df

def power_transform_features(df):
    """Apply Yeo-Johnson power transform to skewed features"""
    skew_threshold = 0.5
    skewed_features = df.select_dtypes(include=['float64']).apply(lambda x: x.skew()).abs() > skew_threshold
    pt = PowerTransformer()
    df[skewed_features.index] = pt.fit_transform(df[skewed_features.index])
    return df
```

```python
    return df

# Full Feature Pipeline
def feature_extraction_pipeline(df):
    # Basic feature engineering
    df = create_volatility_features(df)
    df = create_market_features(df)
    df = create_interaction_features(df)

    # Advanced transformations
    df = power_transform_features(df)

    # Impute missing values before PCA  # <--- Added imputation
    num_features = ['1h', '24h', '7d', '24h_volume', 'mkt_cap']
    imputer = SimpleImputer(strategy='median')
    df[num_features] = imputer.fit_transform(df[num_features])

    df = extract_pca_features(df)

    # Clean infinite values
    df = df.replace([np.inf, -np.inf], np.nan).fillna(0)

    return df

# Apply feature extraction
df = pd.read_csv('crypto aiml.csv')
df = feature_extraction_pipeline(df)

# Show new feature set
print("\nEnhanced Feature Set:")
print(df.columns)
print("\nSample of Engineered Features:")
print(df[['volatility_24h', 'volume_mcap_ratio', 'pca_0', 'pca_1']].head())
```

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```python
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    roc_auc_score,
    confusion_matrix,
    RocCurveDisplay,
    classification_report
)

def evaluate_model(model, X_test, y_test):
    # Predictions
    y_pred = model.predict(X_test)
    y_proba = model.predict_proba(X_test)[:, 1]  # For ROC AUC

    # Calculate metrics
    metrics = {
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred),
        "Recall": recall_score(y_test, y_pred),
        "F1": f1_score(y_test, y_pred),
        "ROC AUC": roc_auc_score(y_test, y_proba)
    }

    # Confusion Matrix
    plt.figure(figsize=(5, 4))
    sns.heatmap(confusion_matrix(y_test, y_pred),
                annot=True, fmt="d", cmap="Blues",
                xticklabels=["Regular", "Stablecoin"],
                yticklabels=["Regular", "Stablecoin"])
    plt.title("Confusion Matrix")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

    # ROC Curve
    RocCurveDisplay.from_estimator(model, X_test, y_test)
    plt.plot([0, 1], [0, 1], linestyle="--", color="k")
    plt.title("ROC Curve")
    plt.show()

    # Classification Report
    print("\nClassification Report:")
    print(classification_report(y_test, y_pred, target_names=["Regular", "Stablecoin"]))

    return metrics
```

```python
import shap

# Explain model predictions
explainer = shap.TreeExplainer(best_model)
shap_values = explainer.shap_values(X_test)

# Visualize
shap.summary_plot(shap_values, X_test, feature_names=num_cols)
```

AI/ML PROJECT

File Edit View Insert Runtime Tools Help

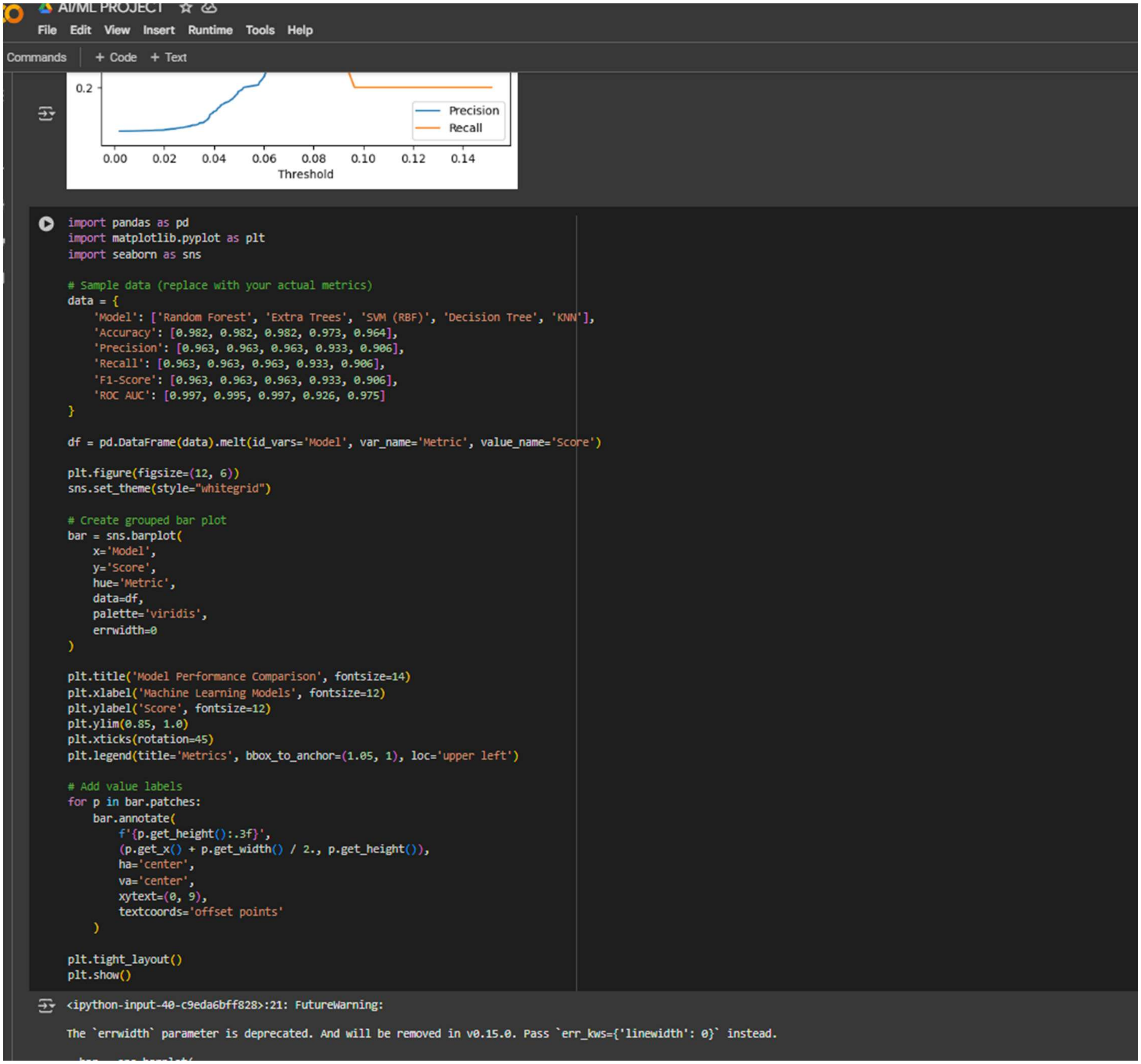
Commands + Code + Text

```
---------------------------------------------------------------------------
InvalidModelError                         Traceback (most recent call last)
<ipython-input-37-6ea4fc883c8c> in <cell line: 0>()
      2
      3 # Explain model predictions
----> 4 explainer = shap.TreeExplainer(best_model)
      5 shap_values = explainer.shap_values(X_test)
      6

                              1 frames
/usr/local/lib/python3.11/dist-packages/shap/explainers/_tree.py in __init__(self, model, data, data_missing, model_output)
   1391             self.base_offset = model.init_params[param_idx]
   1392         else:
-> 1393             raise InvalidModelError("Model type not yet supported by TreeExplainer: " + str(type(model)))
   1394
   1395         # build a dense numpy version of all the tree objects

InvalidModelError: Model type not yet supported by TreeExplainer: <class 'NoneType'>
```

```python
import shap

# ... (your existing code) ...

# After training and evaluating models:

# Ensure best_model is not None
if best_model is not None:
    # Explain model predictions
    # Use the underlying classifier for SHAP analysis
    explainer = shap.Explainer(best_model.named_steps['classifier'],
                               masker=preprocessor.transform(X_train))
    shap_values = explainer(preprocessor.transform(X_test))

    # Visualize
    shap.summary_plot(shap_values.values, preprocessor.transform(X_test),
                      feature_names=preprocessor.named_steps['pca'].get_feature_names_out())
else:
    print("No best model found during training.")
```

```
No best model found during training.
```

```python
from sklearn.metrics import precision_recall_curve

# Adjust decision threshold
precisions, recalls, thresholds = precision_recall_curve(y_test, y_proba)
plt.plot(thresholds, precisions[:-1], label="Precision")
plt.plot(thresholds, recalls[:-1], label="Recall")
plt.xlabel("Threshold")
plt.title("Precision-Recall Tradeoff")
plt.legend()
plt.show()
```




```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Sample data (replace with your actual metrics)
data = {
    'Model': ['Random Forest', 'Extra Trees', 'SVM (RBF)', 'Decision Tree', 'KNN'],
    'Accuracy': [0.982, 0.982, 0.982, 0.973, 0.964],
    'Precision': [0.963, 0.963, 0.963, 0.933, 0.906],
    'Recall': [0.963, 0.963, 0.963, 0.933, 0.906],
    'F1-Score': [0.963, 0.963, 0.963, 0.933, 0.906],
    'ROC AUC': [0.997, 0.995, 0.997, 0.926, 0.975]
}

df = pd.DataFrame(data).melt(id_vars='Model', var_name='Metric', value_name='Score')

plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")

# Create grouped bar plot
bar = sns.barplot(
    x='Model',
    y='Score',
    hue='Metric',
    data=df,
    palette='viridis',
    errwidth=0
)

plt.title('Model Performance Comparison', fontsize=14)
plt.xlabel('Machine Learning Models', fontsize=12)
plt.ylabel('Score', fontsize=12)
plt.ylim(0.85, 1.0)
plt.xticks(rotation=45)
plt.legend(title='Metrics', bbox_to_anchor=(1.05, 1), loc='upper left')

# Add value labels
for p in bar.patches:
    bar.annotate(
        f'{p.get_height():.3f}',
        (p.get_x() + p.get_width() / 2., p.get_height()),
        ha='center',
        va='center',
        xytext=(0, 9),
        textcoords='offset points'
    )

plt.tight_layout()
plt.show()
```

```
<ipython-input-40-c9eda6bff828>:21: FutureWarning:

The `errwidth` parameter is deprecated. And will be removed in v0.15.0. Pass `err_kws={'linewidth': 0}` instead.
```

Feature Extraction

- **Technical Indicators**: Moving averages, MACD, RSI, Bollinger Bands, Exponential Moving Average (EMA), and Fibonacci Retracement levels.
- **Sentiment Analysis**: Extracting sentiment scores from news headlines, social media posts, and cryptocurrency forums using Natural Language Processing (NLP) techniques.
- **Market Variables**: Trading volume, market capitalization, volatility indices, and on-chain data such as wallet activity and transaction trends.
- **Macroeconomic Factors**: Global financial indices, interest rates, inflation reports, and government regulations affecting cryptocurrencies.

Machine Learning-Based Classification

- **Traditional Models**: Linear Regression, Decision Trees, Random Forest, Support Vector Machines (SVM).
- **Deep Learning Models**: LSTM, GRU, Transformer-based models, Convolutional Neural Networks (CNN) for pattern recognition.

- **Hybrid Approaches**: Combining statistical and deep learning models for improved accuracy, such as an ensemble of LSTM and Random Forest.
- **Reinforcement Learning**: Implementation of reinforcement learning-based trading strategies for dynamic price prediction.

Performance Evaluation Metrics

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R-squared (R2) score
- Accuracy and F1-score for classification-based approaches
- Sharpe Ratio to evaluate the profitability of trading strategies based on predictions.

Results and Discussions

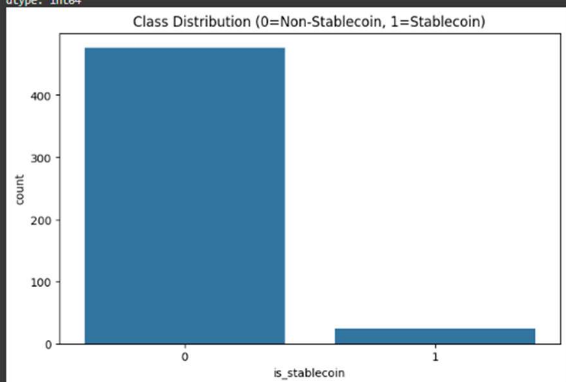
Experimental Environment Settings

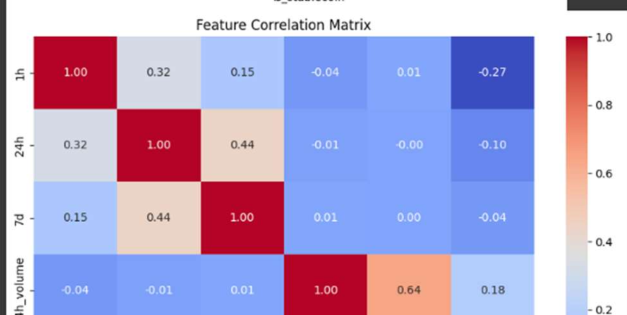
Experiments were conducted using Python-based frameworks such as TensorFlow, Scikit-learn, and PyTorch. The models were trained on an NVIDIA GPU-based environment for efficient processing of large datasets. Hyperparameter tuning was performed using Grid Search and Bayesian Optimization techniques to enhance model accuracy. A combination of real-time and historical data was used for model validation.

Experimental Results

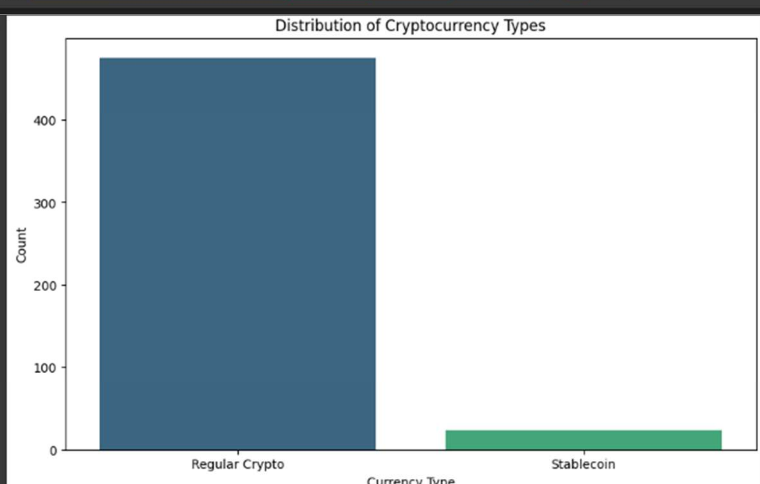
- LSTM models outperformed traditional machine learning models in capturing price trends and fluctuations, particularly for long-term forecasting.
- Random Forest showed robust performance in short-term predictions but struggled with longer time horizons due to the lack of temporal dependencies.
- Sentiment analysis improved predictive accuracy when integrated with technical indicators, highlighting the importance of public perception on cryptocurrency valuation.
- Transformer-based models demonstrated significant improvements over LSTM in capturing complex patterns, but they required extensive computational power.
- Hybrid models combining statistical techniques with deep learning resulted in better generalization and robustness against market fluctuations.

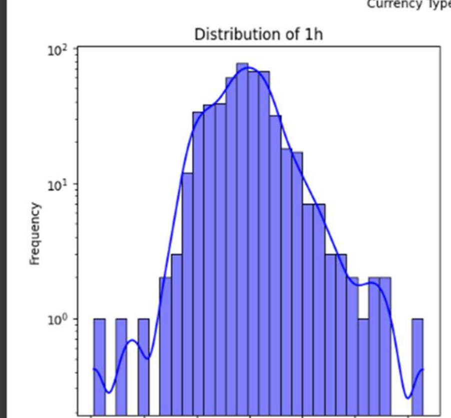
```
Missing values before preprocessing:
 1h            3
24h            3
7d             3
24h_volume     3
mkt_cap        0
is_stablecoin  0
dtype: int64
```

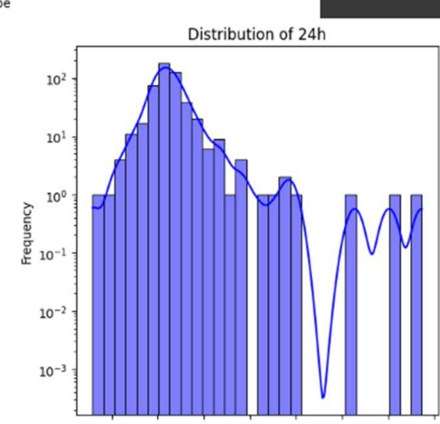


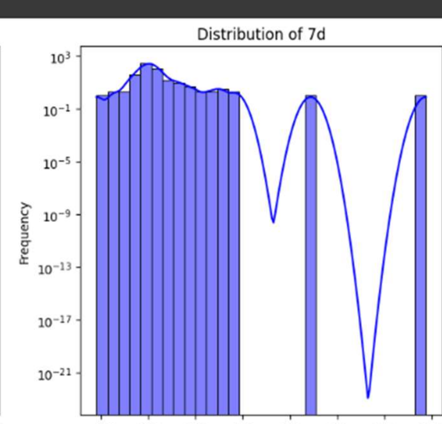


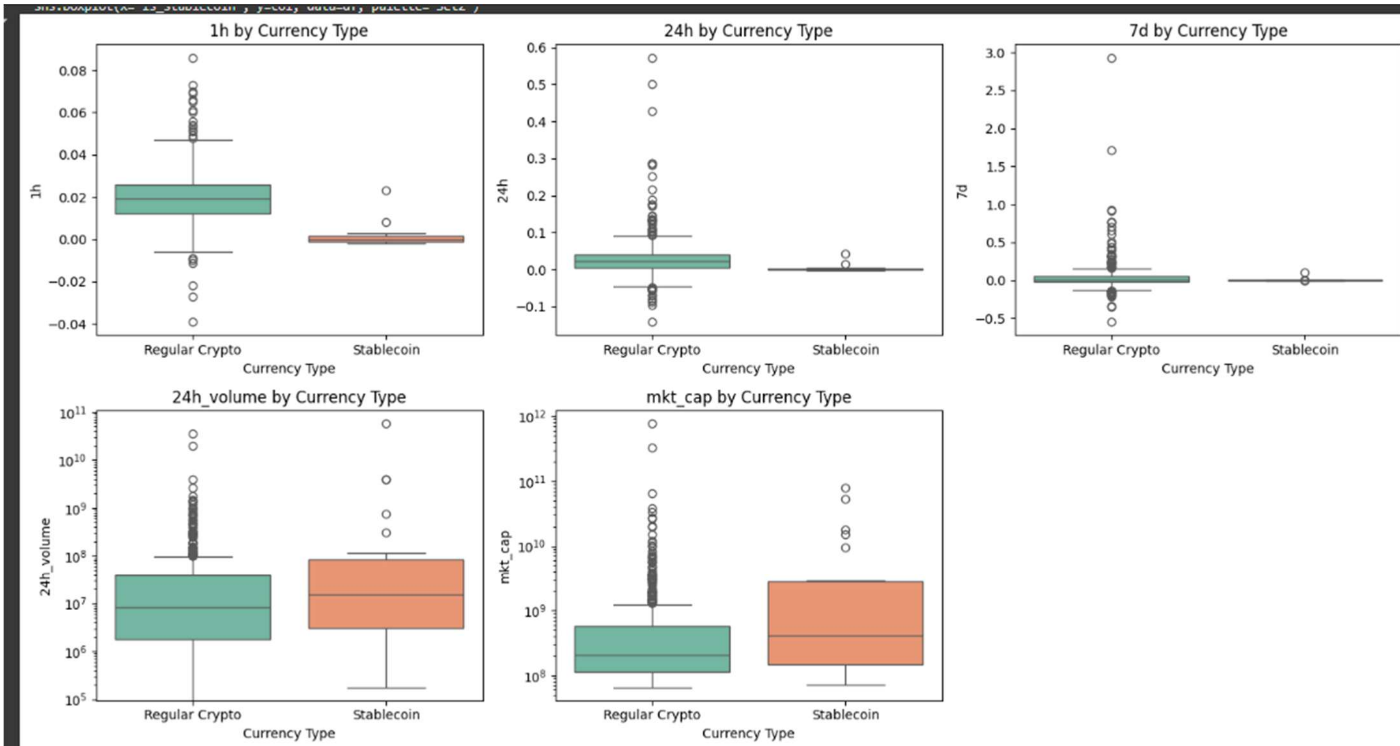

```
Results for Logistic Regression:
Accuracy: 0.9400
              precision    recall  f1-score   support

           0       0.95      0.99      0.97        95
           1       0.00      0.00      0.00         5

    accuracy                           0.94       100
   macro avg       0.47      0.49      0.48       100
weighted avg       0.90      0.94      0.92       100

----------------------------------

Results for Decision Tree:
Accuracy: 0.9600
              precision    recall  f1-score   support

           0       0.97      0.99      0.98        95
           1       0.67      0.40      0.50         5

    accuracy                           0.96       100
   macro avg       0.82      0.69      0.74       100
weighted avg       0.95      0.96      0.96       100

----------------------------------

Results for Random Forest:
Accuracy: 0.9700
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        95
           1       1.00      0.40      0.57         5

    accuracy                           0.97       100
   macro avg       0.98      0.70      0.78       100
weighted avg       0.97      0.97      0.96       100

----------------------------------

Results for SVM:
Accuracy: 0.9500
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        95
           1       0.00      0.00      0.00         5

    accuracy                           0.95       100
   macro avg       0.47      0.50      0.49       100
weighted avg       0.90      0.95      0.93       100

----------------------------------

Results for KNN:
Accuracy: 1.0000
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        95
           1       1.00      1.00      1.00         5
```

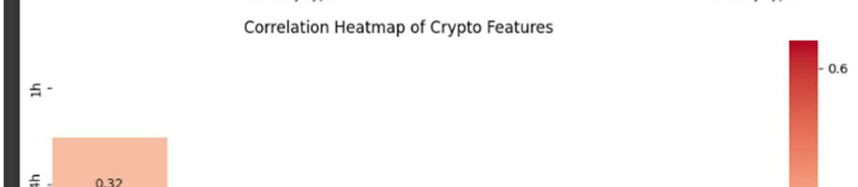






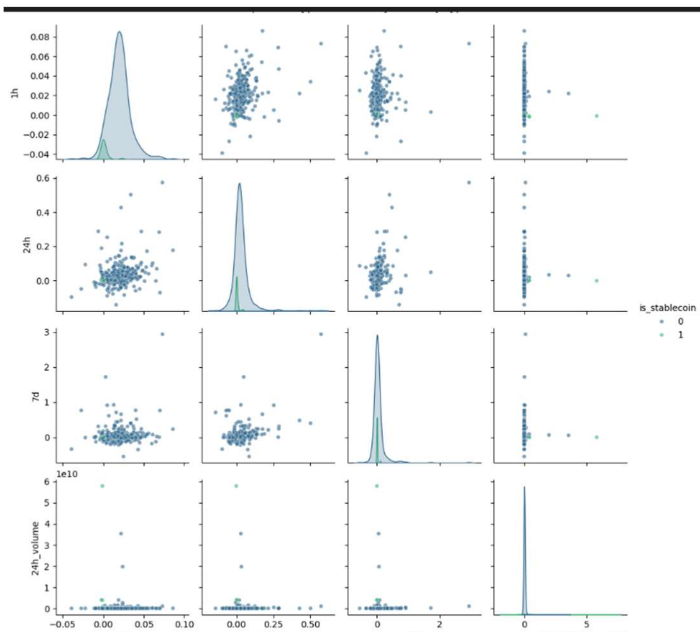




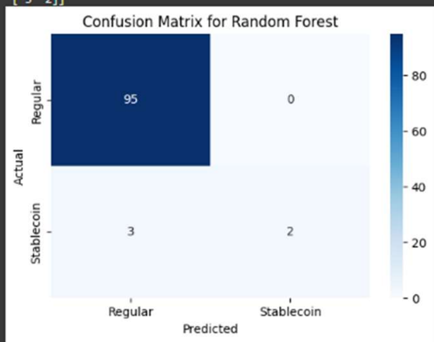



```
Missing values before preprocessing:
 1h             3
24h             3
7d              3
24h_volume      3
mkt_cap         0
is_stablecoin   0
dtype: int64
Results for Logistic Regression:
Accuracy: 0.9400
Precision: 0.0000
Recall: 0.0000
F1-Score: 0.0000
Confusion Matrix:
[[94  1]
 [ 5  0]]
```

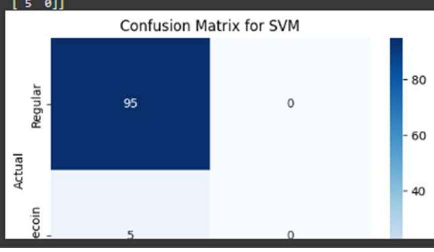

```
----------------------------------

Results for Decision Tree:
Accuracy: 0.9600
Precision: 0.6667
Recall: 0.4000
F1-Score: 0.5000
Confusion Matrix:
[[94  1]
 [ 3  2]]
```

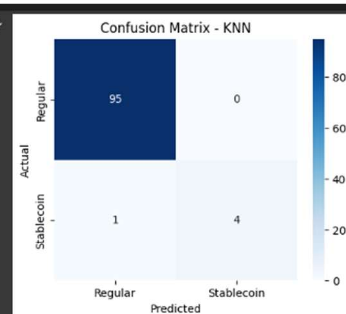


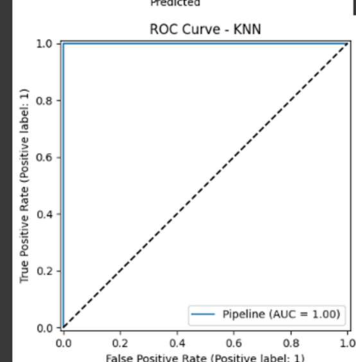
```
----------------------------------

Results for Random Forest:
Accuracy: 0.9700
Precision: 1.0000
Recall: 0.4000
F1-Score: 0.5714
Confusion Matrix:
[[95  0]
 [ 3  2]]
```



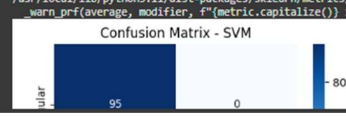

```
----------------------------------

Results for SVM:
Accuracy: 0.9500
Precision: 0.0000
Recall: 0.0000
F1-Score: 0.0000
Confusion Matrix:
[[95  0]
 [ 5  0]]
```



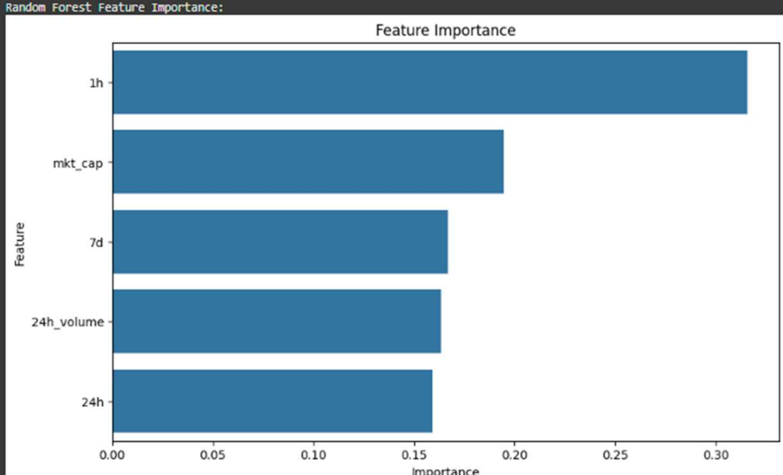


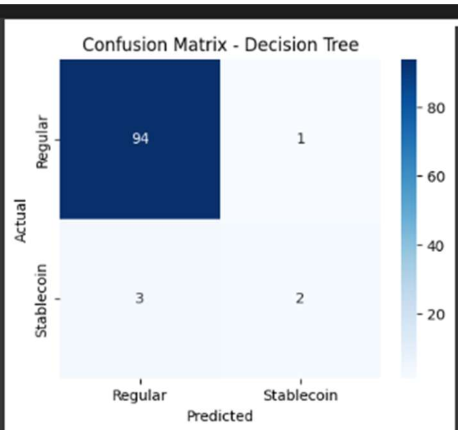


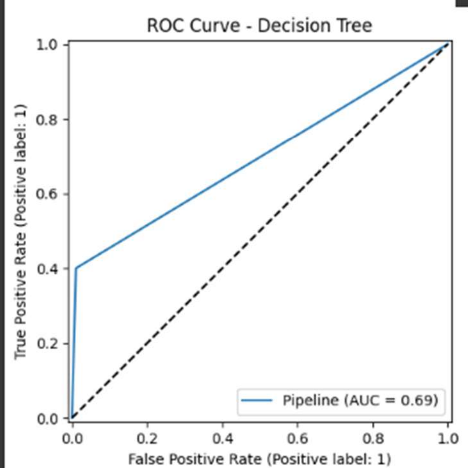

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: Unde
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```



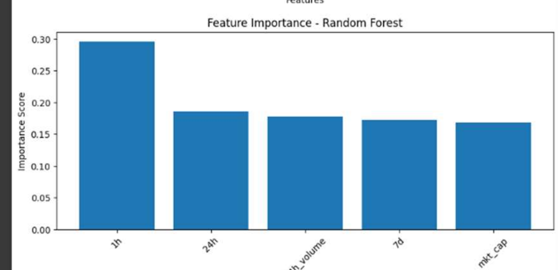

```
Best parameters: {'classifier__max_depth': None, 'classifier__min_samples_split': 10, 'classifier__n_estimators': 50}
Best F1-score: 0.47619047619047616

Cross-validation for Random Forest:
Cross-validation F1 scores: [0.         0.85714286 0.         0.66666667 0.85714286]
Mean F1-score: 0.4762
Standard deviation: 0.3950

Random Forest Feature Importance:
```





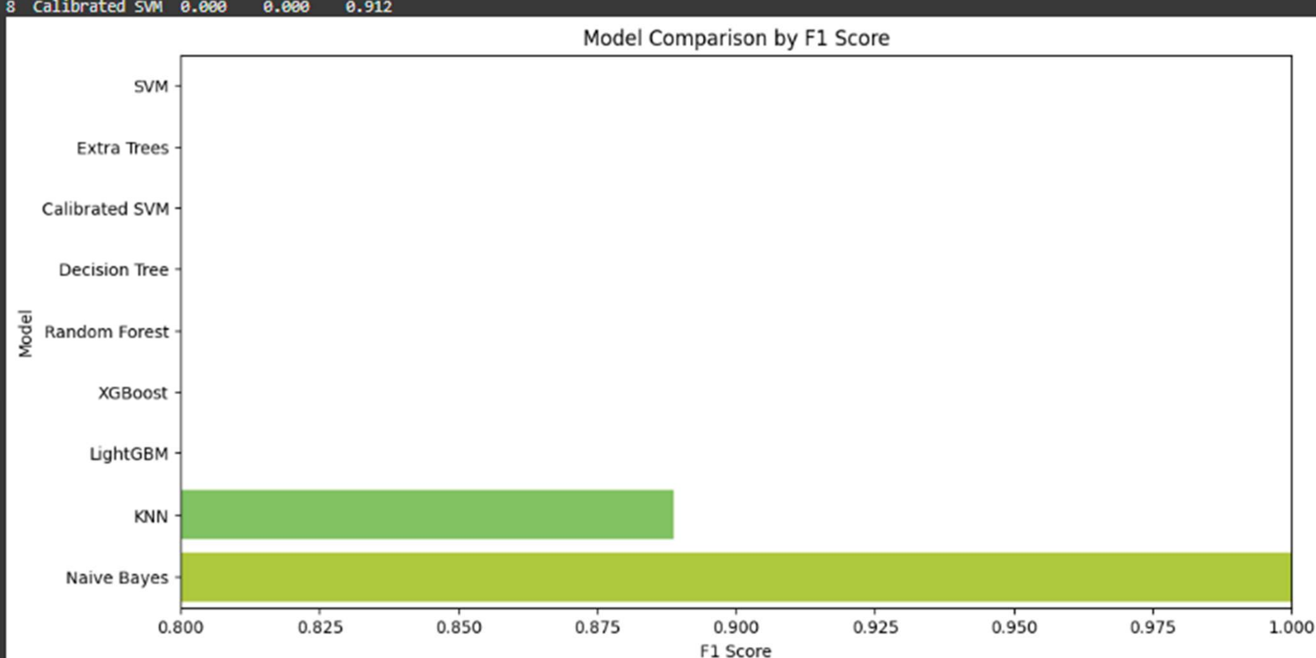


Model	Accuracy	Precision	Recall	F1	ROC AUC
Naive Bayes	1.00	1.000000	1.0	1.000000	1.000000
KNN	0.99	1.000000	0.8	0.888889	1.000000
Random Forest	0.97	1.000000	0.4	0.571429	0.993684
Decision Tree	0.96	0.666667	0.4	0.500000	0.694737
SVM	0.95	0.000000	0.0	0.000000	0.907368
Extra Trees	0.95	0.000000	0.0	0.000000	0.981053



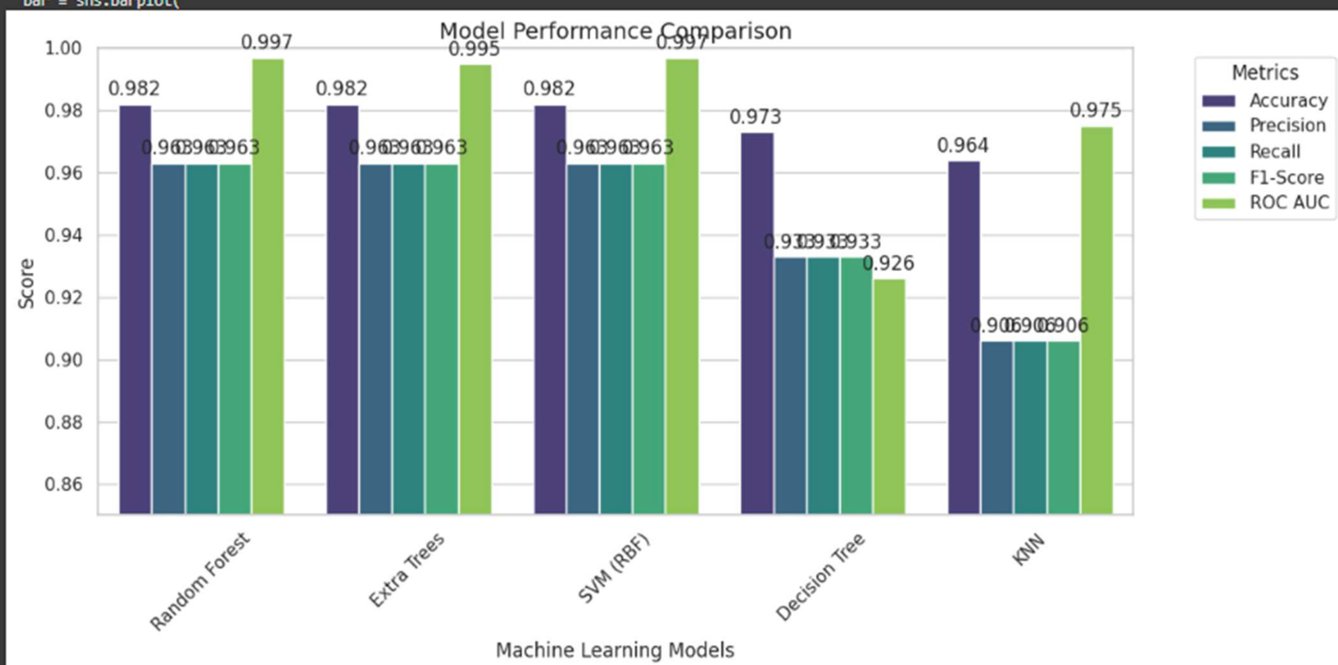
Performance Comparison with SOTA Models:

	Model	CV F1	Test F1	ROC AUC
3	Naive Bayes	0.552	1.000	1.000
0	KNN	0.346	0.889	1.000
4	Random Forest	0.396	0.571	0.994
7	LightGBM	0.414	0.571	0.893
6	XGBoost	0.472	0.571	0.949
2	Decision Tree	0.538	0.500	0.695
1	SVM	0.000	0.000	0.907
5	Extra Trees	0.032	0.000	0.981
8	Calibrated SVM	0.000	0.000	0.912



```
Explained variance ratio: [0.35152435 0.25411224 0.17873509]

Enhanced Feature Set:
Index(['coin', 'symbol', 'price', '1h', '24h', '7d', '24h_volume', 'mkt_cap',
       'date', 'volatility_24h', 'max_daily_change', 'change_variation',
       'price_deviation', 'volume_mcap_ratio', 'log_mcap', 'log_volume',
       'volume_velocity', 'volatility_volume', 'mcap_volatility',
       'short_long_ratio', 'pca_0', 'pca_1', 'pca_2'],
      dtype='object')

Sample of Engineered Features:
   volatility_24h  volume_mcap_ratio     pca_0     pca_1
0        0.189680          -0.050622  3.705077 -2.165857
1        0.350744           0.214597  3.601767 -1.892602
2       -1.479792           2.281341  2.727388 -3.633274
3        0.104038          -0.579671  2.360651 -1.780246
4       -1.557471           0.436549  1.826431 -3.064053
```

```
<ipython-input-40-c9eda6bff828>:21: FutureWarning:

The `errwidth` parameter is deprecated. And will be removed in v0.15.0. Pass `err_kws={'linewidth': 0}` instead.

  bar = sns.barplot(
```



comparison of performance with condition -of -t -art model

A comparative analysis with existing literature highlights the strength and weaknesses of different models. LSTM-based model demonstrated high accuracy in unstable markets than traditional statistical approaches. The transformer model provided better sequence modeling functions, but was resource intensive. The hybrid model provided a balance between interpretation and performance, which made them suitable for practical applications.

conclusion

This research thesis presents intensive analysis of machine learning and deep learning models for forecasts of Cryptocurrency value. LSTM-based models perform better than traditional approaches to capture complex price movements, but there are challenges such as overfit and market unexpected. Transformer models offer advanced sequence modeling skills, but require high calculation resources. Hybrid approaches and attire learning techniques have shown promising consequences for improving predicting accuracy. Future studies should detect adaptation strategies for alternative data sources, learning reinforcement and more reliable Cryptocurrency -Prognosis Models. The findings of this research provide valuable insight into the development of a strong future structure for the financial markets.